

Веселовська Г.В., Ходаков В.Є., Веселовський В.М.

# КОМП'ЮТЕРНА ГРАФІКА



ВИДАВНИЦТВО

**Олди**  
ПЛЮС

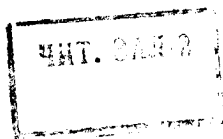
687.32/073,  
B38

**Г.В.Веселовська, В.Є.Ходаков,  
В.М.Веселовський**

# **КОМП'ЮТЕРНА ГРАФІКА**

**Навчальний посібник  
для студентів вищих навчальних закладів**

*Рекомендовано  
міністерством освіти і науки України*



**Херсон 2008**

УДК 004.92

В 38

ББК 30.11

**Г.В.Веселовська, В.Є.Ходаков, В.М.Веселовський**

В 38 Комп'ютерна графіка: Навч. посібник для студентів вищих навчальних закладів / Під ред. В.Є.Ходакова. – Херсон: ОЛДІ-плюс, 2008. – 584 с.

ISBN 966-8447-18-2

Розглянуто математичні основи, апаратне та програмне забезпечення, провідні технології та практичні застосування комп'ютерної графіки; рекомендовано навчальний практикум.

Навчальний посібник призначений для студентів, слухачів ФППК та викладачів вищих навчальних закладів. Може бути використаний для факультативного та самостійного підготування учнів середніх навчальних закладів.

### **Рецензенти:**

**Петров Едуард Георгійович** – д-р техн. наук, професор, зав. кафедрою системотехніки, декан факультету комп'ютерних наук Харківського національного університету радіоелектроніки;

**Фісун Микола Тихонович** – д-р техн. наук, професор, зав. кафедрою комп'ютерних технологій Миколаївського державного гуманітарного університету ім. Петра Могили;

**Ткаченко Володимир Пилипович** – канд. техн. наук, професор, зав. кафедрою інженерної та комп'ютерної графіки Харківського національного університету радіоелектроніки;

**Шарко Олександр Володимирович** – д-р техн. наук, професор кафедри фізики Херсонського державного технічного університету;

**Бергер Еміль Григорович** – канд. техн. наук, професор кафедри нарисної геометрії Херсонського державного технічного університету.

442 500

Рекомендовано Міністерством освіти і науки України  
як навчальний посібник для студентів вищих навчальних закладів.  
(лист № 14/18.2-1205 від 09.07.2003 р.)

**НТБ ВНТУ**

УДК 004.92  
ББК 30.11

ISBN 966-8447-18-2

**М.Вінниця**

© Г.В.Веселовська, 2008

© В.Є.Ходаков, 2008

© В.М.Веселовський, 2008

© ОЛДІ-плюс, 2008

## ЗМІСТ

<b>ВСТУП</b>	<b>7</b>
<b>РОЗДІЛ 1. МАТЕМАТИЧНІ ОСНОВИ КОМП'ЮТЕРНОЇ ГРАФІКИ</b>	<b>17</b>
<b>Глава 1. Моделювання складних геометричних перетворень</b>	<b>17</b>
1.1. Опорні поняття	17
1.2. Моделювання композицій 2D/3D-перетворень	33
1.3. Моделювання проєкцій	44
1.4. Моделювання стереоскопічних зображень	51
Навчальний практикум до гл.1	54
<b>Глава 2. Моделювання реалістичних зображень</b>	<b>57</b>
2.1. Моделювання видимих ділянок об'єктів	57
2.2. Моделювання реалістичної освітленості	71
Навчальний практикум до гл.2	74
<b>Глава 3. Моделювання апроксимованого подання ліній та поверхонь</b>	<b>76</b>
3.1. Моделювання сплайнової апроксимації	76
3.2. Моделювання кусково-лінійної апроксимації методом триангуляції	87
3.3. Векторно-параметричне моделювання криволінійних обводів	93
Навчальний практикум до гл.3	104
Висновки до розд.1	106

<b>РОЗДІЛ 2. ТЕХНОЛОГІЇ КОМП'ЮТЕРНОЇ ГРАФІКИ</b>	<b>107</b>
<b>Глава 4. Графічне програмне забезпечення</b>	<b>107</b>
4.1. Стандартизація графічного ПЗ	107
4.2. Склад та функції графічного ПЗ	134
4.3. Стандартні <i>API</i> для роботи з графікою	139
<b>Глава 5. Графічне апаратне забезпечення</b>	<b>148</b>
5.1. Пристрої введення графічної інформації	149
5.2. Засоби виведення зображень	160
5.3. Пристрої для обробки графіки	169
5.4. Засоби зберігання зображень	173
5.5. Пристрої передавання графічної інформації	183
5.6. Комбіноване графічне обладнання	187
Навчальний практикум до гл.4-5	190
<b>Глава 6. Технології 3D-графіки</b>	<b>194</b>
6.1. Введення до 3D-графіки	194
6.2. Методи побудови 3D-зображень	199
6.3. Технології графічних робочих станцій	219
<b>Глава 7. Технології мультимедіа, гіпермедіа, цифрового фото та інтерактивного відео</b>	<b>228</b>
7.1. Визначення мультимедіа та гіпермедіа	228
7.2. Відео- та аудіо-технології мультимедіа	231
7.3. Технології гіпермедіа в комп'ютерній графіці	240
7.4. Переваги та недоліки цифрових фотокамер	246
7.5. Технології цифрового фото	250
7.6. Технології інтерактивного цифрового відео	255
7.7. Галузі застосування <i>MPEG</i> -відео	260
7.8. Методи стискання цифрової відеоінформації	264
7.9. Програмне забезпечення відеомонтажу	267
7.10. Апаратне забезпечення цифрового відео	270

---

<b>Глава 8. Комп'ютерна графіка в мережі <i>Internet</i></b>	<b>275</b>
8.1. Особливості <i>Internet</i> -графіки	275
8.2. Програми для перегляду <i>Web</i> -сторінок	283
8.3. Програми-редактори <i>Web</i> -сторінок	289
8.4. Оптимізація ілюстрованих <i>Web</i> -сторінок	293
Навчальний практикум до гл.6-8	300
<b>Глава 9. Технології програмування графіки</b>	<b>303</b>
9.1. Програмування та анімація графіки на базі мов програмування високого рівня	303
9.2. Методи швидкого синтезу та анімації зображень із застосуванням мови програмування <i>Assembler</i>	318
<b>Глава 10. Видавничі комп'ютерні технології</b>	<b>339</b>
10.1. Ключові тенденції комп'ютерної поліграфії	339
10.2. Цифровий технологічний цикл поліграфічного виробництва	342
Навчальний практикум до гл.9-10	362
Висновки до розд.2	368
<b>РОЗДІЛ 3. ЗАСТОСУВАННЯ КОМП'ЮТЕРНОЇ ГРАФІКИ</b>	<b>370</b>
<b>Глава 11. Комп'ютерна графіка у творчій та діловій сферах</b>	<b>370</b>
11.1. Основні поняття про комп'ютерні мистецтво, дизайн, рекламу та поліграфію	370
11.2. Формування растрового та векторного ілюстраційного матеріалу	383
11.3. 3D-графіка, мультимедіа-дизайн та <i>Web</i> -дизайн	412
11.4. Швидке формування ділової документації та рекламної продукції	432
Навчальний практикум до гл.11	458

---

<b>Глава 12. Комп'ютерна графіка у науковій та виробничій сферах</b>	<b>466</b>
12.1. Функції та застосування систем наукової комп'ютерної графіки	466
12.2. Основні поняття про інженерну комп'ютерну графіку	479
12.3. Огляд базових класів інженерних САПР	509
12.4. Комплексне застосування САПР	541
Навчальний практикум до гл.12	548
Висновки до розд.3	562
<b>ПІСЛЯМОВА</b>	<b>563</b>
<b>ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ</b>	<b>564</b>
<b>ЛІТЕРАТУРА</b>	<b>568</b>
<b>ВІДОМОСТІ ПРО АВТОРІВ</b>	<b>582</b>

---

---

## ВСТУП

---

---

*Комп'ютерна графіка* – це галузь знань, яка вивчає та розробляє засоби та методи створення та перетворення графічних зображень об'єктів за допомогою комп'ютеру [63, 120].

*Основні задачі комп'ютерної графіки:* введення до комп'ютеру інформації, що початково має графічну форму або визначає її; обробка, оптимізація характеристик, зберігання на носіях, захист, передавання засобами локальних та глобальних мереж цієї інформації; виведення інформації в графічній формі з комп'ютеру [120, 137-139].

*Під графічною формою подання інформації розуміють:* ескізи, креслення, візуальне подання каркасних, поверхневих та твердотільних 3D-моделей різноманітних об'єктів (природних та штучних, статичних та динамічних, живих та неживих), схеми, діаграми, графіки, рисунки, фотографії, відео, анімацію, голограми, мультимедійну інформацію тощо [28, 58, 91, 120].

Маніпуляції з графічною інформацією здійснюють переважно в процесі діалогу людини з комп'ютером.

*В основу комп'ютерної графіки покладено фундаментальний теоретичний апарат* аналітичної та диференційної геометрії, векторної алгебри, нарисної геометрії та креслення, графів, чисельних методів розв'язування математичних задач, математичної логіки, методів оптимізації, прийняття рішень, розпізнавання образів, штучного інтелекту [64, 79, 120].



*У комп'ютерній графіці використовують геометричну версію математичного моделювання: дво-, три- та  $n$ -вимірні ( $n > 3$ ) зображення розглядають як такі, що складені з точок, ліній та поверхонь (цей підхід традиційно вважають найнаочнішим та найзручнішим).*

*Математичне забезпечення комп'ютерної графіки становлять [79, 120]:*

– геометричні моделі, складовими частинами яких є елементарні компоненти – графічні примітиви (точки, відрізки суцільних та пунктирних прямих ліній, дуги кіл та еліпсів, трикутники, прямокутники, багатокутники), поєднані в ієрархічні структури даних (списки, стеки, черги, дерева, мережі);

– методи та алгоритми перетворення геометричних моделей, складовими частинами яких є елементарні операції перенесення, масштабування, обертання та дзеркального відбиття зображень, виділення вікна.

*Апаратне забезпечення комп'ютерної графіки становлять:* засоби введення, вказування та виведення графічної інформації; обчислювальні засоби; засоби зберігання графічної інформації у внутрішній та зовнішній пам'яті ЕОМ; засоби передавання графічної інформації каналами внутрішнього комп'ютерного зв'язку та міжкомп'ютерного зв'язку (локальними та глобальними мережами).

*Програмне забезпечення комп'ютерної графіки поділяють на базове та прикладне, апаратно-залежне та апаратно-незалежне.*

*Для створення прикладного графічного ПЗ застосовують такі класи інструментальних засобів [91]:*

1) стандартні графічні пакети (наприклад, графічні редактори, системи автоматизованого проектування, видавничі системи);

2) стандартні програмні системи, що не спеціалізуються на графіці, але підтримують певний набір графічних функцій (наприклад, офісні пакети та текстові процесори);

3) авторські середовища розробки графічного ПЗ (авторські системи для створення мультимедійної та гіпермедійної продукції, наприклад, для побудови презентацій тощо);

4) мови програмування високого рівня, що містять візуальні компонентні об'єктно-орієнтовані середовища розробника;

5) мови програмування низького рівня.

*Залежно від використаних розробником засобів побудови зображення, що їх надають мови програмування, графічні програми можна класифікувати так:* ті, що базуються на стандартних компонентах, процедурах та функціях графічних бібліотек; ті, що створюють графічні зображення засобами BIOS (*Basic Input/Output System*); ті, що реалізують найшвидший спосіб (пряме звертання до портів відеоадаптеру та чарунок відеопам'яті); ті, в яких код, написаний мовою програмування високого рівня, доповнений вставками, що виконують графічні дії мовою асемблера або в машинному коді [7].

*Жоден із вказаних п'яти класів інструментальних засобів не є універсальним, будучи орієнтованим на свої класи задач та користувачів.*

Системи, що належать до перших трьох класів, роблять швидким та ефективним процес створення графічного ПЗ за типовими зразками (шаблонами). Вони орієнтовані на користувача, що не має професійної підготовки з програмування. Але

в процесі розв'язування нешаблонних задач ці системи надають недостатньо гнучкий інструментарій, надто великий за обсягом та повільно діючий кінцевий графічний програмний продукт.

Мови програмування є гнучким інструментарієм розробки, що дозволяє створити ефективне та швидкодіюче графічне ПЗ. Але процес розробки є повільним та трудомістким, вимагає спеціальних знань з програмування.

Найоптимальнішим є сумісне використання в процесі створення графічного ПЗ інструментальних засобів програмних пакетів, що належать до класів 1-3, та мов програмування.

В основу створення ефективного прикладного графічного ПЗ покладено ряд аксіоматичних положень: чітке визначення мети, зацікавленість у кінцевому результаті та наполегливість у його досягненні; інтенсивне використання широко відомих у програмуванні принципу "розділяй та володій" та методу "покрокової деталізації"; відмова від використання складних та нових алгоритмів там, де можна скористатися простими та існуючими алгоритмами; забезпечення легкості прочитування ПЗ; універсальність ПЗ по відношенню до дисплею будь-якого типу, для досягнення якої, це ПЗ повинне мати відповідний інтерактивний або автономний модуль настроювання [77].

*Важливу роль відіграють такі перспективні комп'ютерні графічні технології [45, 120]:*

– 3D-графіка на базі спеціалізованих 3D-карт та графічних робочих станцій;

– програмування графіки на базі інструментальних засобів візуальних компонентних об'єктно-орієнтованих середовищ мов програмування високого рівня;

- швидкий синтез графічних зображень на базі мов програмування низького рівня;
- каркасне, поверхневе та твердотільне 3D-моделювання та реалістична візуалізація;
- анімаційні графічні технології;
- мультимедійні та гіпермедійні графічні технології;
- інтерактивне відео та віртуальна реальність;
- видавничі графічні комп'ютерні технології;
- технології комп'ютерної графіки на *Web*-сторінках глобальної мережі *Internet*;
- інтелектуальні САПР у мережі *Internet*.

**Комп'ютерна графіка застосовується в багатьох сферах діяльності:** промисловості, науці, мистецтві, телебаченні, журналістиці, освіті, маркетингу та бізнесі, видавництві, криміналістиці тощо [130].

**Базовими класами систем комп'ютерної графіки та провідними галузями їх практичного застосування є:** ділова, наукова, інженерна та ілюстраційна комп'ютерна графіка [63, 137].

**Системи ділової комп'ютерної графіки** призначені для наочного графічного подання даних, які зберігаються в електронних таблицях та базах даних (переважно – в сферах бізнесу, маркетингу, управління підприємствами, економічних розрахунків); вони дозволяють наочно відображати співвідношення чисельних показників у зручній для сприйняття формі у вигляді статичних та динамічних, двовимірних та тривимірних графіків, схем, діаграм [76, 136].

Типові приклади цих систем: *Excel*, *QuattroPro*, *Lotus 1-2-3*, *SuperCalc*, *FoxGraph*, *Boeing Graph*, *PowerPoint*, *Word*, *Лексикон-XL*.

*Системи наукової комп'ютерної графіки призначені для:* динамічної наочної візуалізації процесу та результатів проведення наукових експериментів, автоматизованого проектування наукових та науково-технічних задач, формування наукової документації з застосуванням спеціальної нотації (математичних, фізичних та хімічних формул тощо); дослідження географічних, геологічних, гідрологічних, сейсмологічних, екологічних, метеорологічних, астрономічних та інших природних об'єктів, процесів, явищ та систем, нафтогазових розвідки та видобування, комп'ютерної картографії, медичного діагностування тощо [33, 66].

Типові приклади цих систем: *MathCAD, Mathematica, MathLAB, SPSS, Maple, Statistica, Axum, NCSS, S-Plus, StatGraphics, GraphIt, Super Graph, GS Didger, GS Surfer, GS Grapher.*

#### **Системи інженерної комп'ютерної графіки:**

– призначені для автоматизації креслярсько-графічних та конструкторських робіт у процесі проектування компонентів та систем механічних, електричних, електромеханічних, електронних та радіоелектронних пристроїв та приладів, у будівництві та архітектурі;

– надають можливість виконувати в реальному часі каркасне, поверхневе та твердотільне 3D-моделювання, морфінг, анімацію та реалістичну візуалізацію;

– забезпечують здійснення промислового дизайну;

– дозволяють передати комп'ютеру більшу частину рутинної роботи з проектування та вивільнити завдяки цьому час інженера-конструктора для творчої діяльності, суттєво підвищуючи якість результатів та швидкість проектування [41, 79].

Типові приклади цих систем: *AutoCAD, КОМПАС, ProEngineer, Mapguide Author, bCAD, Engineering Geometry Assistant,*

*VariCAD, Femap, Solide Pipe Designer, Archicad, ArcView, 3D Home Architect Deluxe, P-CAD, OrCAD, Electronics Workbench, MicroCap, ArCon.*

Перспективним напрямком підвищення продуктивності систем інженерної комп'ютерної графіки є їх функціонування в складі інтелектуальних САПР у мережі *Internet*.

**Системи ілюстраційної комп'ютерної графіки призначені для** створення та художньої обробки комп'ютерних зображень, які відіграють роль: ілюстраційного матеріалу – ілюстрацій до друкованих та електронних видань (рисуноків, фотографій, ескізів, умовних схем, географічних карт, відеоматеріалів, мультимедіа-матеріалів, *Web*-матеріалів тощо); дизайнерських розробок; рекламного оздоблення; витворів мистецтва. Вони дозволяють формувати та перетворювати графічні об'єкти настільки ж легко, як масиви чисел або тексти [35, 63, 130, 137, 165, 167].

Типові приклади цих систем: *Illustrator, CorelDraw, Photoshop, Painter, 3D Studio MAX, Maya, Bryce 3D, FaceWorks Studio, Poser, PowerPoint, HyperMethod, Director, ImageReady, FreeHand, Premiere, PageMaker, Ventura, QuarkXPress.*

**Широко відомими класами та галузями практичного застосування систем комп'ютерної графіки, що знаходяться на стику ділової, наукової, інженерної та ілюстраційної графіки та дизайну, інтенсивно взаємодіючи між собою, є:** 2D- та 3D-моделювання, реалістична візуалізація, анімація (мультиплікація), морфінг та відеодизайн; управління технологічними процесами; мистецтво, реклама та дизайн; видавнича діяльність (публікування газет, журналів, книжок, буклетів, календарів, листівок, бланків, афіш, рекламних оголошень тощо); тренажери; комп'ютерні ігри та розваги [63, 138, 139].

*Функції комп'ютерних графічних систем для 2D- та 3D-моделювання, реалістичної візуалізації, анімації (мультиплікації), морфінгу та відеодизайну:*

- створення за допомогою ЕОМ анімаційних (мультиплікаційних) фільмів для демонстрування поведінки об'єктів, процесів та явищ, які моделюють роботу фізіологічних систем, протікання потоків рідин та газів, деформацію конструкцій під дією навантаження, хімічні та ядерні реакції тощо;

- імітація природної анатомії, динаміки та пластики рухів, міміки живих істот, їх іміджу та довіклля на різних фазах їх життєвої діяльності (навчання, професійна діяльність, дозвілля, сон тощо);

- створення художніх відеофільмів та рекламних відеороликів, насичених 3D-спецефектами.

*Функції комп'ютерних графічних систем для управління технологічними процесами:* стеження за станом окремих виробництв та цілих цехів, інформування персоналу про критичні ситуації, що здійснюють на базі вивчення графічних відображень технологічних процесів у режимі реального часу та практикують на великих підприємствах в умовах гнучких автоматизованих виробництв.

*Функції комп'ютерних графічних систем для втілення мистецтва, реклами та дизайну:* створення витворів мистецтва; музейна та реставраційна діяльність; оздоблювання внутрішнього та зовнішнього інтер'єрів помешкань; підготування друкованої продукції рекламно-інформаційного характеру; виготовлення слайдів, відеокліпів та мультимедійних презентацій для подання комерційної, наукової та навчальної інформації; підготування інформаційних та рекламних роликів для телебачення; розвиток індустрії ігор та розваг.

*Функції комп'ютерних графічних систем для видавничої діяльності (публікування реклами, газет, журналів, книжок тощо):* спрощення та скорочення процесів підготовки, перевірки, пересилання та виготовлення друкованих матеріалів різного призначення; підвищення ефективності та якості роботи з кольором у процесі виготовлення друкованої продукції.

*Функції комп'ютерних графічних систем-тренажерів:* тренування льотчиків, шоферів, диспетчерів аеропортів та залізничних вузлів, операторів складних хімічних та енергетичних установок, військових тощо на базі комп'ютерного графічного моделювання відповідних середовищ, об'єктів та ефектів.

*Функції комп'ютерних графічних систем для здійснення ігор та розваг:* тренування логічного мислення, швидкості реакції, пам'яті; розширення кругозору, збільшення обсягу знань та посилення пізнавальної активності; профорієнтація та соціальна орієнтація.

## Питання для самоперевірки

1. Дайте визначення та сформулюйте задачі комп'ютерної графіки, наведіть типові приклади графічної форми подання інформації.

2. Який математичний апарат складає теоретичний фундамент комп'ютерної графіки?

3. Наведіть класифікацію апаратного забезпечення комп'ютерної графіки.

4. Перелічіть основні класи ПЗ комп'ютерної графіки та інструментальних засобів для його розробки.



5. Назвіть ключові класи та галузі практичного застосування систем комп'ютерної графіки, прогресивні комп'ютерні графічні технології.

## РОЗДІЛ 1. МАТЕМАТИЧНІ ОСНОВИ КОМП'ЮТЕРНОЇ ГРАФІКИ

### ГЛАВА 1. МОДЕЛЮВАННЯ СКЛАДНИХ ГЕОМЕТРИЧНИХ ПЕРЕТВОРЕНЬ

#### 1.1. Опорні поняття

**Системи та типи координат у комп'ютерній графіці.** Сучасні системи комп'ютерної графіки надають можливість працювати з будь-якими системами та типами координат. Найчастіше застосовують декартову, афінну, однорідну, полярну, циліндричну та сферичну системи координат, використовуючи при цьому абсолютні, відносні, визначені користувачем, світові, нормалізовані та фізичні типи координат.

Вибір системи координат визначений характером розв'язуваних задач, а вибір типу координат – структурою подання зображень та характером процесу обробки графічних даних [64, 79, 120].

*Афінну та декартову системи координат* однозначно визначають: початок системи координат; взаємне розташування координатних осей; співвідношення одиничних відрізків на координатних осях. Якщо одиничні відрізки не є рівними, система координат є афінною; якщо одиничні відрізки є рівними, але хоча б один кут між осями не дорівнює  $90^\circ$  ( $\pi/2$ ),

НТБ ВІНТУ  
М.Вінниця

система координат є *косокутною декартовою*; якщо одиничні відрізки є рівними та кути між осями є прямими, система координат є *прямокутною декартовою*.

Афінні та декартові системи координат є правоорієнтованими (лівоорієнтованими), якщо позитивні напрямки координатних осей  $Ox$  та  $Oy$  можна сумістити шляхом обертання осі  $Ox$  проти (за) годинникової стрілки на кут, менший  $180$  ( $\pi$ ).

Відповідність між деякою довільною точкою  $3D$ -простору та набором її координат  $(x, y, z)$  можна описати за допомогою: радіус-вектору  $r = xe_x + ye_y + ze_z$  ( $e_x, e_y, e_z$  – одиничні вектори координатних осей); матриці  $(x, y, z)$ .

**Метод однорідних координат** полягає в тому, що замість реальної точки  $A(x, y)$  на  $2D$ -площині формально вводять до розгляду точку  $A(x, y, c)$  у  $3D$ -просторі: фіктивна  $z$ -координата має значення скалярної константи  $c \neq 0$  (найчастіше  $c = 1$ ). У  $3D$ -просторі здійснюють подібні дії:  $A(x, y, z) \rightarrow A(x, y, z, 1)$ . Однорідні координати є основним типом координатних систем у *проективній геометрії*. У комп'ютерній графіці – це зручний штучний прийом для лінеалізації перспективних зображень. Однорідні координати дозволяють записувати невласні (безмежно віддалені) точки простору та подавати афінні перетворення в зручній матричній формі, уникаючи зайвого переповнення розрядної сітки ЕОМ завдяки нормалізації чисел [41, 79, 120].

**Полярні координати** ( $\rho, \varphi$ ) *точки на площині* – це два числа, що визначають її положення відносно фіксованої точки  $O$  (*полюсу*) та променя  $Op$  (*полярної осі*). Координата  $\rho$  – *полярний радіус*, який визначає відстань точки від полюсу. Координата  $\varphi$  – *полярний кут*, який визначає кут обертання осі  $Op$  до співпадання з променем, який з'єднує полюс з точкою

$(\rho, \varphi)$ . Перехід від декартових координат до полярних та навпаки виконують згідно формул

$$\begin{aligned} x &= \rho \cos \varphi, \quad y = \rho \sin \varphi, \quad \rho = \sqrt{x^2 + y^2}, \\ \varphi &= \operatorname{Arctg} \left( \frac{y}{x} \right) = \operatorname{Arcsin} \left( \frac{y}{\rho} \right) \end{aligned} \quad (1.1)$$

*Сферичну систему координат однозначно* визначають: полюс (фіксована точка  $O$ ); орієнтована пряма  $g$ , яка проходить крізь полюс; сімейство напівплощин, обмежених прямою  $g$  (зокрема, напівплощина нульового меридіану); сімейство кінчних поверхонь з вершинами в полюсі та з віссю, що співпадає з прямою  $g$ ; сімейство сфер з центром у полюсі. При цьому задані такі параметри: кут  $\varphi$ , який утворює поточна напівплощина з напівплощиною нульового меридіану (*географічна довгота*); кут  $\theta$  між позитивним напрямком прямої  $g$  та твірною бо-кової поверхні конусу (*полярна відстань*); радіус  $\rho$  сфери. Виходячи з цих позначень, точку в сферичній системі координат описують так:  $(\rho, \varphi, \theta)$ .

*Циліндричну систему координат однозначно* визначають: фіксована точка  $O$  (початок координат); орієнтована пряма  $g$ , яка проходить крізь початок координат; сімейство площин, перпендикулярних прямій  $g$ ; сімейство напівплощин, обмежених прямою  $g$  (зокрема, напівплощина нульового меридіану); сімейство циліндрів, вісь яких співпадає з прямою  $g$ .

При цьому задані: для сімейства площин, перпендикулярних прямій  $g$  – відстань  $z_0$  від точки  $O$  до площини; для сімейства напівплощин – кут  $\varphi$ , який поточна напівплощина утворює з напівплощиною нульового меридіану; для сімейства

циліндрів – радіус циліндру  $\rho$ . Виходячи з цих позначень, точку в циліндричній системі координат описують так:  $(\rho, \varphi, z_0)$ .

*Дано характеристику основних типів координат, які використовують у комп'ютерній графіці [41, 120]:*

– абсолютні координати визначають позицію деякої довільної точки відносно початку системи координат;

– відносні координати визначають позицію однієї точки відносно деякої іншої точки;

– координати, визначені користувачем, показують позицію точки в тій системі координат, яка обрана користувачем та не залежить від конкретних пристроїв відображення;

– світові координати – це декартові координати, що не залежать від конкретних пристроїв відображення та використовуються в прикладній програмі в процесі задавання графічних вхідних та вихідних даних;

– нормалізовані координати можуть бути задані в будь-якій системі координат, незалежній від конкретних пристроїв відображення, та повинні бути нормовані відносно деякого діапазону (як правило, відносно відрізка одиничної довжини  $[0; 1]$ );

– фізичні координати задають у системі координат, залежній від конкретних пристроїв відображення.

**Операції видового перетворення (кадрування) та відсікання зображень на екрані комп'ютеру.** В процесі виведення зображень на екран, що має обмежені розміри, потрібно конкретизувати, яку частину необмеженого віртуального простору та під яким кутом зору ми хочемо побачити на екрані. При цьому повинне бути виконане таке перетворення: світових координат зображення, незалежних від будь-яких пристроїв відображення – у відповідні фізичні координати графічного дисплею.

Основними поняттями в цьому випадку є вікно, поле виведення, операція видового перетворення (кадрування) та операція відсікання [79, 120].

**Вікно** – це задана частина віртуального простору (прямокутної форми), в якій формують зображення, призначене для виведення на графічні пристрої.

**Поле виведення** – це задана частина простору візуалізації, на якій зображують вікно.

**Операція видового перетворення (кадрування)** є операцією відображення вікна на поле виведення.

Розглянемо математичну модель типового видового перетворення (вікно задане в світовій системі координат, поле виведення має прямокутну форму та не покриває всього екрану):

$$x_p = \frac{x_w - x_{cw}}{l_x} L_x + x_{cp}; \quad (1.2)$$

$$y_p = \frac{y_w - y_{cw}}{l_y} L_y + y_{cp}, \quad (1.3)$$

де  $x_w, y_w$  – координати точки вікна в світовій системі координат;  $x_p, y_p$  – координати точки поля виведення в фізичній системі координат;  $x_{cw}, y_{cw}$  – координати центру вікна в світовій системі координат;  $x_{cp}, y_{cp}$  – координати центру поля виведення в фізичній системі координат;  $l_x, l_y$  – розміри вікна, що вимірюють від його центру в напрямку осей  $Ox$  та  $Oy$  в світовій системі координат;  $L_x, L_y$  – розміри поля виведення, що вимірюють від його центру в напрямку осей  $Ox$  та  $Oy$  в фізичній системі координат.

Якщо вікно задане в системі координат користувача, оберненій відносно світової системи координат на деякий кут, схема обчислень є двокроковою: спочатку виконують обертання навколо точки  $(x_{cw}, y_{cw})$ , а потім – кадрування.

*Операція відсікання* є операцією видалення графічних примитивів або їх частин, розташованих поза заданою областю. Вона є актуальною тоді, коли необхідно вивести на екран певну частину зображення, а іншу частину, що не потрапила до вікна, необхідно зробити невидимою: графічні примітиви, розташовані поза межами поля виведення, не відображають на екран; якщо примітив частково потрапляє до екрану, його відсікають по межі поля виведення.

Найдоцільніше здійснювати операцію відсікання до операції відображення: невидимі лінії не будуть підпадати під видове перетворення, що зекономить час та ресурси. Проте, якщо сторони вікна нахилені відносно осей світової системи координат, спочатку здійснюють відображення, а потім – відсікання невидимих ліній отриманого зображення.

У процесі розв'язування задачі відсікання можуть виникнути *два характерних випадки визначення зображення*: як списку точок; як множини відрізків прямих ліній. У першому випадку будь-яку точку, залежно від результату порівняння її координат з координатами меж поля виведення, в будь-який момент часу можна або показати на екрані, або видалити з нього. У другому випадку задача ускладнюється, оскільки кожен відрізок необхідно перевірити на відповідність одній із трьох умов: відрізок розташований у межах вікна; відрізок розташований зовні вікна; відрізок перетинає вікно (перші два варіанти є тривіальними, а в третьому потрібно виконувати операцію відсікання).

В основу алгоритмів виділення видимих ділянок відрізків покладено такий принцип: або відрізок повністю лежить у межах прямокутного поля виведення, або повністю зовні цих

меж, або його можна розділити, подовживши межі поля виведення таким чином, щоб невидима частина була покровоно видалена (рис. 1.1).

Операції видового перетворення та відсікання дозволяють виводити на екран тільки ті частини зображення, що знаходяться всередині вікна. Пересуваючи вікно та призначаючи йому певні розміри, можна отримувати кінематографічні ефекти (великого або малого плану, панорамування тощо).

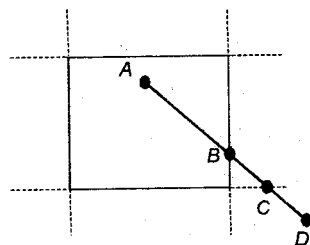


Рис. 1.1. Схема виділення видимої ділянки відрізка

**Растрова розгортка неперервних геометричних об'єктів.** *Растром* (растровою площиною, дискретною площиною, цілочисельними ґратками) називають множину тих точок картинної площини, що мають цілочисельні координати.

Зручно вважати, що на  $2D$ -площину накладено сітку з квадратними чарунками  $1 \times 1$ , геометричні центри яких співпадають з елементами растру.

Точки  $A_{ij}(x_i, y_j)$  та  $A_{pq}(x_p, y_q)$  називають *чотиризв'язними сусідами* на растрі, якщо їх координати відповідають простій логічній умові  $((|x_i - x_p| = 1) \& (y_j = y_q)) \vee ((|y_j - y_q| = 1) \& (x_i = x_p))$ , та *восьмизв'язними сусідами*, якщо  $(|x_i - x_p| \leq 1) \& (|y_j - y_q| \leq 1)$ .

Будь-яка точка растру, за виключенням граничної, має чотирьох чотиризв'язних сусідів та вісім восьмизв'язних. Квадратні околиці чотиризв'язних елементів мають загальну сторону, а восьмизв'язних – загальну сторону або вершину.

Чотиризв'язним (восьмизв'язним) шляхом на растрі називають таку впорядковану множину точок растру, що



$\forall (i = 1, \dots, n-1)$  точки  $A_i$  та  $A_{i+1}$  є чотиризв'язними (восьмизв'язними) сусідами. Чотиризв'язною (восьмизв'язною) **множиною на растрі** називають таку множину точок растру, будь-які два елементи якої можна з'єднати чотиризв'язним (восьмизв'язним) шляхом. Простою чотиризв'язною (восьмизв'язною) **кривою на растрі** називають таку множину точок  $A_1, \dots, A_n$  растру, що  $A_2, \dots, A_{n-1}$  мають по дві сусідніх чотиризв'язних (восьмизв'язних) точки, а  $A_1$  та  $A_n$  – по одній.

**Растрове подання (растрова розгортка, дискретне подання на цілочисельних ґратках) неперервного геометричного об'єкту** (лінії, області, фігури) – це така множина елементів растру, що з певним припустимим рівнем точності та при дотриманні певних умов є дискретним наближенням цього об'єкту.

Растрове подання геометричного об'єкту не є однозначним, оскільки: можна встановлювати різні рівні точності та умови; є велика кількість концепцій та алгоритмів наближення неперервних об'єктів дискретною множиною точок. Важливим є вибір найбільш ефективних алгоритмів растрової розгортки тих чи інших класів геометричних об'єктів [64].

Задача побудови растрової розгортки не належить до типових задач, які програмісти розв'язують повсякденно, оскільки алгоритми растрової генерації геометричних примітивів (прямих ліній, кіл, еліпсів тощо), зафарбовування многогранників та областей площин на екрані монітору, та багато інших алгоритмів реалізовані у будь-якій сучасній мові програмування та системі, що підтримує графічні функції [120].

Знання та вміння в галузі алгоритмізації та програмної реалізації растрової генерації геометричних об'єктів необхідні в

таких випадках: коли атрибути ініційованих пікселів (видимість, колір та інші) залежать від певних умов, наприклад, від розташування цих пікселів на певних лініях, усередині певних областей або фігур (такі ситуації є типовими в процесі розробки алгоритмів видалення невидимих ліній та поверхонь); при роботі з принтерами, плотерами, маніпуляторами типу "миша" тощо [63, 120].

Растрове розгортання відрізка – це послідовна ініціалізація тієї множини пікселів екрану дисплею, що зображує відрізок прямої лінії, розташований між двома заданими точками. Ця множина отримала назву растрового подання або *растрової розгортки відрізка*. Виходячи з цього, необхідно знати, яким чином організовано вказану множину пікселів екрану; володіти способом генерації цієї множини (алгоритмом, який послідовно будує точки множини).

Розв'язок задачі растрового розгортання відрізка не є однозначним – він залежить від того, в якому класі растрових кривих ви хочете отримати растровий образ відрізка та якими властивостями ви хочете його наділити. Навіть якщо зупинитися на елементарних растрових кривих, будемо мати два подання растрової розгортки відрізка: *чотиризв'язна растрова розгортка відрізка*, коли перехід від поточної до наступної точки виконується на одну з чотирьох сусідніх точок растру, квадратні околиці яких мають загальну сторону з квадратною околицею поточної точки; *восьмизв'язна растрова розгортка відрізка*, коли перехід від поточної до наступної точки виконується на одну з восьми сусідніх точок растру, квадратні околиці яких мають загальну сторону або вершину з квадратною околицею поточної точки (рис. 1.2).

Простим методом отримання растрової розгортки відрізка є послідовна ініціалізація всіх точок растру, квадратні околиці яких певним чином перетинають відрізок. Ці пікселі в певному сенсі є найменш віддаленими від ідеального відрізка.

Оскільки координати точки відрізка з вершинами  $(x_1, y_1)$  та  $(x_2, y_2)$  можна однозначно ідентифікувати за допомогою рівняння

$$y = y_1 + k(x - x_1), \quad k = (y_2 - y_1) / (x_2 - x_1), \quad x_1 \leq x \leq x_2, \quad (1.4)$$

то базова схема роботи алгоритмів чотиризв'язної та восьмизв'язної розгортки відрізка за методом Брезенхема буде такою: обчислюють за (1.4) координати чергової поточної точки відрізка; послідовно ініціюють ті пікселі, що є чотиризв'язними (восьмизв'язними) сусідами цієї точки, залежно від типу розгортки; обирають той із ініційованих пікселів, який є найближчим до поточної точки.

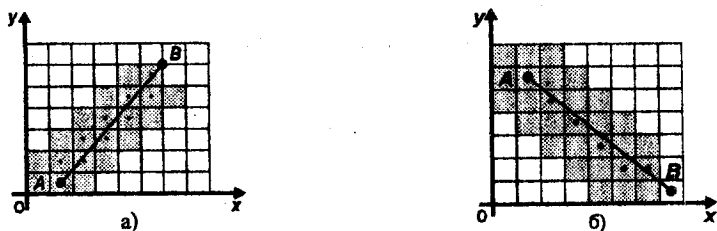


Рис. 1.2. Чотиризв'язна (а) та восьмизв'язна (б) растрові розгортки відрізка

*Розглянемо сутність растрової розгортки зафарбованих геометричних фігур та довільних суцільних областей.*

Базовими задачами растрової розгортки суцільних областей є: растрова розгортка (заповнення, заливка) внутрішньої

області многокутника, заданого опорними точками (відрізками); растрова розгортка (заповнення з затравкою, заливка) внутрішності області, обмеженої замкненим контуром, який подано за допомогою растрової розгортки [120, 174].

У процесі розв'язування задачі заповнення внутрішньої області многокутника  $S$ , що заданий набором своїх вершин або ребер та належить деякій області  $D$  растру, основним є визначення для кожної точки  $A_i \in D$ , якою вона є по відношенню до  $S$ : внутрішньою чи зовнішньою (виконання *тесту належності точки многокутнику*). Найчастіше для цього використовують такий прийом: деяку точку  $W_i$ , що не належить  $S$  (наприклад, будь-яку точку на межі області  $D$ ), з'єднують відрізком прямої лінії з точкою  $A_i$ ; обчислюють кількість  $K_i$  перетинів відрізка  $W_i A_i$  з негоризонтальними ребрами  $S$  (просте торкання  $W_i A_i$  до  $S$  у вершині не враховують) та аналізують отримане число на парність: якщо  $K_i$  – непарне число, то  $A_i \in S$ , інакше –  $A_i \notin S$  (рис. 1.3).

Оскільки метод прямого перебирання точок області  $D$  (у процесі їх тестування на належність многокутнику  $S$ ) є неефективним за обсягом обчислень та часу, впроваджують ряд вдосконалень. Наприклад, за область  $D$  обирають мінімальний прямокутник, який повністю містить  $S$  та сторони якого паралельні осям координат; рядок за рядком зверху донизу "сканують" область  $D$ , знаходячи в кожному рядку (за допомогою тесту належності) точки перетину  $W_i A_i$  з  $S$  та групуючи їх попарно (ці точки є кінцями інтервалів, які

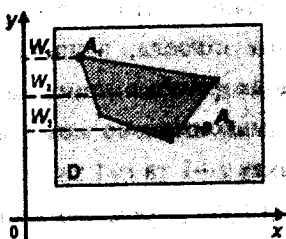


Рис. 1.3. Схема перевірки належності точки многокутнику

належать  $S$  та підлягають зафарбовуванню); упорядковують та відсіюють "неактивні" ребра  $S$  (ті ребра, що через певні умови не перетинають поточну "сканувальну" пряму).

Для розв'язування задачі "залівки області", тобто заповнення певним кольором чотириризв'язного або восьмиризв'язного компоненту області, межу якої задано як множину точок растру та для якої вказано одну з її внутрішніх точок (її називають "затравочною"), застосовують як попередні підходи, так і додаткові прийоми.

Розглянемо один із базових підходів, який використовує прийоми ведення стеку затравочних пікселів та порядкового сканування. Спочатку ініціюють стек затравочних пікселів, помістивши до нього початково заданий піксель області  $S$ , розташований у рядку з номером  $i$ . Потім у циклі, доки стек затравочних пікселів не стане порожнім, виконують такі дії: вилучають поточний затравочний піксель зі стеку; починаючи з цього пікселя, заповнюють праворуч та ліворуч від нього вздовж рядка з номером  $i$ , аж до досягнення граничних пікселів, максимально можливий інтервал, та фіксують його межі; у рядках  $i - 1$  та  $i + 1$  знаходять не заповнені внутрішні пікселі області  $S$ , які утворюють інтервали, а в кожному з цих інтервалів знаходять граничні праві пікселі та заносять їх до стеку.

**Параметричний метод задавання кривих ліній та криволінійних поверхонь, регулярні криві.** У процесі розв'язування задач комп'ютерної графіки, найбільш зручним вважають саме цей метод задавання кривих та поверхонь [41]. Розглянемо його для 3D-випадку (2D-варіант буде витікати з нього як частковий випадок).

**Параметрично задана в 3D-просторі крива** (параметризована, параметрична крива) – це множина точок 3D-прос-

тору, координати яких визначають за параметричними рівняннями кривої, тобто, в розгорнутій формі запису, за співвідношеннями:

$$x = x(t), \quad y = y(t), \quad z = z(t), \quad a \leq t \leq b \quad (1.5)$$

або, у векторній формі запису, за співвідношеннями:

$$r = r(t) = (x(t), y(t), z(t)), \quad a \leq t \leq b \quad (1.6)$$

де  $x(t)$ ,  $y(t)$ ,  $z(t)$  – функції параметру  $t$ , неперервні на відрізьку  $[a; b]$ .

Наприклад:

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x;$$

$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y;$$

$$z(t) = a_z t^3 + b_z t^2 + c_z t + d_z \quad (0 \leq t \leq 1).$$

Параметр  $t$  задає орієнтацію параметризованої кривої – порядок проходження точок у процесі монотонного змінювання параметру. Відрізок  $[a; b]$  на прямій параметру  $t$  зручно нормалізувати до відрізьку  $[0; 1]$  шляхом заміни  $t$  параметром  $u = (t - a) / (b - a)$ .

**Параметрично задана в 3D-просторі поверхня** (параметризована, параметрична поверхня) – це множина точок 3D-простору, координати яких визначають за параметричними рівняннями поверхні, тобто, в розгорнутій формі запису, за співвідношеннями:

$$x = x(u, v), \quad y = y(u, v), \quad z = z(u, v), \quad (u, v) \in D, \quad (1.7)$$

або, у векторній формі запису, за співвідношеннями:

$$r = r(u, v) = (x(u, v), y(u, v), z(u, v)), \quad (u, v) \in D, \quad (1.8)$$

де:  $u$  та  $v$  – параметри;  $D$  – деяка область на площині параметрів  $u, v$ ;  $x(u, v)$ ,  $y(u, v)$ ,  $z(u, v)$  – гладкі функції своїх аргументів.

Наприклад:

$$x(u, v) = a_{11}u^3v^3 + a_{12}u^3v^2 + a_{13}u^3v + a_{14}u^3 + \\ + a_{21}u^2v^3 + a_{22}u^2v^2 + a_{23}u^2v + a_{24}u^2 + a_{31}uv^3 + \\ + a_{32}uv^2 + a_{33}uv + a_{34}u + a_{41}v^3 + a_{42}v^2 + a_{43}v + a_{44}.$$

(для  $y(u, v)$  та  $z(u, v)$  формули подібні).

Для спрощення роботи з параметричними поверхнями, впроваджують область  $D$  у вигляді квадрату з одиничною стороною.

Розповсюдженням у комп'ютерній графіці класом кривих та поверхонь є "регулярні".

*Параметризовану криву  $\alpha$  називають регулярною кривою*, якщо в кожній точці  $\alpha$  існує дотична, що змінюється неперервно вслід за пересуванням уздовж  $\alpha$  її поточної точки, тобто  $\forall(t \in \alpha)$  справедлива нерівність  $r'(t) \neq 0$  (одичний вектор дотичної до кривої  $\alpha$  визначають зі співвідношення  $e_\alpha(t) = r'(t)/|r'(t)|$ ).

Вектор кривизни параметризованої регулярної кривої  $\alpha$ , векторна функція якої має другу похідну, визначають за формулою  $K_\alpha(t) = ([r'(t) \times r''(t)] \times r'(t))/|r'(t)|^4$ .

*Параметризовану поверхню  $\gamma$  називають регулярною*, якщо в кожній точці цієї поверхні існує дотична площина, що змінюється неперервно при неперервному пересуванні вздовж поверхні її поточної точки, тобто

$$\text{rang} \begin{pmatrix} x_u(u, v) & y_u(u, v) & z_u(u, v) \\ x_v(u, v) & y_v(u, v) & z_v(u, v) \end{pmatrix} = 2. \quad (1.9)$$

Розглянемо одну з найтипівіших задач роботи з регулярними кривими та поверхнями, результати розв'язку якої знадоб-

ляться для успішного сприйняття відомостей про сплайн-функції.

Нехай регулярні криві  $\alpha_1$  та  $\alpha_2$  з параметричними рівняннями  $r_{\alpha_1} = r_1(t)$ ,  $r_{\alpha_2} = r_2(t)$ , де  $0 \leq t \leq 1$  мають спільну точку  $r^*$ :

$$r^* = r_1(1) = r_2(0). \quad (1.10)$$

Потрібно визначити, за яких умов буде регулярною крива  $\alpha$  з векторним рівнянням  $r_\alpha = r(t)$  ( $0 \leq t \leq 1$ ), яка об'єднує  $\alpha_1$  та  $\alpha_2$ .

У процесі побудови складеної регулярної кривої  $\alpha$ , першочергову роль відіграють умови сполучення її відрізків  $\alpha_1$  та  $\alpha_2$ .

Цими умовами є, окрім (1.10), ще два критерії: вимога збігу в загальній точці одиничних дотичних векторів (1.11) та вимога збігу в загальній точці векторів кривизни кривих  $\alpha_1$  та  $\alpha_2$  (1.12).

$$r_1'(1)/|r_1'(1)| = r_2'(0)/|r_2'(0)|; \quad (1.11)$$

$$\frac{[r_1'(1) \times r_1''(1)] \times r_1'(1)}{|r_1'(1)|^4} = \frac{[r_2'(0) \times r_2''(0)] \times r_2'(0)}{|r_1'(0)|^4}. \quad (1.12)$$

Опустимо проміжні викладки та наведемо кінцевий розв'язок задачі: умови (1.10), (1.11) та (1.12) буде виконано, якщо радіус-вектори  $r_1(t)$  та  $r_2(t)$  будуть пов'язані такими умовами геометричної неперервності:

$$\begin{aligned} r_2(0) &= r_1(1), \quad r_2'(0) = \Theta_1 r_1'(1), \\ r_2''(0) &= \Theta_1^2 r_1''(1) + \Theta_2 r_1'(1), \quad \Theta_1 > 0, \quad \Theta_2 \geq 0. \end{aligned} \quad (1.13)$$



## Питання та завдання для самоперевірки

1. Назвіть системи та типи координат, які найчастіше використовують у комп'ютерній графіці.

2. Охарактеризуйте практичне призначення понять вікна, поля виведення, операцій видового перетворення та відсікання.

3. Сформулюйте сутність основних методів растрового подання геометричних об'єктів на екрані комп'ютеру.

4. Дайте визначення поняттю параметризованої поверхні.

5. Нехай  $A(x_A, y_A, z_A)$  – довільна точка  $3D$ -простору, координати якої є випадковими дійсними числами, визначеними в декартовій системі координат. Нехай також задані деякі афінна, полярна, циліндрична та сферична системи координат. Розрахуйте координати точки  $A$  в кожній із цих координатних систем.

6. Нехай у віртуальному просторі розташовано  $3D$ -сцену, яку складають куб, пряма тригранна призма та сфера. Задайте вікно, поле виведення та виконайте операції кадрування та відсікання.

7. Виконайте чотиризв'язну растрову розгортку довільного неопуклого чотирикутника, заданого набором вершин, та восьмизв'язну растрову розгортку зафарбованої фігури, обмежувальним контуром якої є самонеперетинна шестиланкова ламана лінія.

## 1.2. Моделювання композицій 2D/3D-перетворень

**Постановка задачі.** З проективної геометрії загальновідомо, що будь-яке складне геометричне перетворення  $S$ , яке змінює розташування, розміри та форму графічних об'єктів на 2D-площині або в 3D-просторі, можна подати у вигляді композиції елементарних афінних геометричних перетворень наступних трьох базових типів [41, 79, 120, 158]:

– **паралельне перенесення** (*translation*) на будь-який заданий вектор  $(\lambda, \mu)$  на 2D-площині або  $(\lambda, \mu, \nu)$  у 3D-просторі  $(\lambda, \mu, \nu$  – довільні дійсні числа);

– **обертання** (*rotation*) на заданий кут  $\alpha$   $(0 \leq \alpha < 2\pi)$  відносно початку координат на 2D-площині або відносно будь-якої з координатних осей (абсцис, ординат, аплікат) у 3D-просторі;

– **масштабування** (*dilatation*) – розтягування або стискання відносно початку координат з дійсними коефіцієнтами масштабування  $d_x > 0$  та  $d_y > 0$  уздовж координатних осей на 2D-площині, або  $d_x > 0, d_y > 0, d_z > 0$  у 3D-просторі.

Найчастіше складний графічний об'єкт розбивають на елементарні параметризовані примітиви (трикутники, чотирикутники тощо), які однозначно визначають їх опорні точки (вершини, центри ваги тощо) та опорні розміри, а геометричні перетворення застосовують до сукупності примітивів (доволі часто – до масиву опорних точок графічного об'єкту).

У процесі розв'язування задач побудови композицій геометричних перетворень, у комп'ютерній графіці важливу

роль відіграє *математичний апарат матричних перетворень та однорідних координат* [79, 158].

**Математичні моделі базових афінних геометричних перетворень на 2D-площині.** Будь-яке відбиття, що здійснює перехід від однієї точки  $A(x, y)$  у декартовій системі координат на 2D-площині до іншої  $A'(x', y')$ , або від однієї декартової системи координат до іншої, може бути описане за допомогою композиції формул, наведених у табл.1.1.

Таблиця 1.1

Математичний опис базових афінних геометричних  
2D-перетворень

Тип перетворення	Формули, що реалізують перетворення	
	матрична форма	розгорнута форма
1. Паралельне перенесення на вектор $(\lambda, \mu)$	$\begin{cases} (x', y') = (x, y) + T \\ T = (\lambda, \mu) \end{cases}$	$\begin{cases} x' = x + \lambda \\ y' = y + \mu \end{cases}$
2. Обертання на кут $\alpha$	$\begin{cases} (x', y') = (x, y) \times R \\ R = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} \end{cases}$	$\begin{cases} x' = x \cos \alpha - y \sin \alpha \\ y' = x \sin \alpha + y \cos \alpha \end{cases}$
3. Масштабування з коефіцієнтами $dx$ та $dy$	$\begin{cases} (x', y') = (x, y) \times D \\ D = \begin{pmatrix} dx & 0 \\ 0 & dy \end{pmatrix} \end{cases}$	$\begin{cases} x' = dx \times x \\ y' = dy \times y \end{cases}$

З табл.1.1 видно, що паралельне перенесення реалізоване в матричному поданні за допомогою операції додавання, а

обертання та масштабування – за допомогою операції множення. Це викликає ускладнення в процесі реалізації обчислень.

Вказаних ускладнень можна уникнути, якщо застосовувати однорідні координати.

У табл.1.2 показано, що введення в обчислення однорідних координат дозволяє реалізувати всі перетворення на базі операції множення матриць [174].

Таблиця 1.2

Математичний опис базових афінних геометричних  
2D-перетворень в однорідних координатах

Тип перетворення	Формули, що реалізують перетворення	
	матрична форма	розгорнута форма
1. Паралельне перенесення на вектор $(\lambda, \mu)$	$\begin{cases} (x', y', 1) = (x, y, 1) \times T \\ T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \lambda & \mu & 1 \end{pmatrix} \end{cases}$	$\begin{cases} x' = x + \lambda \\ y' = y + \mu \end{cases}$
2. Обертання на кут $\alpha$ відносно початку координат	$\begin{cases} (x', y', 1) = (x, y, 1) \times R \\ R = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{cases}$	$\begin{cases} x' = x \cos \alpha - y \sin \alpha \\ y' = x \sin \alpha + y \cos \alpha \end{cases}$
3. Масштабування з коефіцієнтами $dx$ та $dy$ відносно початку координат	$\begin{cases} (x', y', 1) = (x, y, 1) \times D \\ D = \begin{pmatrix} dx & 0 & 0 \\ 0 & dy & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{cases}$	$\begin{cases} x' = dx \times x \\ y' = dy \times y \end{cases}$
4. Довільна композиція перетворень (загальний вигляд)	$\begin{cases} (x', y', 1) = (x, y, 1) \cdot S \\ S = \begin{pmatrix} a_1 & a_2 & 0 \\ b_1 & b_2 & 0 \\ c_1 & c_2 & 1 \end{pmatrix} \end{cases}$	$\begin{cases} x' = a_1 x + b_1 y + c_1 \\ y' = a_2 x + b_2 y + c_2 \end{cases}$

Математичні моделі базових афінних геометричних перетворень у 3D-просторі. Аналогічно табл.1.2 здійснюють математичний опис базових афінних перетворень в однорідних координатах у 3D-просторі – це наочно демонструє зміст табл.1.3 [174].

Таблиця 1.3

Математичний опис базових афінних геометричних перетворень в однорідних координатах у 3D-просторі

Тип перетворення	Формули, що реалізують перетворення	
	матрична форма	розгорнута форма
1. Паралельне перенесення на заданий вектор $(\lambda, \mu, \nu)$	$\begin{cases} (x', y', z', 1) = (x, y, z, 1) \cdot T \\ T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \lambda & \mu & \nu & 1 \end{pmatrix} \end{cases}$	$\begin{cases} x' = x + \lambda \\ y' = y + \mu \\ z' = z + \nu \end{cases}$
2. Обертання на кут $\alpha_x$ відносно осі абсцис	$\begin{cases} (x', y', z', 1) = (x, y, z, 1) \cdot R_x^{(\alpha)} \\ R_x^{(\alpha)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_x & \sin \alpha_x & 0 \\ 0 & -\sin \alpha_x & \cos \alpha_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{cases}$	$\begin{cases} x' = x \\ y' = y \cos \alpha_x - z \sin \alpha_x \\ z' = y \sin \alpha_x + z \cos \alpha_x \end{cases}$
3. Обертання на кут $\alpha_y$ відносно осі ординат	$\begin{cases} (x', y', z', 1) = (x, y, z, 1) \cdot R_y^{(\alpha)} \\ R_y^{(\alpha)} = \begin{pmatrix} \cos \alpha_y & 0 & -\sin \alpha_y & 0 \\ 0 & 1 & 0 & 0 \\ \sin \alpha_y & 0 & \cos \alpha_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{cases}$	$\begin{cases} x' = x \cos \alpha_y + z \sin \alpha_y \\ y' = y \\ z' = -x \sin \alpha_y + z \cos \alpha_y \end{cases}$

Продовж. табл.1.3

Тип перетворення	Формули, що реалізують перетворення	
	матрична форма	розгорнута форма
4. Обертання на кут $\alpha_z$ відносно осі аплікату	$\begin{cases} (x', y', z', 1) = (x, y, z, 1) \cdot R_z^{(\alpha)} \\ R_z^{(\alpha)} = \begin{pmatrix} \cos\alpha_z & \sin\alpha_z & 0 & 0 \\ -\sin\alpha_z & \cos\alpha_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{cases}$	$\begin{cases} x' = x \cos \alpha_z - y \sin \alpha_z \\ y' = x \sin \alpha_z + y \cos \alpha_z \\ z' = z \end{cases}$
5. Масштабування відносно початку координат з коефіцієнтами $d_x, d_y, d_z$	$\begin{cases} (x', y', z', 1) = (x, y, z, 1) \cdot D \\ D = \begin{pmatrix} d_x & 0 & 0 & 0 \\ 0 & d_y & 0 & 0 \\ 0 & 0 & d_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{cases}$	$\begin{cases} x' = d_x \cdot x \\ y' = d_y \cdot y \\ z' = d_z \cdot z \end{cases}$
6. Довільна композиція перетворень (загальний вигляд)	$\begin{cases} (x', y', z', 1) = (x, y, z, 1) \cdot S \\ S = \begin{pmatrix} a_1 & a_2 & a_3 & 0 \\ b_1 & b_2 & b_3 & 0 \\ c_1 & c_2 & c_3 & 0 \\ d_1 & d_2 & d_3 & 1 \end{pmatrix} \end{cases}$	$\begin{cases} x' = a_1x + b_1y + c_1z + d_1 \\ y' = a_2x + b_2y + c_2z + d_2 \\ z' = a_3x + b_3y + c_3z + d_3 \end{cases}$

Основною відмінністю є те, що на 2D-площині базове перетворення обертання визначене відносно початку координат та реалізоване однією формулою, а в 3D-просторі воно визначене відносно кожної з трьох координатних осей та описане за допомогою трьох формул.

У плані ефективності обчислень, доцільним є вибір матричної форми подання тих формул, які реалізують базові афінні геометричні перетворення (сучасні об'єктно-орієнтовані

візуальні графічні середовища дозволяють ефективно працювати з матрицями, що визначені як класи та об'єкти).

**На практиці у переважній більшості випадків необхідно здійснювати композиції базових афінних геометричних перетворень.** Процес отримання математичної моделі складного геометричного перетворення містить декілька етапів [142, 174].

**Етап 1.** У процесі аналізу складного геометричного перетворення формують перелік та фіксують порядок здійснення тих базових афінних геометричних перетворень, композиція яких його складає.

**Етап 2.** Обчислюють конкретні значення параметрів (координат векторів перенесення, кутів обертання, коефіцієнтів масштабування), що необхідні для реалізації цих базових геометричних перетворень (часто етапи 1, 2 виконують паралельно).

**Етап 3.** Для кожного з виділених базових геометричних перетворень формують матрицю, що буде його реалізовувати, відповідно до табл.1.2-1.3.

**Етап 4.** Отримують матрицю  $S$  композиції визначених вище базових геометричних перетворень, виконуючи операцію знаходження добутку матриць, що були побудовані на етапі 3, у порядку їх побудови.

**Приклади розв'язування задач на композиції афінних геометричних перетворень в однорідних координатах.**

**Задача 1.** Побудуйте математичну модель геометричного перетворення обертання на кут  $\alpha$  навколо довільної точки  $C_0(x_0, y_0)$  для многокутника, заданого масивом вершин на 2D-площині.

Побудуємо математичну модель  $2D$ -обертання для узагальненої вершини, а потім застосуємо її для кожної вершини многокутника.

**Еман 1.** Для реалізації геометричного перетворення обертання на кут  $\alpha$  навколо довільної точки  $C_0(x_0, y_0)$  деякої узагальненої вершини  $K(x_k, y_k)$ , необхідно здійснити композицію: паралельного перенесення на вектор  $t^{(-)}(-x_0, -y_0)$  для суміщення центру обертання з початком координат; обертання навколо початку координат на кут  $\alpha$ ; паралельного перенесення на вектор  $t^{(+)}(x_0, y_0)$  з метою повернути центр обертання з початку координат до початкового положення  $C_0(x_0, y_0)$ .

**Еман 2.** Сформуємо для кожного з цих базових геометричних перетворень матриці, що їх реалізують, відповідно до табл.1.2:

матрицю паралельного перенесення на вектор  $(\pm x_0, \pm y_0)$

$$T_{(\pm x_0, \pm y_0)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \pm x_0 & \pm y_0 & 1 \end{pmatrix}. \quad (1.14)$$

та матрицю обертання навколо початку координат на кут  $\alpha$

$$R_0^{(\alpha)} = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (1.15)$$

**Еман 3.** Отримаємо матрицю  $S_{(x_0, y_0, \alpha)}$  композиції визначених вище базових афінних геометричних перетворень:

$$S_{(x_0, y_0, \alpha)} = T_{(-x_0, -y_0)} \cdot R_0^{(\alpha)} \cdot T_{(x_0, y_0)}; \quad (1.16)$$



$$S_{(x_0, y_0, \alpha)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_0 & -y_0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_0 & y_0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ -x_0 \cos\alpha + y_0 \sin\alpha + x_0 & -x_0 \sin\alpha - y_0 \cos\alpha + y_0 & 1 \end{pmatrix} \quad (1.17)$$

Тепер достатньо вектор-рядок координат  $(x_k, y_k)$  кожної вершини початкового многокутника домножити на матрицю (1.17), отримавши координати вершин оберненого багатокутника, та з'єднати нові вершини відрізками.

**Задача 2.** Побудуйте математичну модель складного геометричного 3D-перетворення обертання на кут  $\alpha$  навколо довільної прямої  $l$ , що проходить крізь точку  $C_0(x_0, y_0, z_0)$  та має визначений одиничний напрямний вектор  $n(n_x, n_y, n_z)$  (рис. 1.4).

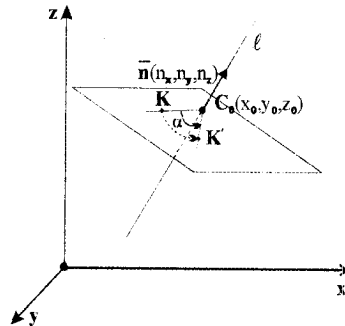


Рис. 1.4. Ілюстрація до постановки задачі 2

**Етап 1.** Для того, щоб

виконати обертання відносно довільної прямої, що, в найбільш загальному випадку, не проходить крізь початок координат, не співпадає з координатною віссю, не лежить в координатній площині та однозначно визначена за допомогою фіксованої точки та напрямного одиничного вектору  $(n_x^2 + n_y^2 + n_z^2 = 1)$ , необхідно здійснити композицію наступних базових афінних геометричних перетворень.

По-перше, це перетворення паралельного перенесення на вектор  $t^{(-)}(-x_0, -y_0, -z_0, 0)$ . Ця дія суміщає точку  $C_0(x_0, y_0, z_0, 1)$  з

точкою  $O(0, 0, 0, 1)$  для того, щоб пряма  $l$  пройшла крізь початок координат.

По-друге, це обертання осі  $z$  відносно початку координат на деякий кут  $\Theta$  для суміщення образу цієї осі  $z^*$  з прямою  $l$ .

Це обертання є складним та повинно бути розкладене на елементарні обертання навколо окремих координатних осей.

Наприклад, його можна подати як композицію таких двох обертань: а) обертання осі  $z$  навколо осі  $x$  на деякий кут  $\Theta_1$ , у межах площини  $zOy$  ( $x = 0$ ), для суміщення образу цієї осі з ортогональною проекцією  $z'$  прямої  $l$  на площину  $zOy$ ; б) обертання образу  $z'$  осі  $z$  навколо відповідного образу осі  $y$  на деякий кут  $\Theta_2$  для суміщення отриманого таким чином образу  $z^*$  осі  $z$  з прямою  $l$ .

Визначимо кут  $\Theta_1$ . Оскільки  $z'$  є ортогональною проекцією  $l$  на площину  $x = 0$ , а напрямний вектор  $l$  в однорідних координатах дорівнює  $(n_x, n_y, n_z, 1)$ , то напрямний вектор  $z'$  в однорідних координатах дорівнює  $(0, n_y, n_z, 1)$ . Звідси, з прямокутного трикутника  $OVQ$  з катетами  $n_y, n_z$  та гострим кутом  $\Theta_1$  при катеті  $n_z$  (рис 1.5), маємо:

$$\sin \Theta_1 = \frac{n_y}{d}; \quad \cos \Theta_1 = \frac{n_z}{d}; \quad d = \sqrt{n_y^2 + n_z^2}. \quad (1.18)$$

Визначимо кут  $\Theta_2$ . Враховуючи результати визначення кута  $\Theta_1$ , із прямокутного трикутника  $OQR$  з катетами  $n_x$  та  $d$ , гіпотенузою одиничної довжини та гострим кутом  $\Theta_2$  при катеті  $d$  (рис.1.5), маємо:  $\cos \Theta_2 = d$ ,  $\sin \Theta_2 = n_x$ .

По-третє, це обертання навколо образу осі аплікат  $z^*$ , що співпадає з прямою  $l$ , на кут  $\alpha$ .

По-четверте, це композиція обертань навколо осі  $y$  на кут  $-\Theta_2$  та навколо осі  $x$  на кут  $-\Theta_1$  для повертання осі аплікат до початкової позиції.

По-п'яте, це перенесення на вектор  $t^{(+)}(-x_0, -y_0, -z_0, 1)$  для повертання точки  $C_0$  до початкової позиції.

**Етап 2.** Запишемо матриці цих перетворень. В процесі множення матриць повинен бути дотриманий правильний порядок перетворень обертання:

$$\begin{aligned} S &= T_{(-x_0, -y_0, -z_0)} \cdot R_x^{(\Theta_1)} \cdot R_y^{(\Theta_2)} \cdot R_z^{(\alpha)} \cdot R_y^{(-\Theta_2)} \cdot R_x^{(-\Theta_1)} \cdot T_{(x_0, y_0, z_0)} = \\ &= T_{(-x_0, -y_0, -z_0)} \cdot R^{(\Theta)} \cdot R_z^{(\alpha)} \cdot R^{(-\Theta)} \cdot T_{(x_0, y_0, z_0)}. \end{aligned} \quad (1.19)$$

Запишемо конкретні значення тих матриць, які є складовими частинами формули (1.19).

$$T_{(\pm x_0, \pm y_0, \pm z_0)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \pm x_0 & \pm y_0 & \pm z_0 & 1 \end{pmatrix}, \quad (1.20)$$

$$R_x^{(\Theta_1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & n_z/d & n_y/d & 0 \\ 0 & -n_y/d & n_z/d & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad R_y^{(\Theta_2)} = \begin{pmatrix} d & 0 & -n_x & 0 \\ 0 & 1 & 0 & 0 \\ n_x & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (1.21)$$

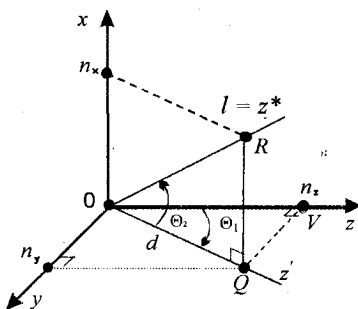


Рис. 1.5. Ілюстрація до розв'язку задачі 2

$$R^{(\Theta)} = R_x^{(\Theta_1)} * R_y^{(\Theta_2)} = \begin{pmatrix} d & 0 & -n_x & 0 \\ n_x n_y / d & n_z / d & n_y & 0 \\ n_x n_z / d & -n_y / d & n_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (1.22)$$

$$R_z^{(\alpha)} = \begin{pmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ -\sin \alpha & \cos \alpha & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad (1.23)$$

$$R^{(-\Theta)} = R_y^{(-\Theta_2)} * R_x^{(-\Theta_1)} = \begin{pmatrix} d & n_x n_y / d & n_x n_z / d & 0 \\ 0 & n_z / d & -n_y / d & 0 \\ -n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (1.24)$$

При отриманні  $R_y^{(-\Theta_2)}$  та  $R_x^{(-\Theta_1)}$  враховувалися відомі тригонометричні співвідношення  $\cos(-\Theta) = \cos(\Theta)$  та  $\sin(-\Theta) = -\sin(\Theta)$ .

**Еман 3.** Надаємо вам можливість самостійно виписати композицію матриць (1.20) – (1.24) відповідно до (1.19), підставити конкретні значення  $x_0, y_0, z_0, n_x, n_y, n_z, d, \alpha$  та отримати результат.

## Питання та завдання для самоперевірки

1. Назвіть основні афінні геометричні перетворення на площині та в просторі, сформулюйте їх особливості.

2. Як можна подати довільне складне геометричне перетворення, що веде до зміни розташування та розмірів

2D/3D-об'єктів, за допомогою елементарних афінних геометричних перетворень?

3. Яку роль відіграють матричний запис та однорідні координати в процесі застосування афінних геометричних перетворень у комп'ютерній графіці?

4. Якою є типова схема розв'язування задач на виконання складних геометричних 2D/3D-перетворень, що є композицією елементарних афінних геометричних перетворень?

5. Застосуйте до одиничного кубу такі перетворення: зсуву вздовж осі  $O_z$ ; масштабування вздовж осі  $O_y$ ; обертання відносно осі  $O_x$ .

### 1.3. Моделювання проєкцій

**Постановка задачі.** Виведення тривимірної або  $n$ -вимірної ( $n \geq 3$ ) графічної інформації на двовимірний екран комп'ютеру вимагає виконання таких операцій:

- моделювання багатовимірної геометричної сцени в координатах, визначених користувачем;
- отримання примітивів виведення, визначених у багатовимірних світових координатах;
- виконання операції відсікання за видимим об'ємом для отримання зрізаних світових координат;
- проектування геометричної сцени на площину проєкцій;
- перетворення отриманої проєкції до поля виведення, що задане в нормованих координатах;

– перетворення нормованих координат проєкцій об'єктів багатовимірної геометричної сцени до фізичних координат точок цих проєкцій на екрані дисплею.

Головну роль відіграє проєктування заданого фрагменту геометричної сцени на площину проєкцій. Воно дозволяє відобразити точки, задані в системі координат розмірністю  $n$ , у точки системи координат розмірністю  $n - 1$  (наприклад, три виміри відобразити в два).

*Апарат отримання проєкцій* містить чотири базових компоненти: центр проєкцій; площину або поверхню проєкцій (картинну площину); зображуваний об'єкт, який подають, як правило, сукупністю точок; пучок проєктувальних прямих (проєкторів, променів), які проходять крізь центр проєкцій та точки зображуваного об'єкту, перетинаючи картинну площину.

*Основними класами геометричних перетворень проєктування в комп'ютерній графіці є: паралельні проєкції; центральні (перспективні) проєкції* [37, 64, 79].

#### *Паралельна проєкція:*

– дозволяє безпосередньо вимірювати лінійні розміри, паралельні координатним осям, оскільки викривлення вздовж осей є постійним;

– поступається в реалістичності центральній проєкції через відсутність ефекту перспективного вкорочування [39, 41, 64, 79].

#### *Центральна (перспективна) проєкція:*

– забезпечує найбільшу реалістичність завдяки природному (для ока людини) ефекту перспективного вкорочування лінійних розмірів, які змінюються зворотно пропорційно збільшенню відстані від центру проєкцій до об'єкту;

– непридатна для безпосереднього визначення лінійних розмірів, площ та кутів через ефект їх перспективного вкорочування [39, 41, 64, 79].

**Паралельне проектування.** У цьому випадку центр проектування є невласною точкою (знаходиться на безмежно далекій відстані від геометричної сцени), а проектори є паралельними прямими лініями. Розглянемо математичні моделі базових підкласів геометричних перетворень паралельного проектування в комп'ютерній графіці [64, 79].

**Прямокутна ортогографічна проекція** характеризується тим, що: проектори перпендикулярні до площини проєкцій (напрямок проектування та напрямком нормалі до картинної площини співпадають); площина проєкцій або співпадає з координатною площиною, або паралельна до неї (основну матрицю проектування помножують на матрицю зсуву площини проєкцій відносно координатної площини). Виходячи з цього:

$$O_x = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \quad O_x^{(s)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ s_x & 0 & 0 & 1 \end{pmatrix}; \quad (1.25)$$

$$O_x^{(p)} = O_x \cdot O_x^{(s)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ s_x & 0 & 0 & 1 \end{pmatrix}. \quad (1.26)$$

де  $O_x$  – матриця проектування вздовж осі  $Ox$  на площину  $yOz$ ;  $O_x^{(s)}$  – матриця урахування зсуву площини проєкцій уздовж осі  $Ox$ ;  $O_x^{(p)}$  – матриця проектування вздовж осі  $Ox$  на площину, що є паралельною до  $yOz$ . По аналогії обчислюють значення матриць  $O_y^{(p)}$  та  $O_z^{(p)}$ .

*Прямокутна аксонометрична проекція* характеризується тим, що: проектори перпендикулярні площині проєкцій (напрямок проєктування та напрямок нормалі до картинної площини співпадають); площина проєкцій утворює певні кути з координатними площинами (осями).

Ця проєкція може бути подана як композиція: двох обертань навколо координатних осей для суміщення площини проєкцій та однієї з координатних площин (або площини, паралельної координатній); ортографічного проєктування на вказану площину.

Залежно від кількості  $N$  однакових кутів між нормаллю до площини проєкцій та ортами координатних осей, прямокутні аксонометричні проєкції розділяють на: ізометрію ( $N = 3$ ), диметрію ( $N = 2$ ), триметрію ( $N = 0$ ).

Для прямокутної аксонометричної проєкції:

$$A(\alpha_x, \alpha_y) = R_y^{(\alpha_y)} \cdot R_x^{(\alpha_x)} \cdot O_z = \begin{pmatrix} \cos\alpha_y & \sin\alpha_x \cdot \sin\alpha_y & 0 & 0 \\ 0 & \cos\alpha_x & 0 & 0 \\ -\sin\alpha_y & -\sin\alpha_x \cdot \cos\alpha_y & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (1.27)$$

де  $A(\alpha_x, \alpha_y)$  – матриця аксонометричного проєктування на площину проєкцій, нормаль якої утворює з координатними осями  $Ox$  та  $Oy$  кути  $\alpha_x$  та  $\alpha_y$ .

$A(\alpha_x, \alpha_y)$  є добутком: матриці  $R_y^{(\alpha_y)}$  обертання на кут  $\alpha_y$  відносно осі  $y$ , матриці  $R_x^{(\alpha_x)}$  обертання на кут  $\alpha_x$  відносно осі  $x$  та матриці ортогонального проєктування без зсуву  $O_z$  уздовж осі  $z$  на площину  $z = 0$ . Аналогічно визначають матриці  $A(\alpha_x, \alpha_z)$  та  $A(\alpha_y, \alpha_z)$ .

Виходячи з цього, для диметрії виконується співвідношення  $\cos^2\alpha_y + \sin^2\alpha_x \sin^2\alpha_y = \cos^2\alpha_x$ , а для ізометрії виконується спів-



відношення  $\sin^2\alpha_y + \sin^2\alpha_x \cdot \cos^2\alpha_y = \cos^2\alpha_x$  ( $\sin^2\alpha_y = 1/2$ ,  $\sin^2\alpha_x = 1/3$ ). Матричні операції, з яких це витікає, такі:  $e_x \cdot A(\alpha_x, \alpha_y) = (\cos\alpha_y, \sin\alpha_x \cdot \sin\alpha_y, 0, 1)$ ;  $e_y \cdot A(\alpha_x, \alpha_y) = (0, \cos\alpha_x, 0, 1)$ ;  $e_z \cdot A(\alpha_x, \alpha_y) = (\sin\alpha_y, -\sin\alpha_x \cdot \cos\alpha_y, 0, 1)$ .

Аналогічні дії виконують з матрицями  $A(\alpha_x, \alpha_z)$  та  $A(\alpha_y, \alpha_z)$ .

**Косокутна проекція** характеризується тим, що: проектори не перпендикулярні площині проєкцій (напрямок проєкування та напрямок нормалі до картинної площини не співпадають).

Косокутна проекція має два характерних різновиди:

– **вільна косокутна проекція** (кут нахилу проєкторів до площини проєкцій дорівнює  $\pi/4$ );

– **кабінетна косокутна проекція** (підвид вільної косокутної проєкції, для якої масштаб уздовж активної координатної осі є вдвоє меншим, ніж уздовж інших двох осей).

Якщо ввести позначення  $a = n \cos\alpha$  та  $b = n \sin\alpha$ , де  $\alpha$  – кут нахилу проєкторів до площини проєкцій (кут між напрямком проєкування та площиною проєкцій),  $n$  – масштаб уздовж активної координатної осі, то:

$$K_{xOy} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a & b & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \quad (1.28)$$

$$e'_z = e_z \cdot K_{xOy} = (0 \ 0 \ 1 \ 1) \cdot K_{xOy} = (a \ b \ 0 \ 1), \quad (1.29)$$

де  $K_{xOy}$  – матриця косокутного проєкування на площину  $xOy$  ( $z = 0$ );  $e_z, e'_z$  – орт осі  $z$  та його проєкція на площину  $xOy$ .

По аналогії обчислюють  $K_{xOz}, K_{yOz}, e'_y$  та  $e'_x$ .

Для вільної проєкції  $n = 1$  та  $a = b = \cos(\pi/4)$ .

Для кабінетної проєкції  $n = 1/2$  та  $a = b = 1/2 \cos(\pi/4)$ .

**Перспективне (центральне) проектування.** Цей вид проектування характеризує те, що проєкції пучків паралельних прямих сходяться в одній точці, яку називають *точкою сходимості*.

Якщо пучок паралельних прямих є паралельним ще й до однієї з головних координатних осей, то відповідну точку сходимості називають *головною точкою сходимості*.

Головних точок сходимості може бути від однієї до трьох: вони відповідають точкам перетину координатних осей з картинною площиною.

Залежно від кількості головних точок сходимості, центральні проєкції поділяють так:

– *фронтальна (однофокусна) перспектива*, коли картинна площина паралельна двом координатним осям, тобто одній із координатних площин;

– *двофокусна перспектива*, коли картинна площина паралельна одній координатній осі (є найбільш розповсюдженим класом перспективних проєкцій у будівництві, архітектурі та дизайні);

– *трифокусна перспектива*, коли картинна площина перетинає всі координатні осі (через складність побудови, її застосовують рідко, наприклад, для зображення висотних будівель та споруд) [64, 79].

У найбільш загальному та складному випадку трифокусної перспективи, перетворення буде визначене формулою

$$(x' y' z' 1) = (x y z 1) * S^{(3)}; S^{(3)} = \begin{pmatrix} 1 & 0 & 0 & -1/s_x \\ 0 & 1 & 0 & -1/s_y \\ 0 & 0 & 1 & -1/s_z \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (1.30)$$

де:  $s_x, s_y, s_z$  – координати центру проектування.

Проаналізуємо витoki розглянутих вище формул на прикладі простої та наочної задачі моделювання однофокусної перспективної проєкції з такими початковими даними (рис. 1.6): площина проєкцій співпадає з координатною площиною  $XOY$  ( $z = 0$ ); центр проєкування  $S$  лежить на координатній осі, перпендикулярній площині проєкцій.

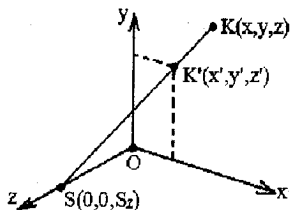


Рис. 1.6. Початкові дані до задачі отримання однофокусної перспективи

Параметричні рівняння проєктору, що проходить крізь деяку довільну точку  $K(x, y, z)$ , мають вигляд  $x' = xt, y' = yt, z' = s_z - (z - s_z)t$ .

Оскільки для точки  $K'$  виконується рівність  $z' = 0$ , то:

$$t = \frac{1}{1 - \frac{z}{s_z}}, \quad x' = \frac{1}{1 - \frac{z}{s_z}} x, \quad y' = \frac{1}{1 - \frac{z}{s_z}} y, \quad z' = 0. \quad (1.31)$$

Тобто центральне проєкування буде визначене формулою (1.32), а матриця перспективного перетворення без проєкування – формулою (1.33).

$$(x' \ y' \ z' \ 1) = (x \ y \ z \ 1) \cdot S^{(0)}; \quad S^{(0)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/s_z \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (1.32)$$

$$P^{(0)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1/s_z \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (1.33)$$

## Питання та завдання для самоперевірки

1. Якою є типова схема виведення багатовимірної графічної інформації на  $2D$ -екран графічного дисплею?

2. Наведіть класифікацію геометричних перетворень проектування, що найчастіше використовуються в комп'ютерній графіці, та сформулюйте їх особливості.

3. Застосуйте до одиничного кубу такі перетворення: паралельного проектування на площину  $XOY$  вздовж осі  $Oz$ ; перспективного проектування на площину  $XOY$  з центром проектування в довільній точці  $3D$ -простору.

### 1.4. Моделювання стереоскопічних зображень

*Ефект об'ємності зображення (стереоскопічний ефект)* виникає у людини завдяки *бінокулярності* зору, а саме – наявності *очного базису* (відстані між центрами зіниць очей), що приводить до сприйняття об'єктів з двох різних точок зору.

Два перспективних зображення, сформовані окремо для правого та лівого ока, називають *стереопарою*.

Для того, щоб у процесі перегляду стереопари виникло відчуття об'ємності зображення, необхідно виконати операцію *сепарації стереопари* – розділити два перспективних зображення так, щоб кожне око сприймало своє зображення [41, 64, 79].

Методи побудови систем формування об'ємних зображень поділяють на такі класи: стереотелевізійні; на базі електронно-оптичних систем з об'ємними або багатошаровими елементами; голографічні.

*У разі стереотелевізійних методів*, об'ємні зображення за допомогою стереопари реалізують на одному або двох телевізійних екранах.

Стереотелевізійні методи класифікують: за способом передавання стереопар – паралельним або послідовним (на один або декілька екранів); за способом сепарації стереопар; за кількістю стереопар, потрібною для створення об'ємного зображення.

Найчастіше використовують *окулярну сепарацію* (зображення розділяють безпосередньо біля очей), базовими видами якої є:

– світло-фільтровий окулярний (відтворення стереопар, що пофарбовані в різні кольори, та здійснення сепарації кольоровими окулярами з різними світлофільтрами);

– поляризаційний окулярний (застосування ефекту поляризації світла, за якого кожне зображення пропускають крізь поляризаційні оптичні фільтри);

– екліпсний окулярний (до окулярів спостерігача вмонтовують спеціальні механічні або оптико-електронні системи, що дозволяють здійснювати почергове переривання випромінювань, які надходять до правого та лівого ока);

– стереоскопічний окулярний (застосування персональних стереоскопів, в яких сепарацію здійснюють за допомогою лінзово-призмових оптичних пристроїв).

*Безокулярну сепарацію* реалізують переважно за допомогою растрових телевізійних екранів, які складаються з тонких циліндричних лінз.

*За умови багаторакурсного способу* відтворення зображення, що отримане з різних точок зору, застосовують виключно методи екранної сепарації:

– чергування зон стереоскопічного бачення здійснюється перед растровим екраном;

– пересуваючись усередині зони стереобачення, різні спостерігачі можуть розглядати об'ємне зображення, змінюючи точку зору на об'єкт.

*У разі голографічного методу*, використовують явище інтерференції (дві світлові хвилі, зустрічаючись у певній точці об'єкту, залежно від їх амплітуди, створюють або більш інтенсивне світло, або затемнення).

Для реалізації голографічного методу, сцену освітлюють когерентним світлом з хвилями однакової фази (наприклад, за допомогою лазера).

До фотоплатівки потрапляють два пучки світла: предметний пучок світла, відбитий від об'єкту; опорний пучок світла, що йде безпосередньо від джерела, обминаючи об'єкт. Голограму створює зареєстрована картина інтерференції предметного та опорного пучків світла.

## **Питання для самоперевірки**

1. Дайте визначення базовим поняттям, пов'язаним із побудовою об'ємних (стереоскопічних) зображень.

2. Наведіть перелік основних класів методів побудови систем формування стереоскопічних зображень.
3. Назвіть базові види окулярної сепарації стереопар.
4. За допомогою яких пристроїв реалізують безокулярну сепарацію стереопар?
5. Сформулюйте сутність голографічного методу.

## Навчальний практикум до гл.1

Нижче наведено перелік задач, який містить як завдання однакового рівня складності, так і завдання для обдарованих студентів. У процесі виконання цих завдань необхідно: побудувати концептуальну та математичну моделі розв'язування задачі; обґрунтувати вибір структур даних, апаратного та програмного інструментарію, технологій створення ПЗ; розробити відповідні алгоритми та програмний код, контрольні приклади до нього.

**1-а група завдань.** Розробіть двовимірний та тривимірний графічні редактори наступної спрямованості: а) ділової графіки (рекомендовані прототипи: *Excel*, *FoxGraph*, *SuperCalc*, *QuattroPro*, *Lotus 1-2-3*); б) наукової графіки (рекомендовані прототипи: *MathCAD*, *Mathematica*, *Statistica*, *SPSS*); в) інженерної графіки (рекомендовані прототипи: *AutoCAD*, *КОМПАС*, *ArchiCAD*, *OrCAD*, *PCAD*); г) ілюстраційної графіки (рекомендовані прототипи: *Illustrator*, *CorelDraw*, *PhotoShop*).

При цьому потрібно реалізувати не весь набір вбудованих графічних об'єктів та функцій прототипу, а лише обмежену множину базових графічних примітивів та операцій для роботи з ними, які вам порекомендує викладач.

**2-а група завдань.** Розробіть програму-тренажер, яка відпрацьовує такі навички: а) керування літаком; б) водіння автомобіля або іншого наземного транспортного засобу; в) спортивної стрільби; г) роботи з клавіатурою комп'ютеру; д) гри у шахи (шашки, доміно тощо); е) складання деталей конструктору; ж) інші типи тренажерів.

Як і в попередній групі завдань, передбачена робота з певним набором графічних  $2D/3D$ -об'єктів та їх геометричних перетворень, а конкретний склад цього набору вам порекомендує викладач.

**3-я група завдань.** Створіть ПЗ для реалізації на комп'ютері таких перетворень: а) обертання відносно заданого нерухомого/рухомого центру; б) обертання відносно заданої нерухомої/рухомої прямої лінії; в) об'льоту заданої нерухомої/рухомої траєкторії (наприклад, гвинтової лінії), та масштабування в процесі руху.

Застосуйте ці перетворення до одного з наведених далі класів геометричних об'єктів: а) одного з каркасних та одного з твердих платонових  $3D$ -тіл (правильних багатогранників – куба, тетраедра, октаедра, ікосаедра, додекаедра); б) каркасного та твердого тривимірних тіл обертання (тора, сфери, конусу, циліндру тощо); в) каркасного та твердого довільних  $3D$ -тіл; г) декількох зчеплених між собою каркасних та твердих довільних  $3D$ -тіл; д) ансамблю з декількох твердих та каркасних  $3D$ -тіл, кожне з яких пересувається власною фіксованою траєкторією.

Центр обертання, вісь обертання та траєкторія можуть бути довільно розташовані відносно  $3D$ -тіла або ансамблю  $3D$ -тіл. Наприклад, центр обертання може знаходитись як усередині,



так і зовні 3D-тіла (ансамбля), а вісь обертання – перетинати або не перетинати його.

**4-а група завдань.** Розробіть та реалізуйте систему комп'ютерного 3D-проекування, що містить один із наступних наборів елементарних логічних операцій над 3D-об'єктами (платоновими тілами, фігурами обертання, довільними геометричними 3D-об'єктами тощо): а) логічні перетин та об'єднання; б) логічні віднімання та доповнення. Ця система повинна демонструвати свою роботу в динаміці процесу проектування.

**5-а група завдань.** Розробіть ПЗ для побудови одного з базових видів: а) паралельних проєкцій 3D-об'єктів; б) перспективних проєкцій 3D-об'єктів.

**6-а група завдань.** Розробіть ПЗ для побудови 3D-проєкцій чотирирівмірних геометричних тіл: а) 4D-куба; б) 4D-повноторія.

## ГЛАВА 2. МОДЕЛЮВАННЯ РЕАЛІСТИЧНИХ ЗОБРАЖЕНЬ

### 2.1. Моделювання видимих ділянок об'єктів

**Постановка задачі.** Видалення (блокування виведення на екран) невидимих ділянок ліній та поверхонь (*задача загороджування*) є одним із найважливіших та найважчих етапів створення на комп'ютері якісного та реалістичного зображення 3D-об'єктів та 3D-сцен за короткий час [41, 63, 64, 79].

З проєктивної геометрії відомо, що деяка точка  $P_1$  поверхні  $S$  *загороджує* точку  $P_2$  цієї поверхні, якщо: проєкції цих точок на картинну площину  $T$  при заданому центрі проєктування та спостереження  $C$  співпадають;  $|CP_1| < |CP_2|$  (рис.2.1). Деяку точку поверхні називають *видимою*, якщо вона не загорожена жодною іншою точкою цієї поверхні або інших об'єктів 3D-сцени.

*Задача видалення невидимих ділянок ліній та поверхонь природно зводиться до виявлення видимих частин цих ліній та поверхонь та зображення їх на картинній площині.* Незважаючи на існування численних загальних концепцій розв'язування задач загороджування, відсутній універсальний алгоритм їх розв'язування. Натомість є ефективні методи розв'язування окремих класів задач загороджування [64, 79, 120].

**Загальні концепції розв'язування задач загороджування.** Нехай геометрична 3D-сцена складається з певної кількості фрагментів (об'єктів)  $A_1, A_2, \dots, A_N$  (рис.2.2). Тоді для будь-якої пари фрагментів  $A_i$  та  $A_j$  потрібно встановлювати такі факти:

якщо жоден із фрагментів  $A_i$  та  $A_j$  не загороджує інший, то їх можна обробляти незалежно один від одного; якщо фрагмент  $A_i$  повністю або частково загороджує фрагмент  $A_j$ , а фрагмент  $A_j$  не загороджує фрагменту  $A_i$ , то встановлюють певний пріоритет їх обробки (спочатку – фрагмент  $A_j$ , а потім – фрагмент  $A_i$ ); якщо фрагменти  $A_i$  та  $A_j$  частково загороджують один одного, то доцільно розбити ці фрагменти на складові частини та повернутися до першого або другого варіантів.

У цілому, задача встановлювання факту загороджування

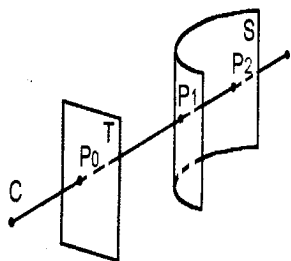


Рис.2.1. Ілюстрація до постановки задачі загороджування

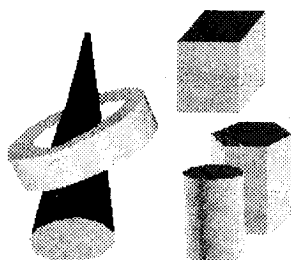


Рис.2.2. Приклад геометричної 3D-сцени з загородженими об'єктами

одним об'єктом іншого об'єкту може бути розв'язана двома способами: у *просторі об'єктів* (шляхом аналізу взаємного розташування геометричних фігур); у *просторі проєкцій* (шляхом аналізу взаємного розташування проєкцій геометричних фігур на картинну площину).

У разі аналізу взаємного розташування 3D-фігур, пріоритет їх обробки встановлюють так: знаходять найдальші та найближчі точки фрагментів по відношенню до спостерігача (або до картинної площини) та аналізують їх розташування (якщо найдальша точка фрагменту  $A_i$  лежить ближче до спостерігача, ніж найближча точка фрагменту  $A_j$ , то  $A_j$  не

загороджує  $A_i$ , в об'єктному просторі знаходять площину, що розділяє його на два напівпростори, в одному з яких знаходяться фрагмент  $A_i$  та спостерігач, а в іншому – фрагмент  $A_j$  (тоді  $A_j$  не може загороджувати  $A_i$ ).

У разі аналізу проєкцій, незалежна обробка геометричних фігур можлива тоді, коли проєкції цих фігур не перетинаються.

*Для ефективного розв'язування задач загороджування, правила обчислення пріоритету та встановлення факту незалежності фрагментів 3D-сцени повинні бути простими.* Тому початкову задачу загороджування необхідно подрібнювати та спрощувати до такого стану, щоб її можна було розв'язати на основі простих правил [79, 174].

Якщо початкові геометричні фігури мають складну форму, доцільно розбити їх на простіші складові частини або помістити, як в оболонку, всередину простіших геометричних фігур (кулі, куба, паралелепіпеда), та працювати з цими оболонками (вони повинні бути мінімальними). Якщо проєкції мають складні форми, їх теж розбивають на простіші складові частини або *поміщають до мінімальних простих оболонок* (до прямокутників зі сторонами, паралельними осям координат).

**Класифікація методів розв'язування задач загороджування.** Базовими для цієї класифікації є такі поняття: *дозвіл картинної площини* (кількість точок растру або пікселів на екрані монітору); *дозвіл многогранної поверхні* (кількість її граней); *глибина елемента поверхні або лінії* (відстань від нього до картинної площини). Найвідоміші методи розв'язування задач загороджування можна класифікувати за чотирма наведеними нижче характеристиками.

По-перше, за вибраним типом формалізованого опису досліджуваної поверхні або лінії (за тим математичним апаратом та структурами даних, які було вибрано для її подання), виділяють три класи методів, які використовують такі подання: *аналітичне* (за допомогою аналітичного виразу); *поліедральне*, тобто многогранне (за допомогою сукупності многокутних граней); *параметричне*.

По-друге, за вибраним типом простору, в якому здійснюють аналіз видимості частин досліджуваних поверхонь або ліній, виділяють такі класи методів: *об'єктні* (аналіз видимості здійснюють в об'єктному просторі, час роботи методів є квадратичною функцією від кількості об'єктів 3D-сцени та їх дозволу); *картинні* (аналіз видимості здійснюється в площині зображення, час роботи методів є лінійною функцією від добутку кількості об'єктів на кількість точок растру); *змішані*.

По-третє, за обраним способом візуалізації поверхні виділяють такі класи методів: ті, що орієнтовані на отримання каркасного зображення (лінії сітки параметричного подання та ребра многогранника називають каркасними лініями, а саме зображення – каркасним – рис.2.3); ті, що орієнтовані на отримання напівтонового зображення (з використанням напівтонового зафарбовування елементів поверхні або накладання на неї текстури, з урахуванням впливу на цю текстуру джерел світла – рис.2.4).

По-четверте, методи розв'язування задач загороджування розрізняють за специфічними геометричними властивостями 3D-об'єктів, 3D-сцен та їх проєкцій, використаними в процесі роботи.

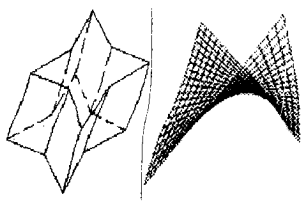


Рис.2.3. Приклади каркасних зображень

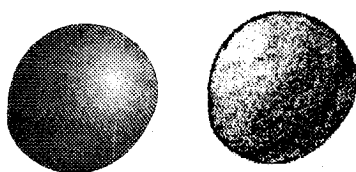


Рис.2.4. Приклади напівтонових зображень

Застосування до розв'язування задачі загороджування тих чи інших методів суттєво залежить від форми, властивостей та взаємного розташування компонентів  $3D$ -сцен. Слід окремо виділити опуклі многогранники та гладкі несамоперетинні поверхні: завдяки їх гарній структурованості та впорядкованості, видимість кожного елементу  $3D$ -сцени залежить від видимості найближчих елементів та змінюється регулярно (кожна грань опуклого многогранника або повністю видима, або повністю невидима). У разі неопуклих многогранних поверхонь, видимість теж змінюється регулярно. Але цих властивостей не буде мати  $3D$ -сцена з довільних, неупорядковано розкиданих у просторі многокутників.

Важливою характеристикою методу розв'язування задачі загороджування є асимптотичний час роботи алгоритму залежно від дозволу зображуваного об'єкту та картинної площини (він може бути пов'язаний з властивостями об'єкту, внутрішньою структурою використаного алгоритму та іншими факторами) [79, 120, 174].

Найбільш розповсюдженими на практиці методами розв'язування задач загороджування є такі методи: прямого перебирання, буферу глибини ( $Z$ -буферу), видалення неліцьо-

вих граней, аналізу кількісної невидимості точок поверхні, розбиття картинної площини за допомогою активних вікон, кеширування вздовж регулярної прямокутної сітки, художника, поточних обріїв, двійкового розбиття об'єктного простору, порядкового сканування.

**Оглядовий аналіз базових методів розв'язування задач загороджування.** Розглянемо сутність, характеристику практичного застосування та продуктивності основних класів методів розв'язування задач загороджування [79, 120, 174].

### **1. Метод прямого перебирання.**

*Сутність методу.* Аналіз видимості об'єктів 3D-сцени передбачає пряме перебирання цих об'єктів. Розглядають кожне ребро кожного об'єкту сцени та аналізують його взаємне розташування з усіма гранями.

Якщо ребро перетинає ту напівплощину, що містить грань, то розглядають окремо частини ребра, розділені точкою перетину.

Якщо ж ребро не перетинає грань, то перевіряють, чи знаходиться це ребро в тому ж напівпросторі відносно грані, що й спостерігач.

*Характеристика методу.* Застосовується для побудови каркасних зображень. Час роботи цього методу квадратично залежить від кількості елементів сцени (дозволу об'єктів), оскільки є пропорційним добутку кількості ребер та граней сцени.

### **2. Метод буферу глибини (Z-буферу).**

*Сутність методу.* Використовуються дві області пам'яті:

– Z-буфер (він же – буфер глибини) зберігає координату Z (глибину) кожного пікселя картинної площини, що є видимим на поточному кроці аналізу зображення;

– буфер кадру містить інформацію про стан атрибутів пікселів (інтенсивності та кольору).

Технологія роботи за методом Z-буферу така.

Крок 1: ініціалізують буфер глибини значенням глибини фону, а буфер кадру – значеннями атрибутів фону. Крок 2: будують растрову розгортку проекції поточної грані. Крок 3: для кожного пікселя  $(x_i, y_j)$  цієї розгортки обчислюють глибину  $z_{ij}$ . Крок 4: порівнюють значення  $z_{ij}$  з відповідним пікселю  $(x_i, y_j)$  значенням  $z_{ij}^{(Buf)}$ , що зберігається в буфері глибини; якщо  $z_{ij} < z_{ij}^{(Buf)}$  ( $z_{ij}$  знаходиться ближче), то  $z_{ij}^{(Buf)}$  присвоюють значення  $z_{ij}$ , заносючи його до відповідної позиції буферу глибини, а атрибути пікселя  $z_{ij}$  записують до буферу кадру. Крок 5: переходять до наступної необробленої грані та до кроку 2.

*Характеристика методу.* Використовується для зображення складних сцен із застосуванням ортогонального проектування на картинну площину. Нескладна модифікація методу дозволяє отримувати перерізи поверхонь паралельними площинами.

Час роботи цього методу: лінійно залежить від кількості точок растру та глибинної складності сцени (усередненої кількості граней, які взаємно закривають одна одну); не залежить від дозволу об'єкту.

Метод Z-буферу є зручним для апаратної реалізації завдяки простоті алгоритму та використаного в цьому алгоритмі набору операцій.

### **3. Метод видалення нелицьових граней.**

*Сутність методу.* Деяку грань многогранника називають лицьовою (нелицьовою), якщо: зовнішня нормаль до грані та центр проектування знаходяться в одному (різних) напівпросторах відносно площини, що містить грань – у разі центрально-



го проектування; вектор зовнішньої нормалі до грані та вектор проектування утворюють гострий (тупий) кут – у разі паралельного проектування. При проектуванні многогранника на картинну площину, нелицьові грані будуть невидимими. Це вдвічі скорочує кількість граней, які потрібно аналізувати на видимість. Для опуклих багатогранників цей метод дає повний розв'язок задачі – достатньо видалити всі нелицьові грані. Для неопуклих багатогранників додатково аналізують на видимість лицьові грані.

*Характеристика методу.* Використовується для побудови зображень многогранників на картинній площині з використанням центральної проекції з заданим центром проектування або ортогональної проекції з заданим напрямним вектором. Час роботи методу є пропорційним кількості граней.

#### **4. Методи аналізу кількісної невидимості точок поверхні.**

*Сутність методів.* У основу цих методів покладено такі поняття: точка гладкої поверхні є регулярною точкою відображення проектування поверхні на картинну площину, якщо вона разом зі своєю малою околицею може бути взаємно однозначно спроектована на цю площину; контурними лініями гладкої поверхні називають зв'язні компоненти, що входять до складу множини нерегулярних точок деякого відображення проектування цієї поверхні на картинну площину (як правило, це гладкі криві).

Ці методи базуються на впровадженні функції кількісної невидимості точок гладкої поверхні (значенням функції у фіксованій точці поверхні є кількість тих точок, які закривають її). Властивості вказаної функції дозволяють звести задачу загороджування на гладкій поверхні до вивчення поведінки цієї

функції на контурних лініях, видимі частини яких є обрисами поверхні. Аналізуючи величини стрибків функції, виявляють: ділянки поверхні, що загороджені складками інших ділянок цієї поверхні; ділянки загороджування поверхнею самої себе в околицях точок зборки проектування.

*Характеристика методів.* Використовуються для розв'язування задач загороджування на гладких поверхнях та (при складнішому процесі аналізу) на многогранних поверхнях. Підсумком роботи методів є контурне зображення, що заповнюють каркасними лініями або напівтоновим зафарбовуванням (для цього використовують властивості функції кількісної невидимості в таких точках, які є прообразом регулярних значень відображення проектування). Час роботи методів лінійно залежить від дозволу поверхні, що є найкращим результатом для алгоритмів, які працюють в об'єктному просторі.

### **5. Метод розбиття картинної площини за допомогою активних вікон.**

*Сутність методу.* Використовується поняття активного вікна – геометричної області, яку можна пересувати вздовж картинної площини та розбивати на вікна меншого розміру.

Вміст поточного активного вікна аналізують на взаємне розташування з проекціями елементів 3D-сцени (найчастіше – з многокутниками, що є проекціями граней многогранників).

У підсумку аналізу, який можна проводити за різними системами правил, розглядають такі типові випадки: якщо вікно повністю покрите проекцією найближчого за глибиною елемента сцени, то зафарбовування вікна здійснюють кольором цієї проекції; якщо до вікна не потрапила жодна з проекцій елементів сцени, то його зафарбовують кольором фону; в разі

невиконання двох перших умов, вікно розбивають на чотири частини та повторюють аналіз для кожної з них.

Подрібнення вікна можна здійснювати до розміру пікселя картинної площини.

У підсумку роботи методу, отримують подання зображення в вигляді тетрарного дерева, листки якого зберігають атрибути кольору та інтенсивності відповідних активних вікон.

*Характеристика методу.* Застосовується для аналізу видимості сцен, складених із многогранних поверхонь. Для підвищення ефективності методу, вдосконалюють: правила формування, аналізу та розбиття активного вікна (для скорочення кількості операцій розбиття); метод аналізу взаємного розташування вікна та проєкцій елементів 3D-сцени (для скорочення кількості перевірочних дій). Час роботи методу залежить від щільності розташування проєкцій елементів 3D-сцени та може бути скорочений завдяки застосуванню для аналізу зображення паралельних обчислень.

#### **6. Метод кеширування вздовж регулярної прямокутної сітки.**

*Сутність методу.* Метод базується на такій властивості: якщо проєкції ребра та грані многогранника можна розмістити в прямокутних оболонках зі сторонами, паралельними координатним осям, та ці оболонки не перетинаються, то: проєкції не перетинаються; ребро повністю видиме відносно грані. Цю пару "ребро-грань" можна вилучити з подальшого аналізу ребра многогранника на виявлення видимих ділянок.

Для реалізації методу, на картинну площину накладають регулярну сітку з прямокутними чарунками. Для кожної чарунки складають список граней, проєкції яких її перети-

нають. Для кожного ребра: складають список чарунок, які перетинає його проекція; аналізують на видимість лише ті пари "ребро-грань", проекції яких мають загальні чарунки перетину.

*Характеристика методу.* Дозволяє уникнути прямого перебирання комбінацій "ребро-грань" многогранника в процесі аналізу взаємного розташування проекцій ребер та граней, націленого на виявлення видимих ділянок ребер (кількість операцій аналізу та час їх виконання стають на порядок меншими, але потрібна додаткова пам'ять для зберігання списків чарунок перетину).

### **7. Метод художника.**

*Сутність методу.* Базова технологія зображення в ортогональній проекції довільної поверхні, визначеної як функція від двох змінних на прямокутній сітці, є наступною: виконують триангуляцію області визначення функції та кусково-лінійну апроксимацію початкової (дискретно заданої) поверхні многогранною поверхнею з трикутними гранями; призначають вузлам сітки таку нумерацію  $(x_i, y_j)$  ( $i = 0, \dots, M - 1, j = 0, \dots, N - 1$ ), щоб було встановлено відповідність між зростанням номерів  $i, j$  та збільшенням відстані від спостерігача; проекції граней (їх зафарбовують кольорами відповідних граней) скидають на картинну площину в зворотному порядку (проекції граней, що лежать далі від спостерігача, обробляють раніше).

*Характеристика методу.* Належить до методів пріоритетів, ознакою яких є обов'язкове попереднє впорядкування елементів сцени за деяким параметром, наприклад, за глибиною елементу в сцені. Дає якісне напівтонове зображення поверхні. Завдяки попередньому впорядкуванню, знижується кількість операцій аналізу взаємного розташування, тобто економиться час.

### **8. Метод поточних обріїв.**

*Сутність методу.* На відміну від методу художника, скидає на картинну площину проекції ребер у прямому порядку. При такому підході, формуються межі області, забороненої для подальшого втручання (побудованої частини зображення): всередині цієї області розміщують ту частину 3D-сцени, що розташована ближче до спостерігача, ніж необроблені елементи сцени. Тому проекції необроблених елементів не повинні загороджувати тієї частини сцени, зображення якої вже отримане на екрані. Максимальну та мінімальну межі області побудованого зображення називають верхнім та нижнім поточними обріями (в процесі роботи методу, їх положення змінюється).

Для врахування факту наявності забороненої зони, використовують такі типові прийоми: для зображення проекції ребра, застосовують алгоритми растрової разгортки (попиксельного рисування) відрізка; виводять тільки ті точки проекції ребра, що знаходяться зовні забороненої області, визначеної верхнім та нижнім поточними обріями.

*Характеристика методу.* Належить до методів пріоритетів (з відповідною економією часу роботи). Використовується для побудови каркасних зображень поверхонь, які кусково-лінійно апроксимовані на базі стандартної триангуляції регулярної прямокутної сітки.

### **9. Метод двійкового розбиття об'єктного простору.**

*Сутність методу.* Базується на концепціях методу художника, тобто: грані попередньо впорядковують за зростанням їх відстані від спостерігача; проекції скидають на картинну площину в зворотному порядку.

Для впорядкування граней, використовують особливий підхід, який базується на наведених нижче домовленостях.

На першому кроці методу фіксують одну довільну грань 3D-сцени. Якщо деяка інша грань перетинає площину, що містить фіксовану грань, то її розбивають лінією перетину на дві окремі грані.

Далі грані 3D-сцени розбивають відносно фіксованої грані на дві підгрупи:

- ті грані, що лежать в одному напівпросторі зі спостерігачем відносно фіксованої грані (тобто не можуть бути загорожені фіксованою гранню);

- ті грані, що лежать у різних напівпросторах зі спостерігачем відносно фіксованої грані (тобто фіксована грань не може бути загорожена жодною з граней цієї групи).

На кожному наступному кроці методу, для кожної з виділених на попередньому кроці підгруп граней, процес фіксації певної грані та розбиття на дві підгрупи повторюють.

У підсумку, отримують впорядковане бінарне дерево, кожен вузол якого містить інформацію про деяку фіксовану грань, ліва (від вузла) гілка – про лицьові (по відношенню до фіксованої) грані, права (від вузла) гілка – про неліцьові грані.

Для отримання зображення, активізують алгоритм рекурсивного обходу бінарного дерева в напрямку "праве піддерево – коренева грань – ліве піддерево", з паралельним скиданням на площину проєкцій тих граней, що відповідають поточним вершинам.

*Характеристика методу.* Належить до методів встановлення пріоритетів. Може успішно використовуватися для створення каркасних та напівтонових зображень. Аналіз

геометричної 3D-сцени виконується виключно в об'єктному просторі. Робота методу не базується на враховуванні форми або перетинів граней.

### **10. Метод порядкового сканування.**

*Сутність методу.* Базовим кроком методу є аналіз взаємного розташування відносно спостерігача та розв'язування задачі загороджування для множини прямолінійних відрізків, що утворюються в процесі перетину граней 3D-сцени горизонтальною площиною, котрій поставлено у відповідність горизонтальний рядок растрової розгортки екрану.

При формуванні зображення, розглядають ряд горизонтальних площин, що в цілому відповідає всім рядкам растрової розгортки зображення на екрані.

*Характеристика методу.* Може формувати напівтонові та каркасні зображення. Працює в об'єктному просторі та в площині зображення. Дозволяє враховувати геометричні властивості 3D-сцени. В растровій модифікації часто використовується для апаратної реалізації.

## **Питання та завдання для самоперевірки**

1. Сформулюйте постановку задачі загороджування та її практичне значення. Чи можна запропонувати універсальний алгоритм для розв'язування будь-якої задачі загороджування?

2. Сформулюйте загальні концепції розв'язування задач видалення невидимих ділянок ліній та поверхонь у 3D-сценах.

3. За якими характеристиками класифікують методи розв'язування задач загороджування?

4. Наведіть перелік розповсюджених на практиці методів розв'язування задач загороджування та коротко охарактеризуйте їх.

5. Побудуйте детальний алгоритм розв'язування задачі загороджування одним із методів, розглянутих у параграфі 2.1 (наприклад, методом  $Z$ -буферу).

## 2.2. Моделювання реалістичної освітленості

Вагомий внесок до побудови на екрані комп'ютеру наочного, реалістичного зображення  $3D$ -сцени належить тим методам, які відтворюють природну освітленість цієї сцени: як в цілому, так і окремих об'єктів та фрагментів (рис.2.5) [41, 63, 64, 79, 120].

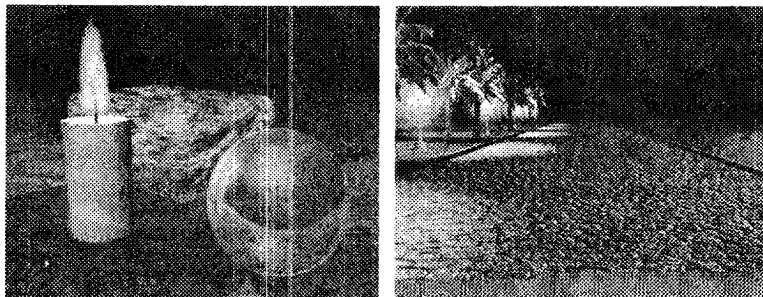


Рис.2.5. Приклади  $3D$ -сцен із реалістичною освітленістю

Найвідомішими формалізованими методами цього класу є:  
– метод зворотного трасування променів (*ray tracing*), який забезпечує високий ступінь реалізму, є простим у користуванні та підтримує розв'язування широкого кола задач;



– метод випромінюваності (*radiosity*), що усуває недоліки попереднього методу, забезпечуючи високу точність та ефективність роботи з дифузними поверхнями, а також обчислення глобальної освітленості окремо від побудови зображення та незалежно від місця розташування спостерігача.

**Метод зворотного трасування променів** базується на моделюванні проходження променів та обчисленні освітленості поверхонь у межах 3D-сцени, виходячи з законів геометричної оптики та відповідних математичних співвідношень.

За опорну точку  $B$  початку всіх променів беруть положення центру ока спостерігача в 3D-просторі. Крізь точку  $B$  та кожен піксель  $A_{ij}$  площини проєкцій у 3D-сцену направляють промінь, який відстежують у зворотному напрямку.

Інтенсивність пікселя  $A_{ij}$  площини проєкцій буде однозначно визначена освітленістю точки  $P$  перетину променя  $BA_{ij}$  з першою поверхнею, що трапилася на шляху цього променя.

У свою чергу, освітленість точки  $P$  визначають відбита та заломлювана енергії, отримані від джерел світла та від інших об'єктів сцени.

Для визначення цих видів освітленості, з точки  $P$  надсилають та аналізують ряд променів: надсилають промені до джерел світла та визначають внесок до освітленості точки  $P$  тих джерел світла, що не закриті іншими об'єктами; надсилають вторинні відбиті та переломлені промені та визначають світлову енергію, яку вони приносять від об'єктів 3D-сцени та з довкілля; надсилають та аналізують пучок променів, які рівномірно розподілені за всіма напрямками (в разі моделювання дифузних поверхонь).

Один промінь, випущений з точки  $B$ , може породити деревовидну структуру з сотень променів. На практиці розглядають лише ті промені, внесок яких до інтенсивності пікселя картинної площини не нижчий встановленого порогового рівня. У процесі трасування променів потрібно враховувати, що колір кожного пікселя визначають інтенсивності його складових частин (червоної, зеленої та синьої).

Розглянутий метод найкраще підходить для 3D-сцен без складних джерел світла та дифузних поверхонь (наприклад, для сцен з точковими джерелами світла та ідеально дзеркальними або прозорими тілами).

*Метод випромінюваності* базується на законі збереження енергії в замкненій системі.

Усі об'єкти 3D-сцени розбивають на фрагменти, для кожного з яких складають три рівняння балансу енергії (для трьох кольорових складових частин), отримуючи три системи лінійних алгебраїчних рівнянь. Їх розв'язують за невелику кількість ітерацій одним із відомих ітераційних методів (наприклад, Гауса-Зейделя), визначаючи значення освітленості кожного фрагменту 3D-сцени.

Далі здійснюють білінійну інтерполяцію освітленості за всіма об'єктами 3D-сцени (вона дає плавне природне освітлення), обирають точку спостереження, проєктують об'єкти сцени на картинну площину та будують зображення.

Для скорочення обсягу обчислень, застосовують прогресивну версію методу випромінюваності: ітераційно ведуть пошук поверхні з найбільшою випромінюваністю, починаючи з джерел світла, та визначають коефіцієнти, що показують її енергетичний вплив на інші фрагменти; відповідно цьому, на

кожному кроці корегують значення випромінюваності тих фрагментів, які залишилися не обробленими.

## Питання та завдання для самоперевірки

1. Охарактеризуйте практичну цінність задачі моделювання реалістичної освітленості зображень та сформулюйте сутність найвідоміших методів розв'язування цієї задачі.

2. Виконайте порівняльний аналіз переваг та недоліків методів зворотного трасування променів та випромінюваності.

3. Застосуйте метод зворотного трасування променів до спрощеної 3D-сцени, що складається з одного точкового джерела світла та однієї дзеркальної плоскої поверхні, розташованих одне відносно одної довільним чином.

## Навчальний практикум до гл.2

Удоскональте розв'язок тієї задачі, що була виконана вами в рамках навчального практикуму до гл.1, застосувавши методи видалення невидимих ділянок елементів 3D-сцени та моделювання реалістичної освітленості зображень.

**Етап 1.** Розробіть ПЗ для реалізації на комп'ютері одного з наведених нижче базових підходів до розв'язування задачі видалення невидимих ділянок ліній та поверхонь: а) метод прямого перебирання; б) метод Z-буферу; в) метод видалення нелицьових граней многогранників; г) метод аналізу кількісної невидимості точок поверхні; д) метод розбиття картинної пло-

щини за допомогою активних вікон; е) метод кешування вздовж регулярної прямокутної сітки; ж) метод художника; з) метод поточних обріїв; і) метод двійкового розбиття об'єктного простору; к) метод порядкового сканування; л) інший метод або комбінацію декількох методів розв'язування задач загороджування (з обґрунтуванням причин їх вибору).

**Еман 2.** Розробіть ПЗ для реалізації на комп'ютері одного з наведених нижче підходів до моделювання реалістичної освітленості зображень: а) метод зворотного трасування променів; б) метод випромінюваності; в) інший метод або групу методів для моделювання реалістичної освітленості зображень (з обґрунтуванням причин їх вибору).

## ГЛАВА 3. МОДЕЛЮВАННЯ АПРОКСИМОВАНОГО ПОДАННЯ ЛІНІЙ ТА ПОВЕРХОНЬ

### 3.1. Моделювання сплайнової апроксимації

**Постановка задачі.** В площинному та просторовому геометричному та графічному моделюванні розповсюджені задачі на апроксимацію [41, 79, 120]. *Апроксимація* – це побудова геометричного образу  $G^*$ , який може замінити з певним ступенем точності та за певних умов початковий геометричний образ  $G$ .

Якщо при проведенні апроксимації поставлено умову проходження геометричного образу  $G^*$  строго крізь усі задані вузлові точки початкового (можливо, дискретного) геометричного образу  $G$ , процес називають *інтерполяцією*; якщо поставлено пом'якшену умову проходження  $G^*$  поблизу вузлових точок, процес називають *згладжуванням*. Необхідність у цих видах побудов виникає тоді, коли на площині або в просторі задано дискретну послідовність точок або ліній, які необхідно з'єднати плавною кривою лінією або поверхнею.

Визначальну роль у розв'язуванні задач інтерполяції та згладжування відіграють *геометричні сплайни*, що активно застосовуються в теорії та практиці наближення функцій, у чисельних методах, в інженерних САПР, у системах автоматизації наукових досліджень, у різноманітних галузях комп'ютерної графіки тощо. Назва "сплайн" походить від англійського терміну "*spline*" – гнучкої полоски металу, за допомогою якої

креслярі проводили крізь задані опорні точки плавні криволінійні обводи корпусів суден, літальних апаратів, автомобілів тощо. У підсумку отримували набір точних перерізів (плазів), який визначав форму поверхні проєктованого об'єкту.

З появою ЕОМ перейшли до ефективнішого способу задавання форм поверхонь проєктованих об'єктів – шляхом підбору та програмування сукупності нескладних формул, які моделюють (надають аналітичний опис) запропонованих конструктором (виходячи з його експертних знань) кривих та поверхонь, дозволяючи відновлювати вигляд виробу з необхідним ступенем точності. Засоби комп'ютерної графіки суттєво допомагають у проєктуванні, демонструючи конструктору можливі результати його дій та надаючи широкі можливості їх аналізу та корегування.

У процесі розв'язування вказаних задач широко використовують аналітичний опис геометричних сплайнів у вигляді сплайн-функцій. Теорія сплайнів є розвиненою та розгалуженою, має велику кількість практичних застосувань, а сплайн-функції поділяють на значну кількість класів [41, 60, 63, 79, 120]. У рамках цього навчального посібника буде викладено початкові уявлення про сплайн-функції.

Будемо розглядати ази теорії сплайнів на базі типових класів задач – за заданим масивом опорних точок на  $2D$ -площині або в  $3D$ -просторі необхідно побудувати криву лінію або криволінійну поверхню, що: а) проходить безпосередньо крізь усі опорні точки (задача інтерполяції); б) проходить поблизу опорних точок із заданим припустимим рівнем відхилення (задача згладжування). В процесі розв'язування цих задач необхідно вирішувати два основних питання [174]: який клас кривих

найдоцільніше обрати за базовий та за якою технологією проводити пошук розв'язку?

**Інтерполяційні кубічні та бікубічні сплайни.** Для забезпечення найефективнішої інтерполяції припустимий клас кривих повинен відповідати таким вимогам: 1) побудована на його основі інтерполяційна крива повинна проходити крізь усі опорні точки; 2) розв'язок задачі повинен бути єдиним; 3) інтерполяційна крива повинна змінюватися плавно; 4) для отриманого класу кривих повинна бути характерна простота опису та застосування на практиці; 5) змінювання кількості або координат опорних точок не повинне призводити ні до необхідності перерахунку всієї кривої, ні до суттєвого впливу на вигляд многочлену та його значного відхилення від очікуваного результату.

Перші чотири властивості є характерними для всіх класів многочленів, які однозначно визначає набір коефіцієнтів. Наприклад, можна взяти за базовий клас кривих відомий у математичному аналізі інтерполяційний многочлен Лагранжа:

$$L_m(x) = \sum_{i=0}^m y_i \frac{w_m(x)}{(x - x_i)w'_m(x_i)}, \quad w_m(x) = \prod_{j=0}^m (x - x_j), \quad (3.1)$$

де  $\{(x_i, y_i), i = 0, 1, \dots, m, x_0 < x_1 < \dots < x_{m-1} < x_m\}$  – опорний набір точок.

Але використання одного многочлену для побудови всієї інтерполяційної кривої не дозволяє задовольнити третю, четверту та п'яту вимоги.

Так, використання многочлену Лагранжа призведе до таких недоліків:

– оскільки ступінь многочлену (3.1) лише на одиницю менша за кількість опорних точок  $m$ , то збільшення  $m$  призводить до крутих сплесків та спадів значень многочлену між

опорними точками, тобто – значного ухилення від очікуваної плавної кривої;

– змінювання лише однієї точки, що часто зустрічається на практиці, вимагає повного перерахунку всієї кривої та може суттєво вплинути на вигляд многочлену.

Багато з цих недоліків нескладно усунути, якщо будувати інтерполяційні сплайн-функції за досконалішим методом: за базовий клас кривих традиційно обирають многочлени, але будують їх інакше – послідовно, ланка за ланкою, окремо для кожної пари опорних точок. Важливо правильно обрати для кожного з цих многочленів: ступінь; коефіцієнти, які забезпечували б гладке сполучення сусідніх ланок (для того, щоб підсумкова крива змінювалася плавно).

Найбільш відомими класами інтерполяційних сплайн-функцій є інтерполяційні кубічні та бікубічні сплайни.

**Визначення 3.1.** Нехай на площині  $xOy$  задано набір з  $m+1$  точок вигляду  $A_i(x_i, y_i)$ ,  $i = 0, \dots, m$ ,  $x_0 < x_1 < \dots < x_{m-1} < x_m$ . *Інтерполяційним кубічним сплайном* називають функцію  $y = S(x)$ , яка має такі властивості: 1)  $\forall (i = 0, \dots, m)$  справедлива рівність вигляду  $S(x_i) = y_i$ ; 2) на кожному з  $m$  відрізків  $[x_i, x_{i+1}]$  ( $i = 0, 1, \dots, m - 1$ )

$$y = S(x) = \sum_{j=0}^3 a_j^{(i)} (x - x_i)^j; \quad (3.2)$$

3) на всій області визначення  $[x_0, x_m]$   $S(x)$  має неперервну другу похідну.

До класу інтерполяційних кубічних сплайнів належить і та функція  $S(x)$ , яка описує профіль гнучкої металевої лінійки між двома сусідніми опорами в тому разі, коли цю лінійку постави-



ти на ребро, закріпити один із її кінців та прокласти між рядом опор.

Для побудови сплайну  $S(x)$  на всій області визначення  $[x_0, x_m]$  необхідно знайти  $4m$  коефіцієнтів, оскільки на кожному з  $m$  відрізків  $[x_i, x_{i+1}]$  ( $i=0, 1, \dots, m-1$ ) сплайн  $S(x)$  однозначно визначають чотири коефіцієнти  $a_0^{(i)}, a_1^{(i)}, a_2^{(i)}, a_3^{(i)}$ . Для цього достатньо розв'язати систему з  $4m$  рівнянь, яка містить:  $(m+1)$  рівнянь вигляду  $S(x_i) = y_i$  ( $i = 0, 1, \dots, m$ );  $3 \cdot (m-1)$  рівнянь-умов на коефіцієнти, що забезпечують неперервність сплайн-функції  $S(x)$ , її першої похідної  $S'(x)$  та другої похідної  $S''(x)$  в усіх внутрішніх вузлах  $x_i$  ( $i = 1, \dots, m-1$ ); граничні умови  $s'(x_0) = l_0, s'(x_m) = l_m$ , де  $l_0, l_m$  – значення перших похідних  $S(x)$  на кінцях відрізка  $[x_0, x_m]$ .

**Визначення 3.2.** Нехай на площині  $xOy$  задано набір з  $(m+1) \cdot (n+1)$  точок вигляду  $(x_i, y_j) \in D$  ( $i = 0, 1, \dots, m, j = 0, 1, \dots, n, x_0 < x_1 < \dots < x_m, D$  – деяка прямокутна область), а в 3D-просторі визначено відповідний набір точок  $A_i^{(3)}(x_i, y_j, z_{ij})$  ( $i = 0, 1, \dots, m, j = 0, 1, \dots, n, y_0 < y_1 < \dots < y_n$ ).

**Інтерполяційним бікубічним сплайном** називають функцію  $z = S(x, y)$ , що має такі властивості: 1) для будь-якого  $i = 0, \dots, m$  та  $j = 0, \dots, n$  справедлива рівність  $S(x_i, y_j) = z_{ij}$ ; 2) на кожному прямокутнику  $[x_i, x_{i+1}] \cdot [y_j, y_{j+1}]$  ( $i=0, 1, \dots, m-1, j = 0, 1, \dots, n-1$ )

$$z = S(x, y) = \sum_{i=0}^3 \sum_{k=0}^3 a_{ik}^{(ij)} (x - x_i)^i (y - y_j)^k; \quad (3.3)$$

3) на всій області визначення  $[x_0, x_m] \times [y_0, y_n]$  функція  $z = S(x, y)$  має неперервну другу похідну.

Графіком інтерполяційного бікубічного сплайну є поверхня в 3D-просторі. Її будують аналогічно 2D-сплайну: визначають  $16mn$  коефіцієнтів  $a_k^{(ij)}$  у формулі (3.3), побудувавши та розв'язавши систему з  $16mn$  лінійних рівнянь, яка зв'язує ці коефіцієнти. Система містить: рівняння виду  $S(x_i, y_j) = z_{ij}$ ; рівняння, що забезпечують виконання третьої властивості функції  $S(x, y)$ ; граничні умови для першої похідної функції  $S(x, y)$ .

Переваги розглянутих підходів до сплайнової інтерполяції такі: графіки сплайн-функцій проходять крізь усі опорні точки, повністю зберігаючи початково задану інформацію; системи лінійних рівнянь, які необхідно складати в процесі побудови сплайн-функцій, є простими, а для їх розв'язування є багато ефективних методів. Але у цієї технології залишається суттєвий недолік: змінювання лише однієї точки, що на практиці є типовим випадком, вимагає перерахунку всіх коефіцієнтів сплайнової кривої.

**Згладжувальні базові сплайни та бета-сплайни.** Часто на практиці набір опорних точок вимушені задавати не точно, а наближено, тобто виконання вимоги про проходження графіка функції крізь усі опорні точки є зайвим: достатньо, щоб графік проходив поблизу них з припустимим рівнем відхилення. В цьому випадку від задач інтерполяції можна переходити до задач згладжування, методи розв'язування яких вільні від багатьох недоліків. Розглянемо особливості задач згладжування на прикладі 3D-випадку (ці підходи будуть справедливими і на 2D-площині).

Одним із найвідоміших підходів до побудови згладжувальної кривої є використання широко відомої кривої Безьє (рис.3.1).

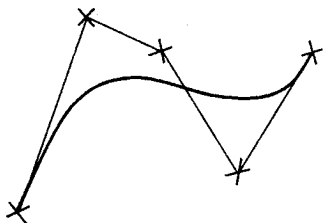


Рис.3.1. Крива Безьє

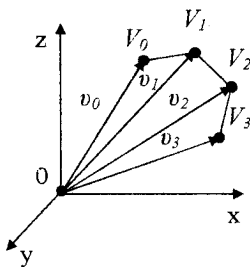


Рис.3.2. Контрольна ламана

**Визначення 3.3.** Нехай на площині або в просторі задано впорядкований набір точок  $V_0, V_1, \dots, V_m$ , визначених векторами  $v_0, v_1, \dots, v_m$  що виходять з початку координат (рис.3.2). Ламану  $V_0, V_1, \dots, V_m$  називають *контрольною ламаною*, породженою масивом векторів  $v = \{v_0, v_1, \dots, v_m\}$ .

**Визначення 3.4.** *Кривою Безьє*, породженою масивом векторів  $v = \{v_0, v_1, \dots, v_m\}$ , називають криву, визначену векторним рівнянням

$$z(t) = \sum_{i=0}^m \alpha_i v_i = \sum_{i=0}^m C_m^{(i)} t^i (1-t)^{m-i} v_i, \quad 0 \leq t \leq 1,$$

$$C_m^{(i)} = \frac{m!}{i!(m-i)!}, \quad (3.4)$$

де  $C_m^{(i)}$  – коефіцієнти в розкладанні біному Ньютона, що дорівнюють кількості сполучень з  $m$  елементів по  $i$ .

Доцільність використання кривої Безьє для розв'язування задач згладжування обумовлена такими властивостями: крива Безьє починається в точці  $V_0$  та закінчується в точці  $V_m$ , торкаючись кінцевих відрізків  $V_0V_1$  та  $V_{m-1}V_m$  контрольної ламаної; крива Безьє є гладкою; функціональні коефіцієнти  $\alpha_i$  при

вершинах  $V_i$ , заданих векторами  $v_i$  ( $i = 0, 1, \dots, m$ ), є універсальними многочленами Бернштейна, тобто

$$\alpha_i \geq 0, i = 0, 1, \dots, m, \sum_{i=0}^m \alpha_i = 1. \quad (3.5)$$

Виходячи з цього, крива Безьє цілком належить до опуклої оболонки вершин масиву  $V = \{V_0, V_1, \dots, V_m\}$  (порядок точок у масиві  $V$  суттєво впливає на вигляд кривої Безьє, бо вона намагається якнайщільніше наблизитися до контрольної ламаної, повторивши її в гладкому варіанті).

Використання немодифікованої кривої Безьє має недоліки, що заважають ефективному розв'язуванню задач згладжування: ступінь функціональних коефіцієнтів  $m$  прямо пов'язаний із кількістю точок у масиві  $m + 1$ ; введення нової точки вимагає повного перерахунку параметричних рівнянь; зміна координат однієї точки призводить до помітної зміни всієї кривої.

Для усунення цих недоліків вдосконалюють векторне рівняння (3.4), замінюючи многочлени Бернштейна  $\alpha_i$  ( $i = 0, 1, \dots, m$ ) на "*базові сплайни*" (*base splines, B-splines*):

$$r(t) = \sum_{i=0}^m N_{i,q}(t) v_i, \quad 0 = t_0 < t_1 < \dots < t_{m-1} < t_m = 1, \quad (3.6)$$

де нові функціональні коефіцієнти визначено рекурентними формулами:

$$\begin{aligned} N_{i,1}(t) &= 1, t \in [t_i, t_{i+1}]; \quad N_{i,1}(t) = 0, t \notin [t_i, t_{i+1}]; \\ N_{i,q}(t) &= \frac{t - t_i}{t_{i+q-1} - t_i} N_{i,q-1}(t) + \frac{t_{i+q} - t}{t_{i+q} - t_{i+1}} N_{i+1,q-1}(t). \end{aligned} \quad (3.7)$$

Функції  $N_{i,q}$  мають такі специфічні властивості: зі збільшенням індексу  $q$  ступінь многочленів, які визначають ці функції, зростає (для функцій на відрізку  $[t_i, t_{i+q}]$  він дорівнює  $q - 1$ ); на інтервалі  $(t_i, t_{i+q})$   $N_{i,q}(t) > 0$ , зовні цього інтервалу  $N_{i,q}(t) = 0$ ; на всій області визначення  $[t_0, t_m]$  функція  $N_{i,q}(t)$ , де  $q \geq 3$ , має не-

перервні похідні до порядку  $q-2$  включно. Функції  $N_{i,q}(t)$  максимально зберігають набір цінних властивостей кривої Безье: крива, задана векторним рівнянням (3.6), виходить з вершини  $V_0$  та входить до вершини  $V_m$ , обов'язково торкаючись кінцевих відрізків  $V_0V_1$  та  $V_{m-1}V_m$  контрольної ламаної; зберігається достатня для більшості практичних задач гладкість цієї кривої (за умови використання  $q \geq 4$  усі функціональні коефіцієнти будуть мати неперервні другі похідні); для функцій  $N_{i,q}$  зберігається рівність

$$\sum_{i=0}^m N_{i,q}(t) = 1, \quad (3.8)$$

завдяки якій крива, задана векторним рівнянням (3.6), завжди належатиме до опуклої оболонки вершин масиву  $V$ .

Окрім цього, криві, побудовані за допомогою векторного рівняння (3.6) з функціональними коефіцієнтами  $N_{i,q}$  виду (3.7), мають таку важливу локальну властивість: змінювання однієї вершини в масиві  $V$  не призводить до необхідності повного змінювання всієї кривої (перерахунок потрібен, в силу властивостей функціональних коефіцієнтів  $N_{i,q}$ , тільки для п'яти доданків).

Оскільки коефіцієнти при чотирьох послідовних вершинах у формулі (3.6) є позитивними, а їх сума дорівнює одиниці, то визначений ними відрізок  $B$ -сплайнової кривої повинен лежати всередині опуклої оболонки цих вершин (чотирикутника на  $2D$ -площині або тетраедра в  $3D$ -просторі).

Незважаючи на позитивні якості,  $B$ -сплайни не вирішують усіх проблем процесу згладжування (наприклад, таку типову проблему, коли після аналізу побудованого  $B$ -сплайну необхідно підправити криву в одному або декількох місцях, не змінюючи початкового масиву точок  $V$ ). Ці задачі доцільно розв'я-

зувати з використанням іншого класу сплайн-функцій – *бета-сплайнів* ( $\beta$ -сплайнів): найзручнішим методом розв'язування цих задач є введення додаткових параметрів до векторного рівняння кривої (3.6), що й дозволяють ефективно здійснити  $\beta$ -сплайни.

Є численні підходи до подання  $\beta$ -сплайнів. Розглянемо один із найбільш розповсюджених та простих із них. Згладжувальну криву  $\alpha$  будують як складену регулярну криву, що відповідає таким умовам:

1) відрізками складеної регулярної кривої  $\alpha$  є часткові регулярні криві  $\alpha_i$  ( $i = 0, 1, \dots, m$ ), які описано векторними рівняннями (3.9) з незалежними від  $i$  ваговими функціональними коефіцієнтами виду (3.10), та радіус-вектори яких пов'язані між собою в точках стику умовами геометричної неперервності виду (3.8):

$$r_i(t) = \sum_{j=-2}^1 b_j(t) v_{ij}, \quad 0 \leq t \leq 1, \quad (3.9)$$

$$b_j(t) = \sum_{k=0}^3 c_{kj}(\theta_1, \theta_2) t^k, \quad j = -2, -1, 0, 1; \quad (3.10)$$

2) крива  $\alpha$  повина починатися в вершині  $V_0$  та закінчуватися в вершині  $V_m$ , торкаючись кінцевих відрізків  $V_0V_1$  та  $V_{m-1}V_m$  контрольної ламаної.

Технологія роботи з  $\beta$ -сплайнними кривими така: для кожної пари радіус-векторів  $r_i$  та  $r_{i+1}$ , визначених формулою (3.9), складають умови геометричної неперервності виду (3.8), тобто отримують систему рівнянь, з якої визначають вагові функціональні коефіцієнти  $b_j(t)$ ,  $j = -2, -1, 0, 1$ ; підставляють значення  $b_j(t)$  до (3.10), отримуючи лінійну систему рівнянь, з якої знаходять числові значення коефіцієнтів  $c_{kj}$ ; підставляють значення  $c_{kj}$  до (3.10) та отримують вирази для вагових функ-

ціональних коефіцієнтів  $b_j(t)$  ( $j = -2, -1, 0, 1$ ) як функцій, залежних від параметрів  $t, \theta_1, \theta_2$ ; підставляючи отримані на попередньому кроці вирази для  $b_j(t)$  до (3.9), отримують вирази для векторних функцій  $r_2(t), \dots, r_{m-2}(t)$ ; складають вирази для векторних функцій  $r_0(t), r_1(t), r_{m-1}(t), r_m(t)$ , враховуючи проходження згладжувальної кривої  $\alpha$  крізь точки  $V_0$  та  $V_m$ , а також її торкання до відрізків  $V_0V_1$  та  $V_{m-1}V_m$ .

Спосіб задавання згладжувальних поверхонь є аналогічним способу задавання згладжувальних кривих, а тому більша частина цих поверхонь успадковує найважливіші властивості однойменних кривих.

Будемо вважати, що в 3D-просторі задано  $V$ -упорядкований набір точок  $V_{ij}, i = 0, 1, \dots, m, j = 0, 1, \dots, n$ , з'єднуючи які прямолінійними відрізками, отримаємо контрольний многогранник масиву  $V$ . Виходячи з цих початкових умов, згладжувальну поверхню будують просто – як тензорний добуток.

Прикладами відомих згладжувальних сплайн-функцій поверхонь є: бікубічна поверхня Безьє (3.11), бікубічна  $B$ -сплайнова поверхня (3.12) та  $(i,k)$ -вирізка бікубічного  $\beta$ -сплайну (3.13) [174].

$$r(u, v) = \sum_{i=0}^m \sum_{j=0}^n C_m^{(i)} C_n^{(j)} u^i (1-u)^{m-i} v^j (1-v)^{n-j} v_{ij},$$

$$0 \leq u \leq 1, 0 \leq v \leq 1; \quad (3.11)$$

$$r(u, v) = \sum_{i=0}^m \sum_{j=0}^n N_{i,4}(u) N_{j,4}(v) v_{ij}, \quad 0 \leq u \leq 1,$$

$$0 \leq v \leq 1; \quad (3.12)$$

$$r_{i,k}(u, v) = \sum_{j=-2}^1 \sum_{l=-2}^1 b_j(u) b_l(v) v_{i+j, k+l}, \quad 0 \leq u \leq 1,$$

$$0 \leq v \leq 1. \quad (3.13)$$

## Питання та завдання для самоперевірки

1. Дайте визначення поняттям апроксимації, інтерполяції, згладжування, геометричного сплайну та сплайн-функції. Охарактеризуйте їх практичне призначення.

2. Поясніть роль та наведіть формальний опис інтерполяційних кубічних та бікубічних сплайнів.

3. Що являє собою крива Безьє та які властивості вона має?

4. Дайте визначення та розтлумачте роль згладжувальних базових сплайнів та бета-сплайнів.

5. Задайте масиви з декількох довільних точок на  $2D$ -площині та в  $3D$ -просторі. Побудуйте для них, відповідно, криві та поверхні таких типів: а) інтерполяційні бі(кубічні); б)  $B$ -сплайнові; в) Безьє; г)  $\beta$ -сплайнові.

## 3.2. Моделювання кусково-лінійної апроксимації методом тріангуляції

**Постановка задачі кусково-лінійної апроксимації поверхонь та ізоліній на нерегулярній сітці методом тріангуляції.** У процесі вирішення багатьох теоретичних та прикладних питань виникає необхідність апроксимації  $3D$ -поверхонь у такій постановці: задано дискретну множину  $z = \{z_1, z_2, \dots, z_N\}$  значень деякої невідомої  $3D$ -функції від двох змінних  $z = f(x, y)$  у  $N$  довільно розташованих на  $2D$ -площині опорних точках  $A_i(x_i, y_i)$ ,  $i = 1, 2, \dots, N$ ; потрібно шляхом апрок-



симації отримати значення цієї функції в іншій,  $(N + 1)$ -ій точці, або (це найчастіше трапляється на практиці) у низці нових точок [63, 79, 111, 120].

Характерним класом задач, що потребують цієї апроксимації, є побудова карт ліній рівня (ізоліній) функцій двох змінних. Наприклад, це задачі створення топографічних карт, які є картами ізоліній функцій висоти рельєфу місцевості: на практиці значення цих функцій найчастіше доступні лише в точках скінченної нерегулярної множини (випадково розподілених точках деякої області площини).

Одним із найвідоміших підходів до розв'язування цієї задачі є кусково-лінійна апроксимація 3D-поверхні методом триангуляції (рис.3.3) [174].

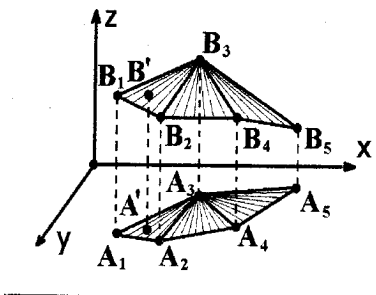


Рис.3.3. Схема триангуляції

**Еман 1.** На 2D-площині здійснюють триангуляцію – створюють нерегулярну сітку неперетинних трикутників з вершинами в опорних точках  $A_i(x_i, y_i)$ ,  $i = 1, 2, \dots, N$  (кількість трикутників, за теоремою Ейлера, не перевищує  $2 \cdot N$ ). Виходячи з вимог до точності інтерполяції, триангуляція повинна відповідати певним критеріям (мінімізації сумарної довжини ребер, досягнення найбільш правильної форми трикутників тощо). Результатом є список ребер трикутників, які утворюють триангуляцію.

**Еман 2.** 3D-поверхню  $P$ , яку визначено впровадженою вище функцією  $f$  (задано дискретно множиною точок  $B_i(x_i, y_i, z_i)$ ,  $i = 1, 2, \dots, N$ ), апроксимують кусково-лінійною 3D-поверхнею

$P^*$ , яку складено з трикутних граней з вершинами в точках  $B_i$  ( $i = 1, 2, \dots, N$ ) та визначено відповідною кусково-лінійною функцією від двох змінних  $f^*$ , яка теж приймає задані значення  $z_1, z_2, \dots, z_N$  у опорних точках  $A_i$  ( $i = 1, 2, \dots, N$ ).

Проаналізуємо, чому, замість використання традиційної регулярної сітки з прямокутними або квадратними чарунками, необхідно здійснювати триангуляцію початкових даних.

На всій площі регулярної прямокутної або квадратної сітки встановлено єдиний розмір чарунки, що визначає однакову детальність зображення поверхні для всіх її ділянок, незалежно від згущень та розріджень початкових точок. Для адекватного зображення поверхонь, які на певних ділянках характеризуються високою швидкістю змінювання значень  $z$ , необхідно значно зменшувати розмір чарунки всієї мережі, що вимагає значних обчислювальних потужностей та веде до створення нестійкостей.

У разі застосування триангуляції ці недоліки відсутні: оскільки нема єдиного для всіх початкових даних масштабу, триангуляція природно підстроюється під ці дані (в місцях згущень опорних точок трикутні чарунки сітки автоматично будуть меншими, розріджень – більшими). Але триангуляцію складніше програмувати, ніж регулярну сітку.

**Особливості триангуляції Делоне.** Найширшого застосування набула триангуляція Делоне: виконана побудова є однозначною, а опорні точки з'єднує сітка максимально правильних трикутників (це зручно в розрахунках, а в процесі обробки великих наборів даних дає прийнятні результати). *Геометрично триангуляція Делоне є планарним графом, подвійним до*

діаграми Вороного, що побудована на множині опорних точок. Розглянемо ці поняття детальніше (рис.3.4).

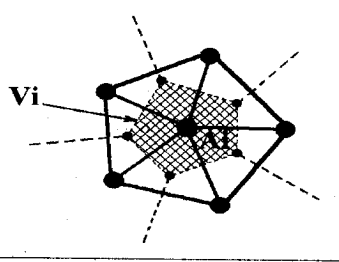


Рис.3.4. Схема триангуляції Делоне

**Многокутник Вороного**  $V_i$  опорної точки  $A_i$  ( $i = 1, 2, \dots, N$ ) – це множина всіх тих точок  $C_k$  ( $k = 1, 2, \dots, M$ ) на  $2D$ -площині, що  $\forall j \neq i$  виконано умову  $\rho(C_k, A_i) < \rho(C_k, A_j)$  ( $\rho$  – позначення функції відстані).

**Діаграма Вороного** – це об'єднана межа многокутників Вороного, створених для кожної з опорних точок.

Для побудови діаграми Вороного: для кожної опорної точки  $A_i$  ( $i = 1, 2, \dots, N$ ) проводять серединні перпендикуляри до відрізків, які з'єднують її з іншими опорними точками; знаходять область перетину напівплощин, які містять  $A_i$  та обмежені цими серединними перпендикулярами. Далі отримують планарний граф, подвійний до діаграми Вороного: з'єднують відрізками кожен таку пару опорних точок  $A_i$  та  $A_j$  ( $i \neq j; i, j = 1, 2, \dots, N$ ), для яких  $V_i$  та  $V_j$  мають спільне ребро. Результат цих дій є триангуляцією Делоне.

Для формального подання цього планарного графу як структури даних, зручно використовувати список вершин – опорних точок  $A_i$  ( $i = 1, 2, \dots, N$ ), та список ребер з подвійними зв'язками  $R_i$  ( $i = 1, 2, \dots, N$ ):  $R_i = \{B_i, E_i, G_i^{(L)}, G_i^{(R)}, P_i^{(B)}, P_i^{(E)}\}$ , де  $B_i$  та  $E_i$  – опорні точки, що є початком та кінцем ребра,  $G_i^{(L)}$  та  $G_i^{(R)}$  – номери граней ліворуч та праворуч від ребра;  $P_i^{(B)}, P_i^{(E)}$  – номери ребер, що йдуть одразу за ребром  $B_i E_i$  при обертанні

навколо вершин  $B_i$  та  $E_i$  в напрямку від ребра  $B_iE_i$  проти годинникової стрілки.

Побудова триангуляції Делоне, виходячи з її геометричного визначення, не є найефективнішим методом: час роботи алгоритму квадратично залежить від кількості опорних точок. Тому використовують топологічні властивості триангуляції Делоне, що отримують з її геометричного визначення. Характерними прикладами є: *колова властивість* (триангуляція Делоне має місце тоді й тільки тоді, коли кола, що описані навколо трикутних чарунок, не містять жодних інших опорних точок, окрім їх вершин); властивість *локальної перебудови* (триангуляція Делоне має місце тоді, якщо в будь-якій парі трикутних чарунок, які мають таке загальне ребро, що його можна переставити, не порушивши планарності триангуляції, при виконанні цієї дії мінімальний з шести кутів у парі трикутників не збільшиться).

Використовуючи ці властивості, можна застосувати індуктивний метод побудови триангуляції Делоне, час роботи якого є прямо пропорційним кількості опорних точок. Нехай на  $s$ -ому кроці триангуляцію побудовано. На  $(s + 1)$ -ому кроці до неї вводять нову опорну точку: розшукують або трикутник, до якого потрапила ця точка, або вершини, з якими її необхідно з'єднати (якщо точка не лежить усередині опуклої оболонки триангуляції, що була побудована на попередньому кроці); перевіряють виконання колового критерію; за необхідності здійснюють локальну перебудову триангуляції (перекиди ребер).

**Застосування тріангуляції до побудови ізоліній.** Лінія рівня  $h$  або *ізолінія* 3D-функції від двох змінних  $z=f(x, y)$  – це множина тих точок  $(x, y)$  на площині  $xOy$ , для яких  $z = f(x, y) = h$  (тобто це проекція на площину  $xOy$  лінії перетину поверхні, що є графіком функції  $z=f(x, y)$ , та площини  $z = h$ ) (рис.3.5).

Для побудови ліній рівня  $h_1, h_2, \dots, h_k$  тих функцій, які було визначено на початку параграфу (задано множиною дискретних значень на нерегулярній сітці), необхідно пройти три етапи:

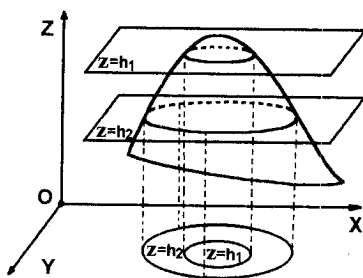


Рис.3.5. Побудова ізоліній

- 1) виконати тріангуляцію області визначення функції;
- 2) здійснити на базі тріангуляції кусково-лінійну апроксимацію поверхні, визначеної початковою функцією, за допомогою многогранної поверхні, що складена з трикутників із вершинами в опорній множині точок  $B_i(x_i, y_i, z_i)$  ( $i = 1, 2, \dots, N$ );
- 3) для кожного  $j = 1, 2, \dots, K$  знайти проекції на площину  $xOy$  прямолінійних відрізків перетину кожної трикутної грані з площиною  $z = h_j$ .

У підсумку отримаємо лінії рівня, кусково-лінійно апроксимовані на нерегулярній сітці методом тріангуляції.

Основною процедурою є знаходження відрізка перетину трикутної грані з площиною, паралельною площині  $xOy$ . При цьому повинні бути передбачені випадки, коли площині належать  $N = 0, 1, 2, 3$  вершин трикутної грані.

Розглянемо для повноти картини випадок, коли множину опорних точок задано на регулярній прямокутній сітці: якщо

початково задана сітка надто велика для подальшої роботи з такою сіткою, що має менші чарунки, то виконують попередню інтерполяцію на прямокутній сітці; розбивають кожну з прямокутних чарунок на два трикутники, отримуючи стандартну триангуляцію області визначення функції; виконують дії другого та третього етапів попереднього алгоритму.

### **Питання та завдання для самоперевірки**

1. Сформулюйте постановку задачі кусково-лінійної апроксимації поверхонь на нерегулярній сітці методом триангуляції.

2. Обґрунтуйте практичну доцільність широкого застосування методу триангуляції.

3. Розкрийте поняття триангуляції Делоне.

4. У чому полягає специфіка триангуляції ізоліній?

5. Задайте невеликий масив довільних точок  $3D$ -простору. Побудуйте для них поверхню та ізолінії, кусково-лінійно апроксимовані на нерегулярній сітці методом триангуляції.

### **3.3. Векторно-параметричне моделювання криволінійних обводів**

Векторно-параметричне моделювання обводів широко використовується в практиці комп'ютерного конструювання та дизайну тривимірних криволінійних промислових виробів у машино-, судно-, авіабудуванні та інших галузях [79, 119, 120].

Тривимірні криволінійні обводи можуть бути лінійними (3D-лінії) чи двопараметричними (3D-поверхні). Їх математичний опис зручно виконувати на основі векторно-параметричного методу задавання кривих ліній та поверхонь, оскільки ця форма запису універсальна та не залежить від кількості координат простору.

Згідно з векторно-параметричним методом, 3D-лінію зображують як траєкторію руху в просторі точки, поточні координати якої  $x, y, z$  та вектор цих координат  $r$  залежать від значень деякого параметру  $u$  (наприклад, від часу). В разі фіксації значення  $u = u_0$ , отримуємо конкретне положення точки на кривій –  $(x_0, y_0, z_0)$ .

Векторно-параметричний метод задавання 3D-лінії передбачає дві форми запису: параметричну (розгорнуту) форму

$$x = x(u), y = y(u), z = z(u) \quad (3.14)$$

та матричну (скорочену) форму

$$r = r(u). \quad (3.15)$$

Наведемо приклад запису відрізка прямої лінії, обмеженого точками 3D-простору  $r_0 (u = 0)$  та  $r_1 (u = 1)$ , за допомогою векторно-параметричного способу задавання:

$$r = r_0(1 - u) + r_1u, \quad 0 \leq u \leq 1. \quad (3.16)$$

Згідно з векторно-параметричним методом, 3D-поверхню зображують як траєкторію руху в просторі 3D-лінії, поточні координати точок якої залежать від значень двох параметрів,  $u$  та  $v$ . Параметр  $u$  відіграє ту ж роль, що і раніше, а від параметру  $v$  залежить величина зсуву 3D-лінії в просторі. В разі фіксації параметрів  $u = u_0$  та  $v = v_0$  для деякої множини значень

$u_0$  та  $v_0$ , отримуємо два сімейства ліній, що створюють на  $3D$ -поверхні  $u$ - $v$ -сітку.

Наведемо приклад запису фрагменту (частини, куска) білінійної поверхні, розташованої між чотирма заданими точками  $r_{00}$ ,  $r_{01}$ ,  $r_{10}$  та  $r_{11}$ , за допомогою векторно-параметричного методу:

$$r = r_{00}(1-u)(1-v) + r_{01}u(1-v) + r_{10}(1-u)v + r_{11}uv, \quad (3.17)$$

де  $0 \leq u, v \leq 1$ . Цю формулу можна розглядати як узагальнення формули (3.16), в якій константи  $r_0$  та  $r_1$  замінено на рівняння прямих, описаних у вигляді функцій від параметру  $v$ .

Якщо чотири точки, що відокремлюють частину білінійної поверхні, не належать одній площині, то будемо мати поверхню другого порядку в вигляді гіперболічного параболоїду або еліптичного гіперболоїду.

Часто за допомогою окремих частин білінійних поверхонь описують складні  $3D$ -поверхні, задані на впорядкованих масивах точок за допомогою білінійного методу інтерполяції.

**Найчастіше застосовують векторно-параметричне моделювання таких класів кривих та поверхонь:** лінійних тривимірних обводів у вигляді інженерних кривих другого та третього порядків, поліноміальних кривих у формі Бернштейна-Безьє, сплайнів; двопараметричних тривимірних обводів у вигляді поверхонь другого порядку, алгебраїчних поверхонь вищих порядків, лінійчастих поверхонь загального вигляду, поверхонь, утворених шляхом пересування та змінювання форми кривих другого та третього порядків, сплайнів [79, 119, 120, 125].

*Розглянемо основи векторно-параметричного моделювання лінійних тривимірних обводів у вигляді інженерних*



кривих другого та третього порядків, поліноміальних кривих у формі Бернштейна-Безьє.

Під інженерними кривими другого та третього порядків розуміють криві, подані в спеціальній, так званій інженерній формі, що є найбільш зручною для інженерного використання.

Дуга кривої другого порядку, для якої задані вектори координат точок характеристичного (контрольного) трикутника  $(r_1, r_2, r_3)$ , ваги цих точок  $(w_1, w_2, w_3)$ , а також відповідність між значеннями параметру  $u$  та точками дуги ( $u = 0$  та  $u = 1$  відповідають початковій точці  $r_1$  та кінцевій точці  $r_3$ ,  $0 < u < 1$  – внутрішнім точкам,  $u < 0$  та  $u > 1$  – точкам поза межами дуги), в інженерній формі подання має такий опис:

$$r = \frac{r_1 w_1 (1-u)^2 + r_2 w_2 u(1-u) + r_3 w_3 u^2}{w_1 (1-u)^2 + w_2 u(1-u) + w_3 u^2}. \quad (3.18)$$

В цілому цей опис визначає дугу кривої другого порядку з такими крайовими та середніми умовами: з початковою точкою  $r_1$  та кінцевою точкою  $r_3$ , із дотичною  $r_1 r_2$  у точці  $r_1$  та дотичною  $r_2 r_3$  у точці  $r_3$ , із проходженням через точку центра ваги характеристичного трикутника.

Раціональна крива третього порядку в інженерній формі подання має такий опис:

$$r = \frac{r_1 w_1 (1-u)^3 + r_2 w_2 u(1-u)^2 + r_3 w_3 u^2(1-u) + r_4 w_4 u^3}{w_1 (1-u)^3 + w_2 u(1-u)^2 + w_3 u^2(1-u) + w_4 u^3}. \quad (3.19)$$

До того, що було сказано про криві другого порядку, слід додати таке: початково визначається характеристичний чотирикутник, за кінцеву точку беруть  $r_4$  із вагою  $w_4$ , а співвідношення ваг  $w_1, w_2, w_3, w_4$  визначає форму кривої.

Перейдемо до розгляду поліноміальних кривих у формі Бернштейна-Безьє. Нехай  $r_i$  ( $i = 0, 1, \dots, n$ ) – радіус-вектори вер-

шин характеристичної ламаної. Поліноміальна крива першого порядку (пряма лінія) в формі Бернштейна-Безьє в векторно-параметричному вигляді описується формулою (3.16).

Поліноміальна крива другого порядку (парабола загального положення) в формі Бернштейна-Безьє, що визначає частковий випадок інженерної кривої другого порядку, в векторно-параметричному вигляді має такий опис:

$$r = r_0(1-u)^2 + 2r_1u(1-u) + r_2u^2. \quad (3.20)$$

Поліноміальна крива третього порядку (кубічна крива Безьє) в формі Бернштейна-Безьє, що визначає частковий випадок інженерної раціональної кривої третього порядку, в векторно-параметричному вигляді має такий опис:

$$r = r_0(1-u)^3 + 3r_1u(1-u)^2 + 3r_2u^2(1-u) + r_3u^3. \quad (3.21)$$

Поліноміальна крива  $n$ -го порядку в формі Бернштейна-Безьє в векторно-параметричному вигляді має такий опис:

$$r = \sum_{i=1}^n \frac{n!}{(n-i)!i!} u^i(1-u)^{n-i} r_i. \quad (3.23)$$

Збільшення  $n$  дає як недоліки, так і переваги: зв'язок між характеристичним многокутником та формою кривої слабшає, але на кінцях дуг кривих вищих порядків можна задавати похідні вищих порядків.

*Розглянемо основи векторно-параметричного моделювання двопараметричних тривимірних обводів у вигляді поверхонь другого порядку, алгебраїчних поверхонь вищих порядків, лінійчастих поверхонь загального вигляду, поверхонь, утворених шляхом пересування та змінювання форми кривих другого та третього порядків.*

Поверхні другого порядку є найбільш вивченими. Тому їх найчастіше використовують на практиці, де, на відміну від

теорії аналітичної геометрії, перевага віддається не неявному способу опису, а задаванню в вигляді векторно-параметричних функцій. Поверхня другого порядку може бути отримана пересуванням твірної, що являє собою криву другого порядку, вздовж напрямних (еліпса, кола, параболи, гіперболи), що належать до того ж класу.

Векторно-параметричні функції кривих другого порядку задано формулами:

$$x = a \cos(u), y = b \sin(u) - \text{еліпс} (a \neq b), \text{ коло} (a = b); \quad (3.23)$$

$$x = au, y = bu^2 - \text{парабола}; \quad (3.24)$$

$$x = a \cdot ch(u), y = b \cdot sh(u) - \text{гіпербола}. \quad (3.25)$$

Розглянемо еліптичні поверхні другого порядку, що мають еліптичний переріз (сфера, еліпсоїд обертання, одно-порожнинний гіперболоїд обертання, параболоїд обертання, еліптичний циліндр та еліптичний конус, утворені при виродженні гіперболоїду).

Розглянемо, як можна вивести векторно-параметричне рівняння поверхні в вигляді триосного еліпсоїду. Виконавши перерізи цієї поверхні площинами  $x = 0$ ,  $y = 0$ ,  $z = 0$ , отримаємо еліпси  $E_x$ ,  $E_y$  та  $E_z$ . За твірну доцільно взяти еліпс  $E_z$  з головними напівосями  $a$  та  $b$ , який формується в перерізі поверхні площиною  $z = 0$  та описується формулою (3.23). За напрямні доцільно взяти криві лінії, що відповідають певним умовам: еліпс  $E_z$  повинен пересуватися паралельно площині  $z = 0$ , а його осьові точки – вздовж еліпсів  $E_x$  та  $E_y$ . Виходячи з цього, векторно-параметричне рівняння поверхні в вигляді триосного еліпсоїду набуде вигляду:

$$\begin{aligned} x &= a(v) \cos(u), a(v) = a \cos(v), y = b(v) \sin(u), \\ b(v) &= b \cos(v), z = c(v), c(v) = c \sin(v), \end{aligned} \quad (3.26)$$

де  $a, b, c$  – головні напівосі, розташовані на координатних осях, або

$$r = r_0 + r_1 \cos(v) \cos(u) + r_2 \cos(v) \sin(u) + r_3 \sin(v), \quad (3.27)$$

де  $r_0$  – центр еліпсоїду;  $r_1, r_2, r_3$  – його діаметральні напівосі (головні в разі ортогональності  $r_1, r_2, r_3$  або косокутні в іншому разі).

Аналогічно можна вивести опис однопорожнинного еліптичного гіперболоїду

$$r = r_0 + r_1 ch(v) \cos(u) + r_2 ch(v) \sin(u) + r_3 sh(v) \quad (3.28)$$

або еліптичного параболоїду

$$\begin{aligned} x &= a(v) \cos(u), \quad a(v) = av; \quad y = b(v) \sin(u), \quad b(v) = bv; \\ z &= c(v), \quad c(v) = cv^2; \quad r = r_0 + r_1 v \cos(u) + r_2 v \sin(u) + r_3 v^2. \end{aligned} \quad (3.29)$$

Так само можна отримати опис інших еліптичних кривих.

Найтипівішими представниками двопараметричних обводів у вигляді алгебраїчних поверхонь вищих порядків є тор та гвинтова поверхня (прямий гелікоїд).

Тор – це тіло обертання, утворене шляхом обертання кола відносно осі, що належить площині цього кола, але не перетинає його. Якщо обертати навколо осі  $y$  коло, що лежить в площині  $xOy$  з центром на осі  $x$ , де  $x = r_1 + r_2 \cos(u)$ ,  $y = r_2 \sin(u)$ , то отримуємо такий опис тору:

$$\begin{aligned} x^* &= x \cos(v), \quad x = r_1 + r_2 \cos(u); \quad y^* = y, \quad y = r_2 \sin(u); \\ z^* &= z \sin(v), \quad z = r_1 + r_2 \cos(u). \end{aligned} \quad (3.30)$$

Гвинтова поверхня формується в підсумку одночасного рівномірного пересування та обертання навколо осі відрізка прямої лінії. Якщо відрізок заданий рівнянням  $x = u$ , а пересування та обертання здійснюється відносно осі  $z$ , то маємо такі рівняння:

$$x = u \cos(v), y = u \sin(v), z = cv. \quad (3.31)$$

Лінійчаста поверхня загального вигляду утворюється внаслідок руху прямолінійної твірної вздовж двох напрямних кривих ліній  $r_1(v)$  і  $r_2(v)$  та має наступне векторно-параметричне рівняння:

$$r = r_1(v)(1 - u) + r_2(v)u. \quad (3.32)$$

Наприклад, напрямними лінійчастої поверхні п'ятого порядку можуть бути такі векторні поліноми четвертого ступеня:

$$\begin{aligned} r_1(v) &= r_{10} + r_{11}v + r_{12}v^2 + r_{13}v^3 + r_{14}v^4, \\ r_2(v) &= r_{20} + r_{21}v + r_{22}v^2 + r_{23}v^3 + r_{24}v^4. \end{aligned} \quad (3.33)$$

Перейдемо до розгляду векторно-параметричного моделювання двопараметричних обводів у вигляді поверхонь, які утворено шляхом пересування та змінювання форми кривих другого та третього порядків. У практиці автоматизованого проектування прагнуть найпростішого визначення поверхонь. Тому рекомендовано формувати вказані поверхні шляхом сполучення окремих відсіків, які утворюються внаслідок руху дуги твірної, що може змінювати свої параметри, по дугам напрямних так, щоб початки та кінці дуг різних напрямних сходилися на одній твірній.

Твірними та напрямними можуть бути дуги інженерних кривих другого та третього порядків, поліноміальних кривих Бернштейна-Безьє тощо. Якщо твірні та напрямні мають різні порядки, сформовані на їх основі поверхні отримують комбіновані назви: бікубічно-раціональна, біполіноміальна Бернштейна-Безьє, бікубічна Безьє, лінійно-квадратична, квадратично-кубічна тощо.

У цій ситуації однією з найголовніших задач конструювання є забезпечення гладкості стиковки фрагментів. Гладкість першого порядку (відсутність зламів при стиковці фрагментів поверхонь) забезпечується першим порядком гладкості стиковки (однозначністю дотичної) напрямних та твірних. У разі необхідності досягти гладкості другого та третього порядків, задача стає складною.

Наведемо приклад роботи з інженерними кривими другого та третього порядків, коли траєкторії руху точок характеристичного многокутника твірної належать до вказаного класу кривих, а тими параметрами, що змінюються, є вагові коефіцієнти характеристичних точок. Підсумкові поверхні як правило будуть алгебраїчними поверхнями четвертого та шостого порядків, а їх порядок визначають суми порядків твірної та напрямних.

У цілому аналітичне визначення поверхні методами інженерних кривих другого та третього порядків базується на векторно-параметричному описі твірних за допомогою формул (3.18) та (3.19), а також за допомогою формул напрямних цих кривих, визначених як функції деякого іншого параметру.

Розглянемо аналітичний опис фрагменту квадратичної поверхні – алгебраїчної поверхні четвертого порядку, *uv*-сітка якої сформована з дуг кривих другого порядку. Фрагмент квадратичної поверхні задається дев'ятьма точками та відповідними значеннями їх ваги. Опис квадратичної поверхні через рівняння твірної, напрямних (траєкторій руху вершин характеристичного трикутника) та закону змінювання вагових коефіцієнтів має такий вигляд:

$$r = \frac{r_1 w_1 (1-u)^2 + r_2 w_2 u(1-u) + r_3 w_3 u^2}{w_1 (1-u)^2 + w_2 u(1-u) + w_3 u^2};$$

$$r_i = \frac{r_{i1} w_{i1} (1-v)^2 + r_{i2} w_{i2} v(1-v) + r_{i3} w_{i3} v^2}{w_{i1} (1-v)^2 + w_{i2} v(1-v) + w_{i3} v^2} \quad (i=1, 2, 3); \quad (3.34)$$

$$w_i = w_{i1} (1-v)^2 + w_{i2} v(1-v) + w_{i3} v^2 \quad (i=1, 2, 3).$$

Якщо підставити формули для  $r_1, r_2, r_3$  та  $w_1, w_2, w_3$  в (3.18), отримаємо такий опис фрагменту квадратичної поверхні:

$$r = R/W; \quad R = r_1 w_1 (1-u)^2 + r_2 w_2 u(1-u) + r_3 w_3 u^2;$$

$$W = w_1 (1-u)^2 + w_2 u(1-u) + w_3 u^2;$$

$$w_i = w_{i1} (1-v)^2 + w_{i2} v(1-v) + w_{i3} v^2 \quad (i=1, 2, 3); \quad (3.35)$$

$$r_i w_i = r_{i1} w_{i1} (1-v)^2 + r_{i2} w_{i2} v(1-v) + r_{i3} w_{i3} v^2 \quad (i=1, 2, 3).$$

**Розглянемо основи векторно-параметричного моделювання тривимірних криволінійних обводів у вигляді сплайнів.**

У найбільш загальному випадку, під сплайном розуміють будь-яку функцію, сформовану шляхом об'єднання декількох відрізків кусково-постійних функцій в їх кінцевих точках таким чином, щоб була досягнена рівність у точках стику відповідних функцій та деяких їх похідних. Тобто всі розглянуті вище типи функцій можуть бути використані для створення сплайнів.

Найвигіднішими для моделювання технічних кривих та поверхонь вважають класичні кубічні сплайни, подані за допомогою кубічних поліномів такого вигляду:

$$f(x) = a_0(i)y(i) + a_1(i)y(i+1) + b_0(i)y'(i) + b_1(i)y'(i+1),$$

$$a_0(i) = 1 - 3u^2 + 2u^3, \quad a_1(i) = 3u^2 - 2u^3,$$

$$b_0(i) = h(u - 2u^2 + u^3), \quad b_1(i) = h(u^3 - u^2),$$

$$h = x(i+1) - x(i), \quad u = (x - x(i))/h,$$

де  $x(i) < x(i+1)$ ,  $i = 0, 1, \dots, N$ ;  $y(i)$ ,  $y'(i)$ ,  $y(i+1)$ ,  $y'(i+1)$  – значення функції та її похідної в точках  $x(i)$ ,  $x(i+1)$ .

Якщо визначений точковий каркас  $x(i)$ ,  $y(i)$ ,  $y'(i)$ , де  $x(i) < x(i+1)$ ,  $i=0, 1, \dots, N$ , його завжди можна інтерполювати гладкою функцією за допомогою лінійного кубічного сплайну.

Через вимогу монотонного збільшення  $x(i)$ , задану умовою  $x(i) < x(i+1)$ ,  $i = 0, 1, \dots, N$ , сплайном (3.36) не вдається моделювати лінії довільної конфігурації. Усунути цей недолік дозволяють векторно-параметричні сплайни, в яких кожна координата задається окремим сплайном від єдиного параметру, за який найкраще взяти набігаючу відстань по довжинах хорд заданої ламаної.

Як правило, значення похідних, потрібні для моделювання кубічного сплайну, треба обчислювати. Для цього існує спеціальна теорія – метод кубічних сплайнів.

## **Питання та завдання для самоперевірки**

1. Яку роль відіграє метод геометричного векторно-параметричного моделювання в автоматизованому проектуванні тривимірних криволінійних обводів?

2. Для яких класів кривих та поверхонь найчастіше застосовують векторно-параметричне моделювання?

3. Запишіть векторно-параметричні моделі кривих та поверхонь, які було розглянуто в параграфі 3.3, спорядивши їх конкретними значеннями коефіцієнтів та меж змінювання параметрів.



## Навчальний практикум до гл.3

У процесі виконання завдань цього практикуму необхідно: побудувати концептуальну та математичну моделі розв'язування задачі; обґрунтувати вибір структур даних, апаратного та програмного інструментарію, технологій створення ПЗ; розробити відповідні алгоритми та програмний код, контрольні приклади до нього.

**1-а група завдань.** Розробіть ПЗ, що є однією з перелічених нижче складових частин програмного пакету з побудови інтерполяційних та згладжувальних сплайнових кривих та поверхонь: а) інтерполяційний многочлен Лагранжа; б) інтерполяційний многочлен Ньютона; в) кубічні та бікубічні сплайни; г) криві та поверхні Безьє; д)  $B$ -сплайнові криві та поверхні; е)  $\beta$ -сплайнові криві та поверхні.

**2-а група завдань.** Розробіть пакет програм для зображення на комп'ютері 3D-поверхонь та побудови ліній рівня, з застосуванням методу триангуляції, для функцій двох змінних, які задані дискретним рядом значень такого типу: а) функцію задано у вузлах регулярної прямокутної сітки; б) функцію задано на нерегулярній множині точок растру.

**3-а група завдань.** Розробіть ПЗ для геометричного векторно-параметричного моделювання лінійних тривимірних обводів на базі одного з наступних класів кривих: а) інженерних кривих другого порядку; б) інженерних кривих третього порядку; в) поліноміальних кривих у формі Бернштейна-Безьє; г) сплайнів.

*4-а група завдань.* Розробіть ПЗ для геометричного векторно-параметричного моделювання двопараметричних тривимірних обводів на базі одного з наступних класів поверхонь: а) поверхонь другого порядку; б) алгебраїчних поверхонь вищих порядків; в) лінійчастих поверхонь загального вигляду; г) поверхонь, утворених шляхом пересування та змінювання форм кривих другого та третього порядків; д) сплайнів.

## Висновки до розд.1

В основу сучасної комп'ютерної графіки покладено складний та розгалужений математичний апарат, розгляд якого в повному обсязі та в усіх деталях, у рамках невеликого лекційного та практичного курсу, є неможливим. Тому у розд.1 цього навчального посібника вашій увазі були запропоновані для вивчення та практичного опанування тільки найфундаментальніші поняття та методи, засвоєння яких є необхідним для успішного оволодіння наступними розділами цього посібника, а саме:

- характеристика основних типів координатних систем та координат, які найчастіше застосовують у комп'ютерній графіці;
- координатні перетворення кадрування та відсікання;
- поняття растру та базові підходи до растрового подання графічних об'єктів на екрані комп'ютеру;
- метод векторно-параметричного подання кривих ліній та поверхонь;
- математичний апарат геометричних перетворень паралельного перенесення, масштабування, обертання, паралельного та перспективного проектування, початкові уявлення про створення стереоскопічних зображень;
- основні підходи до роботи з інтерполяційними та згладжувальними сплайн-функціями, кусково-лінійної апроксимації на нерегулярній сітці методом триангуляції;
- базові концепції розв'язування задач блокування виведення на екран комп'ютеру невидимих ділянок ліній та поверхонь, створення реалістичного освітлення 3D-сцен.

---

---

## РОЗДІЛ 2. ТЕХНОЛОГІЇ КОМП'ЮТЕРНОЇ ГРАФІКИ

---

---

### ГЛАВА 4. ГРАФІЧНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

#### 4.1. Стандартизація графічного ПЗ

*Основні поняття. Головна мета стандартизації в комп'ютерній графіці полягає в тому, щоб основна частина ПЗ була апаратно-незалежною.* Вона повинна забезпечувати: інтерфейс з будь-яким пристроєм введення крізь апаратно-залежний узгоджувач; інтерфейс з будь-яким пристроєм відображення крізь драйвер. Апаратно-незалежне ПЗ повинне обслуговувати різні апаратні реалізації та бути переносимим з однієї графічної системи на іншу [137-139].

Стандарти надають такі переваги: забезпечують високий рівень мобільності прикладного графічного ПЗ; сприяють полегшенню вивчення та використання методів комп'ютерної графіки прикладними програмістами; дозволяють виробникам графічного обладнання визначати найбільш ефективне сполучення графічних можливостей, які підлягають реалізації.

Апаратно-незалежне графічне ПЗ має дві ключові функції: конструювання об'єктів; маніпулювання ними в процесі генерації зображення. Це визначило концепцію двох координатних систем: незалежної (світової), в якій конструюють об'єкт; апаратної, в якій описують те зображення, що підлягає виведенню.

Міжнародні організації зі стандартизації та механізми затвердження міжнародних графічних стандартів. Світові

стандарти в галузі інформаційних технологій, зокрема, міжнародні графічні стандарти, встановлюють дві міжнародні організації зі стандартизації: *ISO* (*International Organization for Standardization*) та *IEC* (*International Electrotechnical Commission*), що взаємодіють із зацікавленими групами з галузей економіки та політики в різних країнах (інші міжнародні організації теж можуть брати участь у встановлюванні стандарту) [137-139]. Технічні стандарти в галузі інформаційних технологій перевіряє технічний комітет *ISO/IEC JTC1* (*Joint Technical Committee*), який призначають сумісно *ISO* та *IEC*. Цей комітет пропонує проект стандарту національним організаціям зі стандартизації для голосування (для затвердження необхідно не менше 75 % голосів).

За стандартизацію в комп'ютерній графіці відповідає підкомітет *ISO/IEC JTC1/SC24* з такими провідними робочими групами: *WG1* "Архітектура", *WG2* "Інтерфейси прикладних програм", *WG3* "Метафайли та інтерфейси з пристроями", *WG4* "Мовні інтерфейси", *WG5* "Верифікація, тестування та реєстрація".

Історичний огляд стандартів комп'ютерної графіки. Історично найбільш відомими графічними стандартами стали: *CORE*, *GKS*, *IGES*, *NAPLPS*, *VDM*, *VDI*.

*CORE* – один із перших відомих стандартів машинної графіки. В основу *CORE-SYSTEM* покладено такі базові концепції: розділення функцій введення та виведення; мінімізації відмінностей між виведенням на графопобудовник та інтерактивний дисплей; двох координатних систем (світової, в якій конструюють зображення, та апаратної, в якій подають дані для його відображення); дисплейного файлу, що містить апаратну

координатну інформацію; дисплейного файлу сегментів, кожен із яких можна незалежно модифікувати як елемент; забезпечення функцій перетворення даних із світової системи координат до апаратної шляхом виклику видового перетворення.

**GKS (Graphical Kernel System – графічна коренева система)** – найбільш розповсюджений міжнародний стандарт на системи машинної графіки. Він визначає: межу між прикладними пакетами та графічним забезпеченням; вміст забезпечувального пакету. GKS є загальноприйнятим стандартом для 2D-графіки.

У рамках GKS: визначені функції генерації, подання, сегментування, перетворення та введення зображень; введено методологічно цілісний та повний опис концепцій машинної графіки. У GKS функції машинної графіки виконуються на рівні, що забезпечує їх незалежність від конкретних пристроїв введення та виведення. Перетворення незалежного подання графічних функцій до такого подання, що є специфічним для конкретної робочої станції, здійснюється драйверами пристроїв. Основними концепціями GKS є такі поняття: примітиви виведення, з яких будують вихідне зображення; системи координат та їх перетворення; робочі станції, що об'єднують ввідні та вивідні графічні пристрої; сегментація зображень та перетворення сегментів; метафайл, що забезпечує тривале зберігання та передавання зображень; списки станів графічної системи; рівні, на які розбито множину функцій; вікно та поле огляду в просторі незалежних та нормалізованих координат.

**IGES** – стандартний протокол обміну графічною інформацією, прийнятий Американським інститутом національних

стандартів. Він передбачає засоби обміну графічними даними між системами автоматизованого проектування та виробництва.

**NAPLPS** – південноамериканський стандарт, прийнятий у Канаді та США. Він використовує послідовність байтів у кодї *ASCII* та коди розширення для опису графічної та текстової інформації в окремих кадрах зображення.

**VDM** – метафайл для віртуального пристрою (апаратно-незалежний формат, що функціонує на рівні, наступному за драйверами пристроїв). Метафайл забезпечує передавання інформації про зображення між різними графічними пристроями.

**VDI** – інтерфейс віртуального пристрою, який діє на тому ж рівні, що й метафайл, та визначає протокол двостороннього зв'язку. Цей стандарт визначає інтерфейс між апаратно-незалежним ПЗ та апаратно-залежними кодами для зовнішніх пристроїв.

**Основні підсумки діяльності ISO та IEC зі стандартизації в комп'ютерній графіці.** Розглянемо стандарти, які є основними на даний час.

**GKS** (*Graphical Kernel System*) – набір базових функцій для двовимірної апаратно-незалежної комп'ютерної графіки. **GKS-3D** (*Graphical Kernel System for 3 Dimensions*) – розширення *GKS* для підтримки базових функцій 3D-графіки. **PHIGS** (*Programmer's Hierarchical Interactive Graphics System*) – набір базових функцій 3D-графіки, аналогічний *GKS-3D*, але орієнтований на накопичування графічної інформації в ієрархічній структурі даних та на застосування, що вимагають швидкої модифікації тих графічних даних, які описують геометрію об'єктів.

**Language bindings** (мовні інтерфейси) – подання функцій та типів даних функціональних графічних стандартів у стандар-

тизованих мовах програмування. *CGM* (*Computer Graphics Metafile*) – апаратно-незалежний формат обміну графічною інформацією, який використовують для передавання та запам'ятовування тієї інформації, що описує зображення. *CGI* (*Computer Graphics Interface*) – набір базових елементів для управління та обміну даними між апаратно-незалежним та апаратно-залежним рівнями графічної системи.

*CGRM* (*Computer Graphics Reference Model*) – модель стандартів комп'ютерної графіки, що визначає концепції та співвідношення майбутніх стандартів. *Ресстрація* – механізм ресстрації стандартизованих аспектів примітивів виведення, узагальнених примітивів, *escape*-функцій (для доступу до апаратних можливостей пристроїв) та інших графічних елементів. *Тестування реалізації на відповідність графічним стандартам* – проект, націлений на специфікування характеристик стандартизованих тестів, які використовують для визначення відповідності реалізації графічним стандартам, та вироблення приписів розробникам функціональних стандартів відносно правил відповідності.

**Загальна класифікація графічних стандартів.** Для забезпечення мобільності графічного ПЗ, потрібно стандартизувати: *базову графічну систему*, тобто графічний інтерфейс (набір базових графічних функцій), використовуючи для цього стандарти *Core System*, *GKS*, *GKS-3D*, *PHIGS*, *PHIGS+* тощо (ці графічні інтерфейси є міжнародними графічними стандартами); *графічний протокол* (порядок та правила обміну інформацією), використовуючи для цього стандарти *IGES*, *CGM* тощо (більша частина цих графічних протоколів є фактичними стандартами, а *CGM* – міжнародним). Зауважимо, що графічний



стандарт *PHIGS+* є розширенням *PHIGS*, з додатковими функціональними можливостями для тих застосувань, які вимагають урахування освітленості та розфарбовування (інтерполяції кольорів уздовж поверхні), додатковими можливостями керування відображенням, новими примітивами для підтримки ефективного опису складних поверхонь.

**Графічні протоколи та проекти з їх стандартизації можна класифікувати так:** апаратно-залежні графічні протоколи (команди графічних пристроїв) типу *TEKTRONIX*, *REGIS*, *HPGL*; мови опису сторінок типу *PostScript*, *PCL*; апаратно-незалежні графічні протоколи (метафайли) типу *NAPLPS*, *GKSM*, *CGM*, *DXF*, *WMF*; прикладні (проблемно-орієнтовані) графічні протоколи типу *IGES* (*Initial Graphics Exchange Specification*), *STEP* (*Standard for the Exchange Product Model Data*), *MAP* (*Manufacturing Automation Protocol*), *VDAFS* (він же *SSI – Sculptured Surface Interface*), *EDIF* (*Electronic Design Interchange Format*); растрові графічні файли.

**Графічні файлові формати.** Для ефективної роботи з графічними зображеннями важливо зробити правильний вибір одного з численних графічних файлових форматів, які призначені для занесення зображень до пам'яті ЕОМ, їх зберігання та передавання каналами зв'язку в найбільш компактному та цілісному вигляді. Різні графічні файлові формати реалізують різні способи опису графічної інформації в файлах зображень та різні технології їх компактного подання. Основними вимогами до них є економність та інформативність, одночасно виконати які на сто відсотків неможливо. Тому важливо знати, для досягнення якої мети є найбільш придатним конкретний формат: неправильно вибраний формат може зайняти надто великий обсяг пам'яті або призвести в процесі стискання

графічної інформації до неприпустимої втрати якості зображення. До найбільш відомих форматів графічних файлів належать: *BMP*, *GIF*, *PNG*, *JPEG*, *FLASHPIX*, *PCX*, *EPS*, *PDF*, *TIFF*, *RAW*, *VRML*, *MPEG*, *MJPEG*, *IFF*, *MAC*, *PICT*, *PixelPaint*, *PIXAR*, *Scitex CT*, *TGA* тощо [63].

**Графічний файловий формат *Bitmap* (*BMP*)** – стандартний, давній та традиційний растровий формат, який став одним із основних у системі *Windows* та який використовують на *DOS*- та *Windows*-сумісних комп'ютерах. *BMP*-файли без проблем передаються між комп'ютерами та обробляються. Оскільки дані не стиснені, то при виведенні на екран та на друк нема втрати якості. *BMP*-файл має просту структуру та зберігає єдине зображення з одним, чотирма, вісьма та двадцятьма чотирма біт/піксель. 1-, 4- та 8-бітове подання відповідають індексованому кольоровому зображенню. Для цих зображень у заголовку *BMP*-файлу зберігається таблиця колірності. Зображення може бути стиснене з використанням кодування довжин серій для 4- та 8-бітових зображень.

*BMP*-файли мають суттєвий недолік – займають забагато пам'яті: чим вища глибина кольору зображення (кількість розрядів кольоровості, зарезервованих для кожної точки), тим більший обсяг пам'яті займає файл. Якщо потрібно впакувати на жорсткому диску багато зображень, слід відмовитися від *BMP*-файлів: навіть невеликі кольорові зображення з дозволом 640×480 вимагають декількох *Mb*. Можна стискати *BMP*-файли за допомогою *RLE* (*Run-Length Encoding*), що змінює розширення імені на "*RLE*".

**Графічний файловий формат *CompuServe Graphics Interchange Format* (*GIF*)**, широко розповсюджений в *Internet*,

є одним із найвідоміших форматів зберігання та транспортування файлів растрових зображень. *GIF* використовують для подання індексованих кольорових зображень та *HTML*-документів в електронних мережах. Він є ущільненим форматом, розробленим для прискорення пересилання файлів телефонними лініями (використовується алгоритм стискання *LZW*, за допомогою якого в кольорових зображеннях завжди можна досягти ступеня стискання 3:1-5:1). *GIF* дозволяє містити в одному файлі декілька самостійних зображень та здійснювати анімацію.

*GIF* використовує найрозповсюдженішу модель – індексоване кольорове зображення. Але це не заважає зберігати зображення дворівневої та монохромної моделей, які реалізують шляхом настроювання таблиці колірності. А ось модель повнокольорового зображення зі зберіганням для кожного пікселя компонентів *R*, *G*, *B* не підтримано через обмеження у 8 біт/піксель. Безперечною перевагою є наявність стандартизованого протоколу передавання *GIF*-зображень лінією зв'язку. Оскільки *GIF* організовано за принципом послідовного доступу, то він не містить змістів або посилань, а тому є зручним для передавання зображень у розподіленій графічній системі.

Основні недоліки формату *GIF* такі: на перший погляд, *LZW*-стискання працює без втрат (декомпресований файл є "ідентичним" оригіналу до стискання), але в дійсності *GIF*-файли заносять до пам'яті лише 8 розрядів на точку зображення, що обмежує глибину кольору 256 рівнями (в порівнянні з можливістю графічних акселераторів відобразити до 16 млн. кольорів), тобто використання одного методу кодування зображень (*LZW*) обмежує сферу застосування формату *GIF*; оскільки

*LZW*-стискання захищене патентом, при його комерційному використанні постають юридичні проблеми. Графічні файли *GIF* донедавна були в *Usenet* найважливішим файловим форматом для обміну кольоровими зображеннями модемним зв'язком. Але вказані недоліки та поява нових, сучасних способів стискання призвели до того, що *GIF* поступово замінюють форматом *JPEG*.

**Графічний файловий формат *Portable Network Graphics (PNG)*** було розроблено як альтернативу *GIF* для подання зображень в електронних мережах: він зберігає всю кольорову інформацію та альфа-канали зображення, а також використовує для мінімізації обсягу файлу такий алгоритм ущільнення, що не призводить до втрати даних.

*PNG* користується успіхом завдяки таким перевагам: він забезпечує високі ступені стискання без втрати якості зображення, оскільки в процесі стискання не втрачається інформація; завдяки компактності графічних файлів, *PNG* надзвичайно підходить для обміну даних в *Internet*; *PNG* підтримує три цифрових типи зображень (8-розрядну палітру, 16-розрядну палітру *Grayscale* – до 65000 градаций сірого кольору, 48-розрядну палітру *True-Color* – 16 млн. кольорів), у той час як *GIF* підтримує тільки перший формат, а *JPEG* – тільки два останніх; велика глибина кольору в 48 розрядів робить *PNG*-файли цікавими для *DTP*-професіоналів, які хочуть працювати з компактними файлами та платформово-незалежно; *PNG* підтримує п'ять різних фільтрів стискання, що визначають обсяг файлу на жорсткому диску; *PNG* є вільним від ліцензування.

Основний недолік *PNG*-формату: алгоритм стискання для *PNG*-файлів працює правильно тільки тоді, якщо на диск заносять кольорові зображення з високою глибиною кольору.

**Графічний файловий формат *Joint Photographic Experts Group (JPEG)*** широко використовують для відображення фотографій та інших тонових зображень в електронних мережах.

На відміну від *GIF*, формат *JPEG* зберігає всю інформацію про колір у *RGB*-зображеннях. Він використовує ефективний алгоритм ущільнення, що дає значне скорочення обсягу файлу завдяки відкиданню надлишкової інформації, що не впливає на якість відображення. *JPEG* сильно стискає кольорові зображення (залежно від змісту зображення, економить від 50 % до 70 % обсягу пам'яті), причому око спостерігача не помічає втрати якості. Графічні файли займають мало місця на жорсткому диску та завдяки своїй компактності є пріоритетними для передавання телефонною лінією за допомогою модему. Більшість зображень як в *Internet*, так і в *Newsgroups* подано форматом *JPEG*.

У процесі *JPEG*-стискання деякі дані втрачаються. Але алгоритм стискання слідкує за тим, щоб у декомпресованому зображенні око не помітило втрати якості: він оцінює, які кольорові розбіжності між сусідніми блоками  $8 \times 8$  зображення є невидимими для ока, та відкидає інформацію про них. У разі відкриття *JPEG*-файлу, він автоматично розпаковується.

Несуттєвим недоліком *JPEG*-ущільнення є часткова втрата даних. Між якістю зображення та ступенем ущільнення існує зворотна залежність: чим вищу якість задати для підсумкового зображення, тим менш компактним буде упакований файл. У більшості випадків різниця між оригіналом та зображенням,

яке отримане в підсумку ущільнення з високою якістю, непомітна. *JPEG* має і серйозні недоліки: в монохромних зображеннях має місце суттєва втрата якості; для ліній та гострих ребер при стисканні виходять великі помилки квантування кольору над блоками  $8 \times 8$ , які заважають у процесі перегляду та друку зображення, відновленого після стискання.

**FlashPix (FPX)** – відносно новий графічний файловий формат, який набув великого успіху. Він є неоціненним для тих, хто займається видавничими системами, скануванням, цифровою обробкою зображень, фотографією, *HTML*, пересиланням файлів зображень телефонною лінією. Багато з тих фірм, які займаються ПЗ, розробляють настроювані пристрої для *FlashPix*. Файли *FlashPix* містять високоякісні фотографії з великою глибиною кольору та можуть бути швидко оброблені середнім мультимедіа-ПК або *Macintosh*. Не потрібно мати високопродуктивний ПК, щоб обробити фото або надрукувати їх на спеціальному папері: принтер переносить на папір те, що подане на екрані, без втрати якості та викривлень кольору.

Файл *FlashPix* побудований інакше, ніж інші файли зображень: він зберігає не тільки зображення оригіналу, а й копії з нижчим дозволом. Завдяки цьому графічні редактори можуть у процесі обробки зображень швидко обирати найкращий дозвіл, не опитуючи оригінальне зображення. Якщо кольорове зображення оброблюють на екрані, в копіях із низьким дозволом здійснюються зміни.

Розширення імені файлу *"\*.MIX"* означає, що формати тексту та *FlashPix* перемішані.

*FlashPix* являє собою інтерес для застосувань, які пересилають файли зображень телефонною лінією, оскільки зображення можна стискати майже без втрати якості.

**Графічний файловий формат ZSoft (PCX)** було розроблено для програми *PC PaintBrush*. *PCX* розповсюджений на *IBM PC*-сумісних комп'ютерах та використовується в графічних редакторах *PaintBrush* та *EgaPaint*, у системах підготування документації *Ventura Deck Top Publisher* та *First Publisher*. Оскільки *PCX* використовує *RLE*-компресію (вона дає низький коефіцієнт стискання, але швидке декодування), цей формат має переваги в інтерактивних системах зі швидким змінюванням зображень.

**Графічний файловий формат Encapsulated PostScript (EPS)** підтримують багато програм верстання та підготування ілюстрацій: він є найбільш переважним для більшості з них. Цей формат підтримує прозорий білий колір у бітовому режимі.

**Графічний файловий формат PDF** використовує програма *Adobe Acrobat* – основний засіб електронного расповсюдження документів фірми *Adobe* на платформах *Macintosh*, *Windows*, *Unix* та *Dos*. Формат *PDF*, розроблений на основі мови *PostScript Level 2*, можна використовувати для подання векторних та бітових зображень. У цьому плані *PDF*-сторінки ідентичні *PostScript*-сторінкам. Перевага *PDF*-файлів – у здатності містити елементи для пошуку та перегляду електронних документів, наприклад, гіпертекстові посилання на електронний зміст.

**Графічний файловий формат Tagged Image File Format (TIFF)** використовують для обміну документами між різними програмами та комп'ютерними платформами. Цей формат

підтримує ряд систем підготування документації. Він є найреальнішим претендентом на стандарт для зберігання та транспортування растрових зображень.

Основною концепцією *TIFF* є кольорова модель – набір характеристик зображення, що визначають спосіб подання кольору. Стандартизовані такі кольорові моделі: *bilevel image* (дворівневе зображення); *grayscale image* (монохромне зображення); *paletted color image* (індексоване кольорове зображення); *full RGB image* (повне кольорове зображення).

*TIFF* є відкритим форматом та дозволяє створювати будь-яку модель зображення, вибір якої залежить від конкретної задачі: дворівнева модель є найзручнішою для систем підготування документації; індексоване кольорове зображення придатне для формату зберігання графічної інформації в растрових графічних дисплеях тощо.

Формат *TIFF* містить не тільки інформацію про модель зображення, а й метричні характеристики зображення (його розміри, щільність у пікселях на одиницю довжини), важливі для систем підготування документації. Цей формат не накладає майже ніяких обмежень на параметри зображення. *TIFF* дозволяє зберігати в одному файлі: будь-яку кількість зображень; декілька копій одного зображення з різними характеристиками (наприклад, варіанти зображення з різною щільністю доцільно мати при роботі з різними принтерами у видавничих системах).

*TIFF* дозволяє зберігати зображення в нестисненому вигляді (це зручно в процесі створення зображення, для довільного доступу до будь-якого елемента за найшвидший час) та упаковувати зображення різними методами (одним із них є *LZW*).



**Графічний файловий формат Raw** призначений для передавання документів між програмами та комп'ютерними платформами. Він являє собою послідовність байтів, які описують усю кольорову інформацію. Піксель зображення описують у двійковому форматі: 0 відповідає чорному кольору, а 255 – білому.

**Графічний файловий формат Amiga Interchange File Format (IFF)** використовується для роботи з *Video Toaster* та обміну файлами з комп'ютерами *Commodore Amiga*, підтримується багатьма графічними програмами для *IBM*-сумісних ПК.

**Графічний файловий формат MacPaint (MAC)** є основним форматом у системах підготування документації та графічних редакторах на ПК фірми *Apple* (*Macintosh*, *Lisa*). Цей формат як правило використовують для передавання бітових зображень у програми системи *Macintosh* (розмірність зображень не повинна перевищувати  $576 \times 720$  пікселів).

**Графічний файловий формат PICT** широко використовують програми верстання та обробки графіки в системах *Macintosh* як загальний формат для обміну документами. Він є ефективним для ущільнення зображень з великими однокольоровими областями. Найбільш відчутний ефект отримують у разі ущільнення альфа-каналів, які містять великі області чорного та білого кольору. При зберіганні *RGB*-зображень у форматі *PICT* можна задавати дозвіл 16 або 32 біт/піксель. Для зображень у градаціях сірого дозвіл може становити 2,4 або 8 біт/піксель.

**Графічний файловий формат PixelPaint** дозволяє відкривати документи в програмі *PixelPaint* на комп'ютерах *Macintosh*. Цей формат використовують для зображень у градаціях сірого або з індексованими кольорами.

*Графічний файловий формат PIXAR* було розроблено для обміну файлами з графічними робочими станціями *PIXAR*, призначеними для професійних програм 3D-моделювання та анімації.

*Графічний файловий формат Scitex Continuous Tone (CT)* можна використовувати для: кольорових зображень у поданні *RGB* або *CMYK*; зображень у градаціях сірого. Його застосовують на комп'ютерах *Scitex*, які використовуються для високоякісної професійної обробки зображень.

Зображення в форматі *Scitex CT* є *CMYK*-файлами (часто — великого обсягу), що утворюються при введенні даних за допомогою *Scitex*-сканерів. Виведення файлів у форматі *Scitex CT* здійснюється на плівку на растровому пристрої *Scitex*, який виконує кольороподіл за допомогою запатентованої напів-тонової системи *Scitex*. Оскільки імовірність появи муару є малою, цю систему часто використовують для виконання професійної роботи з кольором (наприклад, при розробці рекламних оголошень у центральних журналах).

*Графічний файловий формат Targa (TGA)* розроблено для систем, що використовують відеокарту *Truevision*, та підтримано програмами обробки кольорів для *MS-DOS*. При зберіганні *RGB*-зображення, можна вказувати глибину кольору.

**Провідні поліграфічні цифрові стандарти передавання інформації.** *Тривалий час позиції галузевого цифрового стандарту для передавання інформації в поліграфії утримувала мова опису сторінок PostScript* (у виробництві майже всіх друкованих публікацій стали використовувати програмні додатки, що можуть здійснювати *PostScript*-виведення).

*PostScript* інтенсивно витісняють файловий графічний формат *Adobe PDF (Portable Document Format)* та система *Adobe Acrobat*, яка його підтримує: вони застосовуються майже на всіх етапах підготування та випуску друкованої продукції. Оскільки *PDF* побудовано на тих самих принципах, які було закладено до *PostScript*, то основна частина наявного обладнання та ПЗ повністю відповідає новій технології виробництва на базі *PDF*, гарантуючи швидкий перехід до нього. Розглянемо переваги *PDF* у порівнянні з *PostScript*.

Основою програми *Acrobat* є програмні додатки (елементи технології *PDF*) *Acrobat Distiller*, *Acrobat Exchange*, *Acrobat Catalog* та *Acrobat Reader* (для читання *PDF*-файлів).

*Acrobat Distiller* є засобом для растрівання та створення *PDF*-файлів, у основу якого покладено *Adobe CPSI (Configurable PostScript Interpreter)*. Оскільки програми виводять *PostScript*-файли з відхиленнями через власне прочитування специфікацій *PostScript*, *Distiller* відіграє важливу роль на такому етапі додрукарського підготування, як растрівання. Він виконує попередню перевірку документу перед виведенням (*preflight*) – здійснює нормалізацію початкового *PostScript*-файлу, на виході якої отримують надійніше та стабільніше (передбачуваніше в поведінці) виведення, що відповідає початковим специфікаціям *PostScript*. Завдяки цьому можна виконувати роботу силами клієнта (він може перевірити *PDF*-файли, підготовані для відправки на фірму, роздрукувавши їх на власному принтері), з першого разу отримуючи правильний результат.

**Завдяки властивості передбачуваності, *Acrobat (PDF)* є ідеальним інструментом для перевірки перед виведенням:**

створення *PDF*-файлу одночасно є перевіркою перед виведенням; якщо на момент створення файлу будуть відсутні необхідні елементи, користувач отримає попередження; створювати та відправляти до бюро додрукарського підготування та до типографії *PDF*-файли можуть і повинні самі клієнти; застосування *PDF*-формату, як основного стандарту на всіх етапах роботи, знижує навантаження на бюро додрукарського підготування (вони отримують один єдиний файл, текст не пливе, *EPS*-файли не губляться, помилки *PostScript* не впливають).

*Acrobat Exchange* приймає на вході та відкриває *PDF*-файл, пропущений крізь *Distiller*. У процесі роботи можна сформува-ти як простий друкований документ, так і значно досконалі-ший, оскільки *Exchange* дозволяє: вносити до готового файлу мінімальні правки; виконувати підсумкове форматування; визначати як звичайний порядок читання статей та документів, у природній послідовності подання їх тексту, так і спеціальний порядок; вбудовувати до *PDF*-файлу гіпертекстові посилання на окремі фрагменти документу, інші *PDF*-файли, вузли *WWW*, звукове супроводження, відеофільми в форматі *QuickTime*; ви-користовувати надійну схему для підключення додаткових *plug-in* модулів сторонніх фірм. **Можна створити великий *Internet*-каталог у форматі *PDF* із посиланнями на *Web*-сто-рінки та локальні *PDF*-файли.** Завдяки наявним на *Web*-сай-тах засобам *Byte-Serving*, можна подорожувати по *PDF* мере-жею *Internet* (ці засоби забезпечують попереднє швидке отри-мання та перегляд першої сторінки *PDF*-файлу). Можна вклю-чати до *PDF* посилання на мультимедійні файли зі звуком та відео, що робить *Acrobat* інструментом підготування мультиме-діа-додатків та дозволяє додавати готовим друкованим матеріа-лам певної оригінальності.

*Acrobat Catalog* швидко працює та надає можливість створювати для документів алфавітні вказівники та здійснювати ефективний пошук інформації в PDF-файлах: легко та швидко здійснює аналіз тексту, що містить окремий PDF-файл, каталог або компакт-диск PDF-файлів, складаючи по ньому алфавітний вказівник і каталоги та споряджаючи його посиланнями; забезпечує гарні можливості пошуку інформації за допомогою *plug-in Verity*, який можна застосовувати в *Internet* та публікаціях на *CD*. Робота модуля *Verity* відрізняється від команди *Find* програми *Exchange* більшою швидкістю та глибиною пошуку: швидкість у сотні разів вища; можна зібрати до одного каталогу алфавітний вказівник за декількома сотнями PDF-файлів (*Find* обмежується тільки відкритим та активним PDF-документом). Програма *Catalog* працює вправно, а після того, як буде переглянуто весь обсяг PDF-файлів, пошук у них стає швидким та легким.

Проаналізуємо переваги PDF для вирішення проблеми віддалених перевірки та редагування (цифрові методи вимагають забезпечення інтенсивного обміну інформацією величезних обсягів, чого не можуть надати комунікаційні технології, наприклад, факсимільна технологія забезпечує чорно-біле виведення з таким дозволом, за якого не розрізнити дрібні деталі): можна відправляти PDF-файли з супроводжувальними записками електронною поштою, зробивши їх невеликого обсягу; програма *Acrobat Exchange* надає можливість вставляти до файлу анотації типу наклеюєк "Post-IT", які часто застосовують в офісах; на будь-яку частину PDF-файлу можна "наклеїти" записку з описом тих змін, які необхідно внести до неї; отримавши копію PDF-файлу, можна вивести її на свій

локальний принтер з високим дозволом (*PDF* відіграє роль кольорового редагованого *PostScript*-факсу з високим дозволом, а якість відтворення зображення залежить від принтеру на приймальній стороні).

*Залежно від постановки задачі, PDF можна використовувати як інструментарій для: додрукарського підготування; Web-"будівництва"; створення CD.* Слід використовувати гнучкість та керованість *PDF* для отримання результатів, які є оптимальними для конкретного застосування: у разі створення *PDF*-файлу для друку на офсетній машині, доцільно зберегти максимальний дозвіл растрових зображень; *PDF*-файл для публікування в мережі *Internet* повинен мати мінімальний обсяг (наприклад, шляхом переведення зображення в дозвіл броузеру, застосовуючи *JPEG*-компресію) тощо. У разі використання в додрукарських процесах технології *OPI*, перехід на *PDF*-технологію не викликає ускладнень. Оскільки *Adobe* тісно інтегрувала *PDF*-формат до останніх версій *PostScript*, приймати замовлення в форматі *PDF* вигідніше та безпечніше.

*Головною перевагою PDF над іншими електронними засобами публікації є унікальна можливість надавати однакове виведення на будь-яких комп'ютерній платформі та принтері* завдяки тому, що всі *PostScript*-шрифти, описи сторінок та зображення вбудовують безпосередньо до файлу.

Крос-платформова сумісність *PDF* дозволяє організаціям використовувати *Internet* для видання своїх матеріалів набагато успішніше, ніж із застосуванням формату *HTML*. Справа в тому, що *Internet*-документи у *HTML*-форматі залишають вибір їх подання на екрані залежним від того настроювання

параметрів броузеру кінцевого користувача, яке, як правило, користувачі залишають по умовчанняю. Те, з чим звикли мати справу дизайнери та художники, у сенсі контролю над смаком та кольором, значно відрізняється від тих можливостей керування, що надає формат *HTML*.

Підтримкою розповсюдження у видавничій справі формату *PDF* є розвиток технологій струминних принтерів: різниця в якості між традиційним та струминним друком швидко зменшується, тому кінцевий користувач у себе вдома може друкувати сторінки, що майже ідентичні офсетним та є дешевими. *Acrobat* виконує для струминного принтеру інтерпретацію *PostScript* нижнього рівня. Для більшості тих принтерів, які не мають вбудованих *PostScript*-можливостей, програми *Acrobat Reader* та *Exchange* дозволяють виводити на них так, немовби вони є *PostScript*-принтерами, з максимальним для них дозволом.

**В цілому** до основних переваг *PDF* належать: автоматична перевірка на відсутність помилок та наявність зв'язків у ході інтерпретації; можливість кросплатформового використання в звичайному та цифровому друці, в *Internet* та в разі запису на *CD*.

**Стандарт *Virtual Reality Modeling Language (VRML)* для 3D-зображень у *World Wide Web*.** Мову моделювання віртуальної реальності *VRML* було створено в 1994 році консорціумом фірм на чолі з *Silicon Graphics* як *plug-in* для *Internet*-броузерів. На даний час *VRML* визначає: нову концепцію використання *WWW*, що трансформує її в середовище, схоже на реальний світ (тривимірний, емпіричний, інтерактивний, зв'язний, протяжний), а також, випереджаючи на рівень *HTML*, робить *Internet*

та *WWW* доступними мільярдам (завдяки більшій природності роботи в *3D*-світах замість *2D*-вікон); стандарт, згідно з яким віртуальні світи пересуваються інформаційними каналами, та платформово-незалежний файловий формат для опису *3D*-об'єктів у *WWW*; спосіб, яким можна легко відображати ландшафти та оточуючі світи (авторських систем для побудови *3D*-сцен в *Internet* достатньо; файл *VRML* складається з тексту *ASCII*, подібно до *HTML* при відображенні *Web*-сторінок) [8].

Авторами *VRML 1.0* є М.Піц та Т.Парізі, а його наступних версій – група фахівців *VRML Architecture Group (VAG)*.

Коротко опишемо нові можливості, надані *VRML* [49].

По-перше, це просторова організація інформації (*VRML* відіграє роль навігатора в *WWW*, яку органічно вписано до просторової моделі світу). В основу цього підходу покладено те, що топологія об'єктів довкілля (робочого кабінету, письмового столу тощо) позитивно впливає на орієнтування, згадування та знаходження потрібної інформації.

По-друге, це надання, завдяки впровадженню третього виміру, нескінченної додаткової робочої поверхні, з природним пересуванням уздовж неї. Це усуває ряд проблем традиційного розташування інформації на площині: на *2D*-екрані з обмеженою кількістю пікселів неможливо вмістити всю потрібну інформацію, а тому доводиться погоджуватися на перетин, перекриття, пересування та тимчасове усунення інформаційних ділянок (наприклад, це проблема множини вікон, які заважають одне одному, та обмеженості розмірів вікна).

По-третє, це потужний інструментарій для візуалізації: простір організовано так, як він виглядає в дійсності; *3D*-моделі дають ті уявлення про об'єкти, що важко отримати за фотогра-



фіями; просторова візуальна організація даних полегшує пошук потрібної інформації. На практиці цей інструментарій: є неоціненним для агентів з нерухомості; економить час на таких заходах, як відвідування книгарень, бібліотек, екскурсбюро; зменшує втрати часу на навігацію в погано структурованих БД.

Ключовими напрямками розвитку *VRML* є: моделювання 3D-світів із асоційованою поведінкою; інтеграція змісту все-світньої мережі; анімація та моделювання поведінки предметів та дійових осіб; оптимізація 3D-навігації в *Internet*; 3D-ігри та розваги по *Internet*; середовища для багатьох користувачів.

Галузями практичного застосування *VRML* є: спілкування, навчання, торгівля, маркетинг, організація та візуалізація даних у фінансах та статистиці, наукова візуалізація, індустрія розваг. Найсерйознішими застосуваннями *VRML* є розробки для тривимірних БД, наприклад: *Knowledge Browser* – розробка центру геофізичних досліджень (*GFZ*) та вищого навчального закладу в Потсдамі, системи *Lyberworld* товариства математики та обробки даних (*GMD*) та *WebViz* університету Мінесоти.

**Стандарти для цифрового кодування відеозображень.** У сфері цифрового відеомонтажу існують суттєві відмінності між відео високої якості мовлення для телебачення (до нього висувають жорсткі вимоги та на протязі багатьох років розробляли професійні стандарти) та можливостями реалізації відео на ПК. Через ці відмінності, визначення та характеристики цифрового відео якості мовлення суттєво відрізняються від тих, які є прийнятими серед комп'ютерних професіоналів [90].

Користувачі ПК добре знайомі з терміном *MPEG*, з яким пов'язана можливість перегляду повнометражних повноекранних відеофільмів. Комп'ютерне відео стало звичним на до-

машніх ПК завдяки: зниженню вартості плат для програвання *MPEG*; здатності комп'ютерів *Pentium* впоратися з задачею за рахунок ПЗ; розвитку технології *MMX*.

**За розробку стандартів серії *MPEG* для стиснення відео/аудіо даних відповідає "Експертна група з рухомих зображень" (*Motion Picture Expert Group*) організації *ISO* [2].**

*MPEG-1* було розроблено для доставки відео на *CD-ROM* (швидкість потоку даних  $\approx 1,5$  Мбіт/с). Він отримав широке розповсюдження завдяки дискам *VideoCD*.

*MPEG-2* призначено для роботи з відео якості мовлення (швидкість потоку даних – 3-15 Мбіт/с). Його широко використовують у цифровому телемовленні та в кодуванні відеоматеріалів для дисків *DVD-Video*.

*MPEG-3* ухвалив розширення меж *MPEG-2* у напрямку забезпечення уніфікації в компресії потоків даних зі швидкостями 20-40 Мбіт/с для телебачення високої чіткості (*HDTV*).

*MPEG-4* став у світі мультимедіа головною діючою особою – претендентом на роль єдиного концептуального способу опису, подання та обробки мультимедіа-даних. *MPEG-4* підпорядкував собі *MPEG-1* та *MPEG-2* (їх галузі застосування є втіленнями способів кодування одного з численних типів даних, передбачених *MPEG-4*) [6].

*MPEG-4* задає принципи роботи з цифровим поданням медіа-даних (контентом) для таких галузей: інтерактивного мультимедіа (наприклад, того, що розповсюджують на оптичних дисках та крізь мережу); графічних застосувань, які являють собою синтетичний контент; цифрового телебачення (*DTV*).

*MPEG-4* задає правила організації об'єктно-орієнтованого середовища, оперуючи не тільки потоками та масивами медіа-

даних, а й таким ключовим поняттям, як медіа-об'єкт. Медіа-об'єкти класифікують: за типом поданих ними медіа-даних (аудіо-, відео-, аудіо-візуальні, графічні, текстові); за природністю (природні, тобто записані, відзняті, відскановані, або синтетичні); за структурою (елементарні та складені). Прикладами медіа-об'єктів є: нерухомий фон; відеоперсонажі на прозорому фоні; мова, синтезована на основі тексту; музичні фрагменти; 3D-модель, яку можна рухати та обертати в кадрі; анімований спрайт тощо.

Кожен медіа-об'єкт має систему координат, за допомогою якої ним керують у просторі та часі. Медіа-об'єкти можуть бути потоковими. Кожен медіа-об'єкт має набір дескрипторів, де задані: властивості об'єкту; операції декодування асоційованих з об'єктом поточкових даних; розташування в сцені; поведінка та припустимі реакції на вплив з боку користувача.

Медіа-сцену описують ієрархічною структурою, вузлами якої є медіа-об'єкти та яку можна динамічно перебудовувати. Сцена має власну систему координат. У процесі розташування об'єкту в сцені, здійснюється приведення його локальної системи координат до системи координат сцени або старшого за ієрархією об'єкту. Звукові об'єкти теж можуть мати та динамічно змінювати координати в просторі сцени, що дає стерео-ефекти та *Surround*-ефекти. Об'єкти та сцена можуть бути наділені: поведінкою; контрольованим рівнем композиції в процесі візуалізації сцени (колір об'єкту, характер звуку тощо).

*MPEG-4* визначає: правила кодування об'єктів; їх ієрархії та способи композиції в процесі побудови сцени; методи взаємодії користувача з об'єктами всередині сцени. У *MPEG-4* визначено двійкову (компактну та швидко в інтерпретації) мову

*BIFS* опису об'єктів, класів об'єктів та сцен, яку характеризують як розширення *C++* та багато з концепцій якої було запозичено у *VRML*. Команди мови *BIFS* дозволяють: анімувати об'єкти; змінювати координати, розміри та властивості об'єктів; задавати поведінку об'єктів та їх реакцію на вплив користувача; змінювати властивості середовища; змінювати та поновлювати сцену; виконувати геометричні побудови тощо. **Група MPEG та Web 3D Consortium продовжують роботи зі зближення MPEG та VRML.**

*MPEG-4* не тільки затверджує усталену практику – він є прогресивним стандартом: упереджувальним, структуроутворювальним та фундаментальним законом, який створює основу для виробництва, розповсюдження контенту та способів доступу до нього в новому єдиному цифровому середовищі; таким, що відкриває принципово нові можливості для авторів, дистриб'юторів та споживачів контенту. На відміну від *MPEG* версій 1 та 2, які для користувачів є чорними ящиками з функціями стискання та розпаковування, *MPEG-4*: безпосередньо впливає на спосіб мислення та роботи фахівців; вимагає вивчення своїх численних потенційних можливостей.

З *MPEG-4* працює широке коло користувачів: програмісти; розробники декодерів; ті, хто створює інтерактивну мультимедіа-продукцію; телевізійні автори та ті, хто здійснює мовлення; мережні провайдери; користувачі домашніх ПК.

Інтереси *MPEG* є ширшими за проблеми стискання медіаданих, у процесі вирішування яких організація *MPEG* досягла великого успіху в примиренні підходів та інтересів провідних корпорацій та індустрій. Ці інтереси пов'язані: з розвитком процесів інтегрування телебачення та *Internet*, ПК та розва-

жальних приставок і плеєрів; з тим, що медіа-контент, отриманий із цих джерел, усе більше стає інтерактивним. Це вимагає нових стандартів, які б допомогли: розробникам контенту – донести їх вироби до користувачів максимальною кількістю способів; користувачам – отримувати зі свого пристрою доступ до медіа-контенту в будь-якій його формі.

У *MPEG-4* зібрано, узагальнено та поєднано в якісно нову технологію досягнення в різноманітних самостійних галузях та технологіях: *QuickTime*, *VRML*, 3D-графіка, інтерактивна персонажна анімація (*Macromedia Director*), розробка відеоігор, відеокмпозитінг, телемовлення, потокові відео та звук. У рамках подальшого розвитку *MPEG* за вказаними вище напрямками, було розроблено стандарт *MPEG-7* під офіційною назвою *Multimedia Content Description Interface*.

**Виконаємо порівняльний аналіз *MPEG* з *AVI* та *Motion JPEG*, *QuickTime*, *Indeo*.** На відміну від формату *AVI* (алгоритм стискування – *Motion JPEG*), *MPEG*-файли займають менше місця (для оцифрування та редагування однієї хвилини відео з якістю *VHS* потрібно 100-200 *Mb* на жорсткому диску, а за допомогою *MPEG*-стискування можна під монтаж півгодинного ролику зайняти лише 1 *Gb*). З іншого боку, на відміну від *AVI*, *MPEG* складно редагувати, та й коштують системи для *MPEG*-оцифрування більше за звичайні плати введення відео.

*QuickTime* та *Indeo* – це розробки компаній *Apple* та *Intel*, які орієнтовані на ринок мультимедіа та не підтримують ряду технологій, необхідних для професійного застосування (наприклад, повноекранного відтворення з якістю мовлення, оцифрування зображень із повною розгорткою по два поля

на кадр). *QuickTime* передбачає коефіцієнт стискування 50:1, *Indeo* – до 10:1, у той час як *MPEG* – 200:1, досягаючи вищої якості. До того ж тільки *MPEG* може бути реалізований як програмно, так і апаратно на обох платформах – *PC* та *MAC*.

*MPEG* є стандартом-лідером для повнометражного цифрового відео. Виключенням є сфера нелінійного цифрового монтажу, де більш розповсюджено *Motion JPEG* (зі збільшенням на ринку кількості кодувальних *MPEG*-систем, і цю сферу опановує *MPEG*). З кожним роком з'являються нові сфери застосування *MPEG*-технології, починаючи з високоякісних цифрових *DVD*-відеодисків, новітніх ігрових систем та завершуючи доскональними цифровими комплексами – для мовлення та монтажними.

## Питання та завдання для самоперевірки

1. У чому полягає головна концепція стандартизації в комп'ютерній графіці та які міжнародні організації встановлюють графічні стандарти?
2. Надайте класифікацію графічних стандартів.
3. На прикладі обраного вами графічного зображення (наприклад, зробленого вами цифрового фото або відсканованої ілюстрації), дослідіть на практиці особливості його подання за допомогою найпопулярніших графічних форматів (*BMP*, *JPEG*, *GIF*, *TIFF*, *CDR*, *FLASHPIX* тощо).

## 4.2. Склад та функції графічного ПЗ

Розвиток уявлень про оптимальний склад, функції та взаємодію базового та прикладного графічного ПЗ тісно пов'язаний із активною діяльністю з його стандартизації. На першому етапі розвитку засобів комп'ютерної графіки увага акцентувалася на апаратному забезпеченні та графічних системах, орієнтованих на конкретне обладнання [133]. На другому етапі увага акцентувалася на створенні ПЗ: розробці алгоритмів машинної графіки для генерації примітивних елементів, інтерполяції та апроксимації; створенні форм та методів подання зображень; створенні інструментальних засобів комп'ютерної графіки (графічних мов, пакетів процедур, мов діалогу). ПЗ стало промисловим виробом, виникла індустрія ПЗ. Це висунуло на перший план задачу стандартизації графічного ПЗ, значення якої підсилив розвиток комп'ютерних мереж з різнотиповими термінальними пристроями, що вимагало незалежності ПЗ від апаратури [120].

Відпрацьовування ключових напрямків розвитку стандартизації ПЗ повинне запобігти таким негативним наслідкам: не повинен стримуватися розвиток нового, нестандартного АЗ та ПЗ; намагання охопити якнайширший спектр застосувань не повинне призвести до громіздкості структур даних, набору засобів, програмного коду; намагання досягти легкої адаптованості не повинне призвести до надлишковості засобів запиту до оточення.

Зараз роботи зі стандартизації ПЗ націлено на: специфікацію мінімального набору базисних функцій; забезпечення багатофункціональності пакетів графічних підпрограм; підви-

щення функціонального рівня стандартизації в конкретних галузях застосувань.

Для підвищення ефективності прикладних програм потрібен набір різних функціональних можливостей з різних рівней, запропонованих стандартами. Цікавим є таке рішення, коли графічну систему моделюють як сукупність слабо пов'язаних підмножин засобів формування зображень, проміжного зберігання інформації, введення, перетворень зображень та керування графічними пристроями. Потрібної конфігурації системи досягають шляхом статичного або динамічного збирання з окремих модулів конвеєру, з передаванням даних крізь єдиний міжмодульний інтерфейс.

Перші підсумки стандартизації концепцій про структуру та функціонування ПЗ були отримані в рамках розробки протоколів для апаратно- та програмно-незалежного подання графічних даних у мережі *ARPA* – це був *Network Graphics Protocol (NGP)*.

Модель роботи користувача в мережі з застосуванням графічного протоколу можна описати так. Користувач ініціює термінальну програму, призначену для: інтерпретації протоколу до апаратно-залежної форми; реалізації функцій введення та запиту до оточення; встановлення зв'язку з ЕОМ мережі, на якій буде надане необхідне обслуговування. Прикладна програма головної ЕОМ, за необхідності виконати введення/виведення, використовує обслуговувальну програму, що виконує функцію інтерфейсу з мережним графічним протоколом. Оскільки дані мережею передають у стандартній формі, то на передавальному боці потрібно виконувати кодування, а на приймальному – декодування інформації.



Протокол виведення має такі рівні: сегментований формат, коли зображення побудоване зі списків графічних примітивів, а термінальна програма має бути простою та виконувати перекодування; структурований формат, коли зображення побудоване з викликів підкартин, обсяг даних зменшується, але на термінальній ЕОМ необхідне програмне виконання перетворень.

Перелічені далі методи використовують у роботі з віртуальними пристроями введення: виконання запиту та отримання стану пристрою введення; надання користувачу дозволу здійснити дії, що приведуть до події введення, та отримання на головній ЕОМ повідомлення про неї.

Апаратну незалежність забезпечують: засоби опитування, що дозволяють ідентифікувати конфігурацію та можливості використаних пристроїв; адаптація до цих можливостей шляхом виконання налаштувань прикладної та обслуговувальної програм.

Роботи зі створення протоколів стали початком інтенсивної діяльності зі стандартизації. Було сформульовано основну мету стандартизації в комп'ютерній графіці – переносимість графічних систем, якої досягають стандартизацією інтерфейсу між графічним ядром системи (базовою графічною системою, що реалізує власне графічні функції та є незалежною від обчислювальних систем, мов програмування, галузі застосування та графічних пристроїв) та між моделювальною системою (проблемно-орієнтованою прикладною програмою, що використовує функції графічного ядра).

Структура прикладної графічної системи, що відповідає цим вимогам, може бути описана шестирівневою моделлю (рис.4.1).

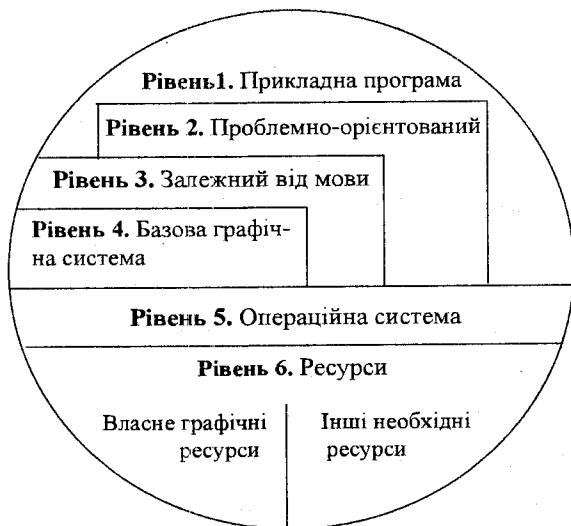


Рис 4.1. Шестирівнева модель структури прикладної графічної системи

Процеси перетворення координатної інформації в графічній системі підпорядковані певному алгоритму. У разі виведення, ці процеси складаються з наведених далі етапів, а в разі введення від координатних пристроїв є зворотними. Етап 1 – модельні перетворення: проблемно-орієнтований рівень формує з геометричних моделей об'єктів, заданих у власних координатних системах, опис сукупного об'єкту в єдиній світовій системі координат, який надходить до графічної системи. Етап 2 – нормалізовувальні перетворення: графічна система переводить опис сукупного об'єкту в систему нормалізованих координат з фіксованими межами змінювання. Етап 3 – перетворення сегментів: необхідні тоді, коли графічна система надає засоби маніпулювання окремими підкартинами зображення (сегментами). Етап 4 – видові перетворення: формують опис

зображення в апаратно-незалежній формі, придатній для виведення на будь-який пристрій (у разі 3D-сцени та 2D-пристрою виведення проєктують зображення на задану картинну площину; в разі 2D-сцени та 3D-пристрою виведення виконують перетворення, що веде до потрібного 3D-розташування зображення; в процесі виконання цих перетворень може стати необхідним відсікання частин зображення). Етап 5 – перетворення робочої станції: переведення даних з апаратно-незалежної форми до координат конкретного пристрою для подальшого виконання операції виведення на цей пристрій.

Архітектуру переносимої графічної системи подано на рис.4.2: верхній рівень стандартизації (*IGES*) забезпечує мобільність компонентів графічної системи; середній рівень (*GKS*) – робить базову графічну систему незалежною від галузі застосування (він визначений вибором базових функцій системи); нижній рівень (*CGI*) – робить базову графічну систему апаратно-незалежною (він залежить від вибору примітивів введення/виведення, що є абстракцією можливостей пристроїв).

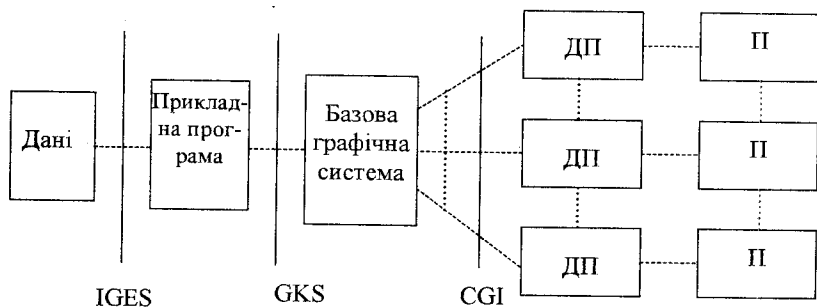


Рис.4.2. Архітектура переносимої графічної системи  
(ДП – драйвер пристрою, П – пристрій)

Незалежність від обчислювальних систем та мов програмування забезпечено дисципліною програмування та взаємодії з системами програмування.

### Питання для самоперевірки

1. Поясніть поняття базового та прикладного графічного ПЗ.
2. З яких етапів складається процес перетворення координатної інформації в графічній системі?
3. Подайте структурну модель ПЗ прикладної графічної системи.

### 4.3. Стандартні API для роботи з графікою

**Загальні прогресивні концепції функціонування графічних програмних інтерфейсів.** Традиційно взаємодія з 3D-акселераторами на низькому рівні полягала в прямому програмуванні відеопристрою та його акселераторів. У разі появи відеопристроїв від нових виробників, це викликало проблеми: з кожним відеопристроєм доводилося працювати відповідно його специфіці. По-перше, це незручно для програміста: якщо невідоме середовище використання програми, доведеться враховувати всі особливості численних відеоплат, а це найчастіше неможливо. По-друге, це прискорює старіння ПЗ: технічне забезпечення розвивається набагато швидше за

програмне, а з новими відеопристроями старі програми працювати не можуть.

Вирішенням проблеми стала розробка технологій *Direct-Draw*, *Direct 3D*, *OpenGL*, *Glide API*, *QuickDraw3D* та інших, ключова концепція яких полягає в створенні єдиного програмного інтерфейсу для виконання різноманітних графічних маніпуляцій, принципи роботи з яким не залежать від використаних пристроїв.

У цих технологіях реалізовано триелементну систему "користувач – інтерфейс – пристрій": заміна одного відеопристрою іншим вимагає тільки заміни деяких драйверів інтерфейсу; користувач, нічого не помічаючи, продовжує взаємодіяти з інтерфейсом, який працює з іншим відеопристроєм.

Таким чином, ми підійшли до розуміння терміну графічних *3D-API* (*Application Programming Interface*, інтерфейсів програмних додатків), що є *3D*-бібліотеками для полегшення та прискорення розробки прикладних програм з *3D*-графікою (об'ємними об'єктами, ефектом перспективи, *3D*-деформаціями, модифікаціями та трансформаціями, атмосферними ефектами, реалістичним освітленням *3D*-сцени тощо) шляхом задавання системи загальних методів обробки графіки ПЗ [91,108].

Робота з графічними *3D-API* не вимагає програмування тих *3D*-можливостей, що підтримані стандартними функціями конкретної бібліотеки (достатньо звернутися до потрібної функції, задати перелік її параметрів та отримати очікуваний *3D*-ефект).

Багато графічних *3D-API* працюють гнучко (як з *3D*-акселераторами, так і з ЦП), досягаючи потрібного *3D*-результату

навіть у разі відсутності акселератору або очікуваних від нього функцій, шляхом перекладання задачі на ЦП.

**Порівняльний огляд найпопулярніших API для роботи з 3D-графікою.** Найпопулярнішими графічними 3D-бібліотеками є *Direct3D*, *OpenGL*, *Glide API* та *QuickDraw 3D*.

На PC широко розповсюджені API 3D-графіки *Direct3D*, *OpenGL* та *Glide API*. Кожна з них, завдяки певному набору позитивних якостей та недоліків, завоювала свою сферу практичного застосування: *Direct3D* та *Glide API* є загальноприйнятими API для 3D-ігор; *OpenGL* є загальноприйнятим API для CAD/CAM (професійного 3D-моделювання та САПР).

Завдяки безперечним перевагам API *OpenGL* (простоті, доступності для вивчення, потужності тощо), його часто використовують в ігровій індустрії.

**Графічна бібліотека *Direct3D* з набору бібліотек *Microsoft DirectX*** є прошарком між програмним додатком та 3D-акселератором. Набір *DirectX* постачають з *Windows* та окремо – з відеоадаптерами та дистрибутивами програмних додатків багатьох виробників АЗ та ПЗ. *DirectX* є стандартним API для розробки мультимедіа-додатків та ігор під *Windows*.

З точки зору розробника ПЗ, *DirectX* – це базовий набір інструкцій та компонентів, які дозволяють: виконувати мультимедіа-додатки та ігрові додатки на будь-якому ПК під *Windows*; створювати сучасне ПЗ, не піклуючись про сумісність з АЗ.

*DirectX* повинен: забезпечувати взаємодію АЗ та ПЗ, що постійно ускладнюються; надавати той рівень продуктивності, якого б досягли розробники ПЗ під конкретні пристрої; бути

для розробників простим у використанні, не збіднюючи можливостей використання функцій та ефектів.

З точки зору користувача ПК, *DirectX* підтримує: маніпулятори (*DirectInput*); голосові та звукові ефекти (*DirectSound*); відтворення музики (*DirectMusic*); 2D- та 3D-графіку (*DirectDraw* та *Direct3D*); можливість гри в *Internet* (*DirectPlay*).

Фірма *Microsoft* постійно вдосконалює *DirectX*, додаючи нові функції та оптимізуючи їх (забезпечуючи швидші та якісніші 3D-розрахунки, використовуючи нові системи команд, які суттєво полегшують програмування на цьому API тощо).

*Direct3D* є визнаним стандартом індустрії розваг. Значною перевагою *Direct3D*, завдяки правильній маркетинговій політиці *Microsoft*, є тотальна сумісність з усіма графічними 3D-акселераторами та великою кількістю 3D-ігор. *Direct3D*, на відміну від *OpenGL*, практично не потребує настроювання.

Якість та швидкість генерації 3D-зображень, легкість програмування за допомогою API *Direct3D* поступаються потужнішому 3D-стандарту *OpenGL*.

**Графічна бібліотека *OpenGL* є розробкою компанії *SGI***, що є провідним виробником високопродуктивних графічних станцій. Нею було створено бібліотеку *Graphics Library (GL)* для візуалізації 3D-сцен, яку використовували програмні додатки, що працювали на графічних станціях *SGI*. Програмісти *SGI* першими зрозуміли необхідність заміни множини окремих рішень стандартним API для 3D-графіки. Тому *SGI*: вдосконалює програмне та апаратне забезпечення, що підтримують *OpenGL*; створила консорціум *Architecture Review Board (ARB)* з компаній, що підтримують стандарт *OpenGL*.

*OpenGL*, на відміну від *Direct3D*, є міжплатформовим стандартом: це суттєво полегшує перенесення програм на платформи *PC*, *SGI*, *Mac* та інші.

Для платформи *PC*, підтримку якої здійснює *SGI*, є достатня кількість *OpenGL*-прискорювачів, щоб стимулювати розробників до перенесення на неї *3D*-додатків.

*OpenGL* є потужною, легкою в застосуванні, добре відлагодженою графічною бібліотекою, з використанням якої написано багато програмних додатків (основною їх частиною є професійні пакети *3D*-графіки).

Вагомими перевагами графічної бібліотеки *OpenGL* є: переносимість (незалежно від *ОС* та графічної оболонки, *OpenGL*-додатки на *OpenGL*-сумісному обладнанні дають однакову візуалізацію); стабільність (завдяки контролю за всіма доповненнями та їх своєчасному анонсуванню); зручність використання (завдяки гарній структурованості та зрозумілості для програмістів, можливості використання численних відлагоджених *OpenGL*-програм, відсутності необхідності для розробників знати специфіку обладнання); масштабованість (оскільки *OpenGL*-додатки виконуються на різних системах, від *ПК* до суперкомп'ютерів, вони легко переносяться на будь-яку платформу) [15].

Завдяки цим перевагам, майже всі професійні *3D*-прискорювачі призначені переважно для роботи з *OpenGL*.

Для взаємодії *OpenGL* з конкретним відеоакселератором, окрім спеціальних команд, програмі потрібен драйвер до цього акселератору та спеціально написаний порт.

*OpenGL*-порт можна реалізувати двома способами: як *Installable Client Driver (ICD)* – високооптимізований драйвер



низького рівня, що забезпечує максимальну швидкодію на апаратному рівні, але є складним у програмуванні через необхідність реалізувати всі функції *OpenGL*; як *Mini Client Driver (MCD)* – програмний порт, який дозволяє підмінити ті фрагменти коду, які розробник хоче оптимізувати для конкретного обладнання.

Професійні плати мають власні *ICD*-драйвери, а складність їх програмування обумовлює зростання ціни цих плат.

*MCD* значно поступається *ICD* за швидкодією. Якщо ж *3D*-програма використовує таку функцію, що не має в *MCD*-драйвері апаратної підтримки, то апаратне прискорення буде відключене зовсім.

Компанія *Microsoft* створила для *OpenGL* *MCD*-порт, який задіює основні можливості цієї бібліотеки та здатен повільно функціонувати без використання акселератору, на ресурсах ЦП.

Перевагою цього *MCD*-порту є можливість легкого сполучення з драйверами низького рівня, а основним недоліком – неможливість для виробників акселераторів досягти максимального прискорення додатків (його головним призначенням є прискорення однієї складової частини *3D*-обчислень – остаточної растеризації зображення в процесі виявлення загороджених точок *3D*-сцени та видалення невидимих частин поверхонь тощо).

Компанія *Microsoft* вважає, що *OpenGL* має бути стандартом для професійних *CAD/CAM*-застосувань та *3D*-анімації, а для розробників ігор ідеалом є *Microsoft Direct3D*.

На думку багатьох експертів та розробників ігор, *OpenGL* та *Direct3D* мають однакове призначення, більше того – *OpenGL* краще підходить для створення ігрових додатків.

Єдина проблема полягає в тому, що *MCD*-порт від *Microsoft* не дозволяє використовувати всі функції *OpenGL*, що штучно призводить до її відходження на другий план.

Розкриття всіх можливостей *OpenGL* доступне тільки в разі наявності *ICD*-порту, що повністю підтримує цю бібліотеку. Компанією *SGI* розроблено спеціальний *ICD*-порт, який застосовує для прискорення додатків для *Microsoft NT*.

Багато виробників акселераторів додають до комплекту драйверів свої версії *ICD*-портів, які підтримують максимально ефективне використання можливостей прискорення, наданих набором *3D*-мікросхем. Виробники можуть вільно додавати до *ICD*-портів нові функції, не передбачені в *OpenGL*.

Виробники чіпсетів для *3D*-акселерації систематично випускають версії оптимізованих *ICD*-портів, а тому з'явилося багато ігор, орієнтованих на роботу з *OpenGL*. Графічна бібліотека *OpenGL* на платформі *PC* набуває все більшої популярності серед розробників *3D*-акселераторів та виробників ігрового ПЗ.

Але існування великої кількості *3D*-акселераторів та модифікованих версій *ICD*-портів призвели до того, що: при всій легкості *OpenGL* у програмуванні та наданні нею широких можливостей акселерації та створення неперевершених *3D*-ефектів, на практиці важко примусити працювати з підтримкою *OpenGL* деякі версії ігор.

Ці недоліки є наслідком того, що: виробники *3D*-акселераторів не набули навичок у написанні та відлагодженні *ICD*-портів та драйверів до *3D*-акселераторів; *OpenGL* недостатньо активно використовують.

Для усунення цих недоліків: виробники відеоадаптерів та ігор розробляють програми, що виправляють помилки в іграх,

та оновлюють драйвери; до дистрибутивів 3D-ігор включаються підбірки драйверів та ICD-портів для різних моделей акселераторів; застосовуються програми-оптимізатори, що керують стартовими параметрами гри.

**Графічна бібліотека *Glide API*** є розповсюдженою. Її застосовують у великій кількості ігор. Переваги *Glide API* такі: вона дозволяє створювати реалістичні ефекти та забезпечує високу швидкодію; в ній гарно реалізована підтримка 3D-акселерації для DOS (цього не вистачає *Direct3D* та *OpenGL*); функціональні можливості *Glide API* подібні до *OpenGL*.

Особливістю *Glide API* є націленість на використання в ігрових додатках, незважаючи на відсутність таких широких можливостей міжплатформового перенесення додатків, як у *OpenGL*. Як і *OpenGL*, бібліотека *Glide API* має проблеми з настроюванням програмних додатків, які її використовують.

**Графічну бібліотеку *QuickDraw 3D* з пакету *QuickTime*** створено фірмою *Apple*. Ця фірма розробила стандарт *QuickTime* та підтримує його версії для *MacOS*, *Java*, *Windows/NT*. Пакет *QuickTime* містить набір кодеків, підтримує потокове відео для *Internet* та більше 150 відеоефектів, має функції віртуальної реальності. Фахівці порівнюють значення *QuickTime* для мультимедіа з *PostScript* для принтерів, тому він має підтримку виробників мультимедійного ПЗ.

До складу пакету *QuickTime* входить *QuickDraw 3D*, який дозволяє маніпулювати 3D-графікою: вбудовувати, переглядати, керувати в режимі реального часу. *QuickDraw 3D* – єдина можливість стандартизації солідних *Mac*-додатків. Лише деякі з *PC*-додатків підтримують формат *3DMF* цієї архітектури та систему апаратного прискорення *RAVE*.

## Питання та завдання для самоперевірки

1. Окресліть роль стандартних *API* для роботи з графікою.
2. Назвіть найвідоміші *API* для роботи з графікою.
3. Вкажіть галузі застосування графічних *API*.
4. Вивчіть та проаналізуйте особливості організації та функціонування ПЗ графічних підсистем *АСУ*, *АСУТП* та *САПР* на прикладі декількох підприємств та установ вашого міста (району, області).
5. Дослідіть декілька найкращих вітчизняних та зарубіжних зразків упорядкування ПЗ графічних підсистем великих комп'ютеризованих виробництв (комплексів) за матеріалами фундаментальних посібників, періодичної преси, науково-практичних конференцій, *Web*-сторінок мережі *Internet*.

## ГЛАВА 5. ГРАФІЧНЕ АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ

Розширення архітектури комп'ютерних систем новими пристроями відкриває додаткові галузі їх застосування, де комп'ютер використовують не тільки як обчислювальний засіб. Наприклад, ПК з потужною відеокартою дозволяє: отримувати моделі складних геометричних фігур та сцен, які зображують будівлі, споруди, механізми, ділянки земної поверхні, живих істот та природні явища; відрисовувати високоякісні графічні зображення цих моделей з будь-якої точки зору; анімувати складні процеси в динаміці. У цьому зв'язку важливим є розгляд складу та функцій апаратного забезпечення графічної підсистеми ЕОМ.

У формуванні зображення (на екрані комп'ютеру, в файлі, в вигляді голограми, на традиційних носіях інформації тощо) приймає участь більшість пристроїв комп'ютеру.

*Типова підсистема формування зображення за допомогою ЕОМ містить* [28, 47]: пристрої введення інформації, що описує зображення, до ЕОМ; пристрої передавання інформації, що описує зображення, каналами міжкомп'ютерного зв'язку (локальними та глобальними мережами тощо); центральний процесор комп'ютеру (ЦПК), який виконує в процесі формування зображення роль координатора (ті дії, що неможливо або недоцільно виконувати засобами ЦПК, він спрямовує до процесорів специфічних пристроїв); математичний співпроцесор комп'ютеру, що виконує складні математичні обчислення з високою швидкістю та точністю результату; оперативний

запам'ятовувальний пристрій комп'ютеру, необхідний для зберігання даних та коду графічної програми; запам'ятовувальні пристрої довготривалого зберігання графічної інформації; пристрій формування відеосигналу (відеокарта), що перетворює отримані дані до такого сигналу, який сприймає пристрій відображення; пристрої виведення графічної інформації (зокрема, відображення відеосигналу); шини даних комп'ютеру, що забезпечують зв'язок між вказаними пристроями.

Деякі з вказаних пристроїв мають власні підсистеми, що складаються з процесорів, чипів пам'яті тощо. Так, відеокарта може містити: власний процесор; певний обсяг пам'яті; систему формування відеосигналу та систему, що ним керує; процесори графічних прискорювачів тощо.

До наведеного переліку основних пристроїв можуть бути додані інші класи пристроїв (наприклад, датчики для збирання інформації, що характеризує графічне зображення).

## 5.1. Пристрої введення графічної інформації

*Основними класами пристроїв введення графічної інформації є: дигітайзери (графічні планшети); пристрої сканувального введення; маніпулятори; пристрої клавіатурного введення; пристрої тактильного введення; пристрої мовного введення [47].*

**Дигітайзери (графічні планшети).** Ці пристрої є найприроднішим засобом графічного введення інформації та найефективнішим знаряддям працівників сфер рекламного та графічного дизайну, CAD- та 3D-додатків. Назва "дигітайзер"

походить від англійського слова *digit* (цифра), тобто перекладається як оцифровувач або аналогово-цифровий перетворювач. Оскільки найчастіше дигітайзери виконують у вигляді планшету, їх називають графічними планшетами.

Графічні планшети застосовують для поточкового координатного введення графічних зображень у САПР, комп'ютерній графіці та анімації. Графічний планшет надає не найшвидший та не найзручніший спосіб побудови креслень та рисунків складної геометрії, але забезпечує найточніше введення графічної інформації до комп'ютеру.

Типовий графічний планшет забезпечує дозвіл не менше 100 *dpi*, підключається до комп'ютеру кабелем та містить: робочу площину, на яку може бути нанесена (для полегшення введення до комп'ютеру складних зображень) допоміжна координатна сітка; кнопки управління, розташовані поряд із робочою площиною; пір'їну або координатний пристрій з прицілом (його підключають кабелем до планшету) для введення інформації.

У найдосконаліших та найдорожчих *п'єзоелектричних дигітайзерах* робоча поверхня планшету має тактильну чутливість на базі п'єзоелектричного ефекту. Завдяки цьому введення графічної інформації з дозволом 400 *dpi* здійснюють без спеціальних пір'їн та прицілів.

П'єзоелектричні дигітайзери дозволяють креслити так само, як на звичайній креслярській дошці.

За принципом п'єзоелектричного дигітайзера побудовано тактильні пристрої для роботи з графічним інтерфейсом користувача в ОС *Windows* та *OS/2*. Зручніше та легше водити пальцем уздовж віконця дигітайзера, що має розмір не більше

коробочки сірників, ніж застосовувати мишу або трекбол: рухи пальця вздовж планшету чутливо та слухняно повторює курсор на екрані комп'ютеру; додаткові кнопки відсутні (зробивши на екрані вибір, двічі стукають пальцем по віконцю дигітайзеру).

Для введення графічної інформації можна використовувати деякі планшетні графопобудовники. Але більшість зображень (карти, рисунки, креслення, графіки, фотографії, слайди, кінофільми) зручніше вводити відеодигітайзером. Є багато спеціальних графічних систем з різними типами відеодигітайзерів, що вводять до ПЕОМ кольорові зображення з паперового носія або зі слайдів. Наприклад, відеодигітайзерами є цифрові фотокамери та відеокамери. У кіностудіях, для перенесення зображення з плівки в комп'ютер, також використовують спеціальні дигітайзери (після цифрової обробки зображення знову переносять на плівку).

*Завдяки дигітайзерам стало можливим широкомасштабне практичне впровадження технологій віртуальної реальності*, створеної в пам'яті комп'ютеру.

Так, за допомогою дигітайзеру в комп'ютери кіностудій вводять костюми, інтер'єри, нарисовані декорації, фотографії пейзажів. Можна вводити до комп'ютеру фотографії акторів, формувати шляхом їх комбінування нові персонажі, реалістично анімувати їх тощо.

Найбільше розповсюдження, широку підтримку виробників ПЗ та визнання фактичним індустріальним стандартом отримали графічні планшети фірми *WACOM*.

Розглянемо переваги графічних планшетів: п'яра планшету надає гарні можливості контролю за отримуваним зображенням; високий дозвіл планшетів (на порядок більший,



ніж у миші) забезпечує високу точність, плавні рухи та зручність роботи в разі великого збільшення зображення; висока чутливість графічних планшетів до сили натискання та кута нахилу пір'їни підтримана численним ПЗ для дизайнерів та комп'ютерних художників на різних платформах; завдяки абсолютній адресації, графічний планшет працює як звичайна указівка (це швидше, зручніше та менше втомлює, ніж катати мишу вздовж килимка); здатність планшету відчувати пір'їну на певній відстані від поверхні дозволяє пересувати курсор та натискати на кнопки, не торкаючись планшету, що цінується в процесі верстання; за допомогою пір'їни можна швидко та інтуїтивно видаляти зайвий текст; протилежний кінець пір'їни може працювати як будь-який інструмент для малювання; графічний планшет зберігає здоров'я та сили працівника, сприяє зручній роботі.

Після використання графічного планшету, застосування миші створює враження жодної зручності та інтуїтивності, напруження в руці та зап'ястку, складності здійснення точних рухів.

Швидко звикаючи до планшету, користувачі намагаються повністю замінити ним мишу. Це не рекомендується робити з економічних причин (кінчик пера зтирається).

**Пристрої сканувального введення.** *Сканувальні пристрої (сканери, цифрові відеокамери та фотоапарати)* здійснюють пасивне оптичне введення інформації та перетворення її в цифрову форму з наступним обробленням. Сканер дозволяє оптично вводити до ЕОМ чорно-білу або кольорову, друковану чи рукописну графічну інформацію з паперу або іншого носія.

Сучасні сканери використовують у складних інтелектуальних системах *OCR (Optical Character Recognition* – оптичного розпізнавання символів), які дозволяють швидко вводити до комп'ютеру та надалі читати текст (наприклад, рукописний), не користуючись клавіатурою.

***Залежно від конструкції та принципу дії, сканери поділяють на такі класи: ручні, планшетні, барабанні, проєкційні.***

Ручні сканери є найпростішими та найдешевшими, мають невеликі габаритні розміри. Але досягти високої якості зображення з їх допомогою важко. Не мають вони й інтелектуальності. Тому ручні сканери використовують для обмеженого кола задач.

Планшетні сканери є розповсюдженим типом сканерів. Початково вони були призначені для сканування непрозорих оригіналів. Завдяки зйомній кришці, можна сканувати товсті оригінали. Деякі моделі мають механізм подавання окремих аркушів, що зручно в роботі з *OCR*-програмами. Багато фірм пропонує слайд-модуль для сканування прозорих оригіналів.

***Барабанні сканери, конструкція яких забезпечує найвищу якість сканування, можуть: обробляти будь-які типи оригіналів; сканувати непрозорі та прозорі оригінали одночасно.***

Проєкційні сканери застосовують для сканування слайдів невеликого формату (до 4"×5") з високими дозволом та якістю. Вони мають дві модифікації: з вертикальним та горизонтальним розташуванням оптичної осі зчитування. Оригінали бувають двох типів: прозорі негативні та позитивні слайди, що сканують у світлі, яке проходить крізь слайд; непрозорі оригінали, якими є аналогові зображення (фотографії) або дискретні зображення (ілюстрації з друкованих видань).

Механізми зчитування сканерами зображення базуються на фотопримножувачах або на ПЗЗ-елементах. ПЗЗ-сканери є швидкодійнішими за сканери на фотопримножувачах.

Сканери розрізняють за багатьма параметрами: технологією зчитування зображення, типом механізму та іншими. Тими параметрами, що впливають на якість зображення, є: оптичний дозвіл; кількість напівтонів та кольорів, які можуть бути переданими; діапазон оптичних щільностей; інтелектуальність; світлові викривлення; точність фокусування (різкість).

Сучасний сканер функціонально складається з двох частин: сканувальний механізм (*engine*); програмна частина (*TWAIN*-модуль, система керування кольором тощо) [176].

**Обґрунтуємо доцільність введення до комп'ютеру кольорових та чорно-білих фотознімків:** фоторепортерам часто потрібні миттєві фотознімки, щоб одразу впевнитися в їх якості та виразності; цифрові фотознімки можна одразу використувати для електронного верстання в журналістській практиці (в газеті, на телебаченні, в інформаційному агенстві тощо); файл із оцифрованим зображенням можна одразу переправити каналами зв'язку на будь-яку відстань; цифрові комп'ютерні зображення зручно редагувати, ретушувати, споряджати спец-ефектами, кадрувати; процес отримання цих зображень є екологічним у сенсі відмови від застосування хімічних процесів; тривале зберігання цих фотознімків на компакт-дисках є зручнішим та надійнішим, ніж традиційний спосіб; за допомогою комп'ютеру зручно демонструвати фотознімки великій аудиторії (наприклад, з навчальними цілями); цифрові фотознімки необхідні для створення мультимедіа.

*Існує альтернатива в способі введення зображення з оригіналу або з фотознімку до пам'яті комп'ютеру:* цифрову камеру призначено для введення зображень безпосередньо з оригіналу до комп'ютеру; для введення кольорового зображення з фотознімку до пам'яті комп'ютеру потрібен кольоровий сканер або дигітайзер для введення слайдів; живі кадри можна ввести безпосередньо з відеокамери або відеомагнітофону. Цифрові фотокамери забезпечують якісніше введення, ніж відеокамери, є найпростішим та найшвидшим способом введення зображення до комп'ютеру. Цифрові камери записують зображення до пам'яті, інформація з якої, без застосування додаткових пристроїв, може бути введена до комп'ютеру крізь порт зв'язку. Для збереження знімків, фірма *Kodak* розробила практичну та недорогу технологію розташовування електронних фотографій на компакт-диску в стандарті *Kodak Photo CD* (на одному *CD* розміщують фотоальбом). За допомогою плеєру ці диски можна переглядати на телевізорі та комп'ютері.

**Маніпулятори.** *Розновсюдженими класами цих пристроїв є миша, джойстик, трекбол та геймпад.*

Маніпулятор "миша" (*mouse*) є найпопулярнішим пристроєм введення графічної інформації до ПЕОМ та базовим засобом вказування в екранному багатовіконному діалозі. Сигнал, який виробляє миша в процесі пересування вздовж поверхні, перетворюється в сигнал керування розташуванням курсору: можна пересувати курсор з точністю до одного елементу розкладання, що вигідно відрізняється від світлової пір'їни. Для відстежування пересування миші, використовують електромеханічні, тензометричні та оптичні датчики. Миша з електромеханічним датчиком потребує скороченого обсягу

технічного обслуговування, з тензодатчиком – працює на будь-якій поверхні та не вимагає спеціального чутливого планшету, з оптичним датчиком – забезпечує ще вищу чутливість, але вимагає поверхні зі спеціальним рисунком у вигляді сітки.

*Загальними недоліками маніпуляторів "миша" є: певна невідповідність у просторі пересування курсору на екрані та маніпулятору на поверхні столу, що ускладнює координацію рухів та вимагає тренування.*

Маніпулятори "миша", орієнтовані на роботу в *Web*, окрім однієї або двох основних, мають додаткові кнопки або коліщатка, що застосовуються для вертикального скролінгу сторінки в вікні броузера.

Трекбол нагадує перевернуту догори колесиком мишу: рухається не корпус, а розташована зверху куля, що підвищує точність управління курсором. Миші та трекболи є координатними пристроями введення інформації до комп'ютеру, а тому не можуть повністю замінити клавіатуру.

Джойстики є аналоговими координатними пристроями введення інформації. Фірмами-лідерами в галузі їх виробництва є *Gravis* та *TrushMaster*.

Компанія *Gravis* має світову відомість завдяки високоточним маніпуляторам джойстикам та геймпадам.

Фірма *TrushMaster* є однією з найкращих у виробництві ігрових контролерів. Її провідною ідеєю є адаптація до ПК обладнання професійних авіатренажерів: зручніше та цікавіше керувати літальним апаратом за допомогою контролеру, подібного до справжнього, на якому є всі необхідні функції, ніж за допомогою клавіатури комп'ютеру. Згідно з цим компанія виконує дизайн найдосконаліших джойстиків та ігрових компонентів.

**Пристрої клавіатурного введення.** Найбільш універсальним пристроєм введення графічної інформації є клавіатура, що виконує дві глобальні функції: введення інформації та управління нею. Перевагою клавіатурного введення, окрім багатофункціональності, є відсутність необхідності спеціального підготування користувачів.

**Основними типами клавіатур є:** клавіатури, складені з окремих клавіш (їх перевагою є незалежні від інших клавіш натиснутий та відпущений стани); клавіатури з інтегральним виконанням контактного поля (їх перевагою є невисока вартість), які поділяють на ємносні, індуктивні та оптичні (з відсутністю деренчання та високою надійністю).

Перспективним напрямком розвитку клавіатурного введення є безпроводовий інтерфейс із системним блоком за допомогою інфрачервоного (ІЧ) випромінювання.

**Пристрої тактильного введення.** Основним представником цього класу пристроїв є **сенсорний екран (touch screen)** – пристрій безпосереднього введення інформації, за допомогою якого можна керувати тією інформацією, що подана користувачу на екран дисплею (див. рис.5.1).

У практичному застосуванні сенсорний екран є: ефективним засобом піктограмного діалогу некваліфікованого користувача з ПЕОМ (особливо в системах з інтелектуальним графічним інтерфейсом); пристроєм, який несе на собі основне навантаження діалогу користувача з ПЕОМ в інтегрованих системах організаційного управління, що поєднують засоби мовного введення та багатовіконний графічний інтрфейс (особливо в системах, орієнтованих на керівників). У разі введення великих обсягів даних, сенсорний екран не замінить клавіатури.



Рис.5.1. Сенсорний монітор фірми *LG*: достатньо легкого натискання на екран, і будь-яка опція меню та значення перемикача будуть обрані та приведені в дію

Основними типами сенсорних екранів є: резистивні; з ємносними датчиками; з акустичними датчиками; з оптичними датчиками вздовж периметру екрану.

Є сенсорні екрани, що можна кваліфікувати як інтелектуальні наступники маніпулятора "миша", наприклад: пристрій *Cruise Cat* компанії *Cirque*, керований паличкою (пласка тонка сенсорна панель, яку підключають до послідовного порту ПК та постачають із ПЗ, що перетворює ступінь легкості натискання, кількість натискань паличкою на панель та написання нею користувачем певних символів, в команди типу клацання лівою або правою кнопкою миші, виклику *Internet*-закладок тощо); пристрій американської компанії *Varatouch* – невелика кругла пластина, на вершині якої

розташовано рухливу кульку (чутливий сенсор відстежує пересування кульки, а для охоплення всієї області пересування курсору на екрані достатньо обернути кульку на 15°).

**Пристрої мовного введення (ПМВ).** Цей вид введення інформації є перспективним, оскільки підвищує природність спілкування людини з ЕОМ та поєднує в єдину інформаційну систему мову та дані.

У багатьох ПЕОМ серійного випуску використовують модуль ПМВ з різними пакетами прикладних програм. Наприклад, модуль ПМВ *Voca Link (CSRB 240)* американської фірми *Interstate Voice Products* дозволяє програмувати 240 окремо вимовлюваних команд керування стандартними програмами ПЕОМ *BM PC* для бухгалтерських розрахунків та текстової обробки. Він працює в двох режимах: клавішної імітації (клавіатурне введення замінене мовними командами); ПЕОМ (складаються програми для введення спеціальних команд).

**Основними параметрами оцінки ПМВ є:** можливість розпізнавати природну злитну (неперервну) мову або ізольовані слова (команди), що розділені штучними паузами певної довжини; ступінь залежності від диктору; обсяг словника.

### Питання для самоперевірки

1. Класифікуйте пристрої введення графічної інформації.
2. Охарактеризуйте пристрої сканувального введення.
3. Чому дигітайзери є найприроднішим засобом графічного введення?



## 5.2. Засоби виведення зображень

*Базовими класами пристроїв виведення графічної інформації є:* плотери (графопобудовники); принтери; дисплеї [47].

**Плотери (графопобудовники, *plotters*).** Це пристрої, що виконують функції виведення графічної інформації на паперовий та деякі інші носії (рис.5.2). Основними класами плотерів є: пір'їнні; олівцево-пір'їнні; струминні; електростатичні; прямого виведення зображення (*DIP*); на основі термопередавання (*TTP*); лазерні; світлодіодні (*LED*);

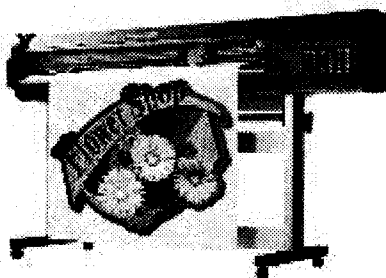


Рис.5.2. Плазмовий плотер (є одним із найсучасніших)

**Пір'їнні плотери (*pen plotter*)** – електромеханічні пристрої векторного типу: на них традиційно виводять зображення векторні програмні пакети (*AutoCAD*, *CorelDraw* та інші). Вони створюють зображення за допомогою писальних елементів – пір'їн. Є декілька видів пір'їн, які відрізняються використанням типом рідкого барвника.

Пір'їни розділяють на: одноразові; багаторазові (з можливістю перезарядження). Пір'їну закріплено в тримачі писального вузла, що має один або два ступеня свободи пересування.

Пірітніні плотери розділяють на планшетні та барабанні (рулонні). Точність виведення інформації барабанними плотерами нижча в порівнянні з планшетними, але відповідає вимогам більшості застосувань. Ці плотери компактніші та можуть відрізати від рулону аркуш необхідного розміру автоматично, що визначило їх домінування на ринку великих пірітніних плотерів. Пірітніні плотери формату А3 як правило є планшетними.

Перевагою пірітніних плотерів є висока якість зображення та гарне передавання кольорів. Швидкість виведення інформації пірітніними плотерами є невисокою, незважаючи на збільшення швидкості механічних елементів та оптимізацію процедури рисування. Також доводиться вирішувати проблему підбирання пари "носій – чорнила".

*Олівцево-пірітніні плотери (pen/pencil plotter)* – різновид пірітніних плотерів, який дозволяє встановлювати писальний вузол з цанговим механізмом для використання олівцевих грифелів. Ця технологія надає такі переваги: не потрібно слідкувати за виведенням інформації; олівець пише на будь-якій швидкості та на будь-яких паперових носіях, навіть не високоякісних (при цьому зображення є якісними, дають гарні відбитки при копіюванні та піддаються корегуванню ластиком); грифелі легко придбати, значно економлячи на витратних матеріалах.

Пірітніні та олівцево-пірітніні плотери привабливі для тих, кому важливішою є якість, а не кількість зображень, та хто має скромний бюджет.

Інші плотери створюють зображення, використовуючи фізичні процеси, зокрема – вдаючись до дискретного (растрового) способу створення.

**Струминні плотери (*inkjet plotter*)** базуються на технології створення реактивної бульбашки (*bubblejet*) – спрямованого розпилення чорнил на папір за допомогою сотен дрібних форсунок одноразової друкувальної голівки.

Друкувальні голівки можуть бути "кольоровими", маючи відповідну кількість груп форсунок. Для створення повноцінного зображення, застосовують стандартну для поліграфії кольорову схему *СМУК*, що використовує чотири кольори: блакитний (*Cyan*), пурпурний (*Magenta*), жовтий (*Yellow*), чорний (*black*).

Позитивними якостями струминної технології є: простота реалізації, високий дозвіл, низька питома потужність, непогана швидкість друку. Прийнятна ціна, висока якість та великі можливості роблять струминні плотери конкурентом пір'їнних. Зростає попит на струминні плотери користувачів настільних видавничих систем та САПР, які випускають складні креслення формату А0. Обмежують застосування струминних плотерів невисока швидкість виведення та вицвітання кольорового зображення (якщо не використовувати ламінування або самоламініувальний папір).

**Електростатичні плотери (*electrostatic plotter*)** використовують електростатичну технологію створення прихованого електричного зображення (потенційного рельєфу) на поверхні електростатичного паперу.

Електростатичні плотери вимагають стабільних температури та вологості в приміщенні, ретельного обслуговування та

дорого коштують. Важливими є: висока стійкість зображення до впливу ультрафіолетових променів; висока вартість електростатичного паперу, на рівні високоякісного друкарського. Для досягнення максимальної ефективності, ці плотери працюють як мережні пристрої, споряджені адаптерами мережного інтерфейсу.

Електростатичні плотери застосовують у разі високого ступеня автоматизації проектних робіт у солідних організаціях та в геоінформаційних системах з виправдано високими вимогами до продуктивності та якості.

**Плотери прямого виведення зображення (*direct imaging plotter*)** створюють зображення на термопапері, просякненому теплочутливою речовиною. Зображення виходить високоякісним (дозвіл до 800 dpi), але монохромним. Відзначимо, що: ціни на термопапір знизилися; ряд його недоліків (чутливість до змінювання температури довкілля, низька контрастність зображення) ліквідовані; перелік типів термоносіїв містить стандартний білий папір, кальку та поліефірну плівку, а якість носіїв відповідає найсуворішим архівним вимогам.

Враховуючи високу надійність, продуктивність (50 аркушів формату A0 за день) та низькі експлуатаційні витрати цього класу плотерів, їх застосовують у великих проектних організаціях для виведення перевірочних копій, а тому до їх стандартної конфігурації входить мережний адаптер.

Технічні характеристики плотерів прямого виведення зображення відповідають вимогам прикладних задач інженерного проектування, архітектури, будівництва, міського планування та електросхемотехніки.

**Плотери на основі термомпередавання (*thermal transfer plotter*)**. Відмінність цього класу плотерів від попереднього

полягає в тому, що між термонагрівачами та папером або прозорою плівкою розташовано донорний кольороносії. Зображення отримують за чотири проходи, витрачаючи на кожен аркуш в чотири рази більше фарбувальної стрічки, ніж для монохромного зображення. Через високу вартість відбитку, плотери на основі термопередавання використовують: у складі засобів автоматизованого проектування, для високоякісного виведення об'єктів 3D-моделювання; у системах картографії, де потрібна висока якість відтворення кольорів; рекламними агентствами для виведення кольоропроб плакатів та транспарантів для барвистих презентацій.

**Лазерні плотери (laser plotter)** базуються на електрографічній технології. Приховане зображення на барабані створюється лазером. Для управління лазерним променем застосовується складна система оптичних елементів, які обертаються – дзеркальних багатогранників (призм, лінз). Тому лазерні плотери бояться трясіння та ударів, які збивають їх налаштування.

**Світлодіодні плотери (led plotter)** дозволяють уникнути складностей з оптикою, властивих лазерним плотерам. Вони роблять систему простішою, легшою та надійнішою завдяки застосуванню лінійок точкових напівпровідникових світлодіодів (*light-emitting diode – LED*). LED-плотери стають усе популярнішими, хоча за вартістю еквівалентні монохромним електростатичним. Галузями їх застосування є: складний технічний дизайн, архітектура, картографія та інші, де вимоги до продуктивності та якості результатів високі, а наявність кольору не вимагається.

Лазерні та LED-плотери, завдяки високій швидкодії (аркуш формату A1 – максимум за півхвилини), зручно викорис-

товувати як мережні пристрої (у їх стандартній комплектації є адаптер мережного інтерфейсу). Вони можуть працювати на звичайному папері, що скорочує експлуатаційні витрати. Кольорові лазерні плотери коштують дорого.

Стандартні плотери є доступними широким масам споживачів: їх характеризують невисока ціна та гарна якість друку.

Основними параметрами плотерів є: параметри носія зображення, параметри точності, дозвіл друку, повторюваність, погрішність зупинки пір'їни, параметри продуктивності, параметри пам'яті, формати даних, креслярські характеристики.

Тенденції ринку плотерів такі: ринок широкоформатних плотерів зростає на фоні гострого суперництва використаних технологій; стійкі позиції утримують олівцево-пір'їнні плотери (завдяки дешевим витратним матеріалам); значно покращилися технічні характеристики кольорових струминних плотерів (маючи меншу вартість, за якістю зображення вони наближаються до кольорових електростатичних плотерів); розширюється коло виробників струминних плотерів, що підвищує гостроту конкуренції та знижує ціни на ці вироби, приводячи до ширшого використання кольору в підготуванні документів; щоб протистояти наступу струминних кольорових плотерів, виробники інших типів пристроїв вдосконалюють їх характеристики; висока ціна LED-плотерів робить їх мало доступними широким масам користувачів, але ці пристрої знаходяться в одній "ваговій категорії" з монохромними електростатичними та починають їх витісняти.

**Принтери.** *Основними класами принтерів є:* матричні, струминні, лазерні, LED-принтери, термопринтери (з термопеченням, з термосублімацією), зі змінюванням фази барвнику.

Найбільш розповсюдженими принтерами є матричні, струминні та лазерні, що відрізняються технологією друку. Якісні та кількісні характеристики принтерів різних класів суттєво відрізняються. Основними характеристиками друку є: швидкість друку, якість друкованої продукції, її дозвіл, вартість тощо.

**Матрична технологія друку** має перевагу в малій ціні відбитку. Всі інші параметри (якість зображення, швидкість друку, шумність, надійність) належать до недоліків, через які застосування цієї технології для графічного та геометричного моделювання в САПР недоцільне.

**Струминна технологія друку** для принтерів – така ж, як і для плотерів.

**У лазерних (Laser) принтерах** використовують електрографічний принцип створення зображення. Найважливішими складовими частинами лазерного принтеру є: фотопровідний циліндр (друкувальний барабан), напівпровідниковий лазер та прецизійна оптико-механічна система, що пересуває світловий промінь.

**У LED-принтерах (Light Emitting Diode)** напівпровідниковий лазер замінено гребінкою світлодіодів, що робить непотрібною складну рухоми оптичну систему дзеркал та лінз. Зображення одного рядка на світлочутливому барабані формується одночасно.

**Принтери з термонесенням (Thermal Wax Transfer)** працюють із застосуваннями термопластичної фарбувальної речовини. Конструктивно цей спосіб друку простий та забезпечує безшумну роботу. Оскільки між друкувальною голівкою та папером механічний контакт відсутній, термопринтери належать до безударних пристроїв.

*Принтери з термосублімацією* використовуються для кольорового друку (комбінацією кольорів барвників можна підібрати будь-яку кольорову палітру) та належать до класу "high-end".

*Принтери зі змінюванням фази барвника (phase change inc-jet)* також називають принтерами з твердим барвником (*solid inc-jet*): на відміну від струминної технології, не буває просякання, розтікання та змішування барвників, а тому можлива робота з будь-яким папером.

**Дисплеї.** Ці пристрої розділяють на такі базові класи: електронно-променеві (пласкі, звичайні); рідинно-кристалічні; плазмові.

Найширше розповсюджені дисплеї на базі електронно-променевих трубок (ЕПТ-монітори), що пояснюється дешевизною технології виготовлення, якістю зображення, простотою в експлуатації та невисокою ціною.

В ЕПТ-моніторі електроніка розташована окремо, що позитивно впливає на експлуатаційні характеристики, робить можливими ремонт та заміну несправних компонентів. Якість зображення ЕПТ-монітору є найвищою, відсутня інерційність зображення.

У виробництві рідинно-кристалічних (РКД) та плазмових дисплеїв, брак сягає 40 % через складну технологію виготовлення. Особливо це стосується електронної частини, вмонтованої до РКД-дисплею, оскільки неможливі ремонт та заміна компонентів, які вийшли з ладу.

У РКД та плазмових дисплеях спостерігається інерційність зображення (в процесі руху об'єкту повільно перерисовується).



ується весь екран, створюючи шлейф розмитості зображення), що обумовлене принципами їх функціонування.

Кут огляду, контрастність, яскравість у кожного виду пристроїв є індивідуальними. При цьому жоден з них не має всіх цих показників із високим ступенем якості.

ЕПТ-монітори є найпрактичнішими. Ціна – це головний критерій для покупця, але багато залежить від того, для яких цілей будуть використовувати дисплей: для мобільних користувачів підходить РКД-дисплей (він є плоским, компактним, зручним, легким та економічним); якщо потрібно проводити конференції та збори, потрібен плазмовий дисплей або РКД-дисплей (їх легко виговити великими за розміром при гарній якості зображення на відстані); для офісу найбільш підходять ЕПТ-монітори.

Пласкі ЕПТ-монітори є прогресивним напрямком. У їх основу покладено технологію *DynaFlat*, яка гарантує високий дозвіл, яскравість та концентрацію зображення.

## Питання для самоперевірки

1. Надайте класифікацію пристроїв виведення графічної інформації.
2. Виконайте порівняльний аналіз основних класів дисплеїв.
3. Сформулюйте тенденції сфери практичного застосування плотерів та принтерів.

### 5.3. Пристрої для обробки графіки

До цих засобів належать ЦПК, математичний співпроцесор та певні складові елементи відеокарти [47, 104].

**Розподіл функцій з обробки графічної інформації між центральним процесором та відеокартою.** ЦПК призначено для обробки будь-яких видів інформації. Ця універсальність дозволяє програмно реалізовувати будь-які обчислення, що здійснюються на спеціалізованих пристроях. На долю інших пристроїв припадають введення, виведення, передавання та зберігання інформації.

Початково функцією відеокарти було відображення на моніторі зображення, складеного ЦПК: на основі необхідних даних ЦПК створював цифрове подання 2D-зображення в системній пам'яті комп'ютеру, копіював його (відповідно формату поточного графічного режиму екрану) до активної або неактивної сторінки пам'яті відеокарти та надсилав команду перемикавання сторінки; відеокарта сканувала активну сторінку та надсилала до монітору послідовність сигналів для правильного її відображення.

Цей підхід примушував ЦПК здійснювати обчислення, які спеціалізований процесор виконує ефективніше. Тому з'явилися відеокарти, процесор яких може взяти на себе частину обчислень з обробки графіки. При цьому ЦПК може паралельно здійснювати інші обчислення та псевдопаралельно виконувати іншу програму без відображення на моніторі (процесор по черзі перемикається між програмами), а відеокарта може неперервно продовжувати обчислення та

виведення зображення. Першими можливостями, які апаратно підтримували відеокарти, було відображення на площині простих фігур. Поступово в відеокартах реалізовували підтримку більших можливостей обробки зображень.

3D-акселератори дозволили вивільнити ЦПК від обробки великих обсягів даних, пов'язаних із обробкою текстур поверхонь (обертанням, розтягуванням, стисканням, відображенням динамічного освітлення, розмиттям тощо). Оскільки текстури поверхонь займали великі обсяги системної пам'яті та вимагали постійного прокачування готового зображення до відеопам'яті, завантажуючи шину, 3D-акселератори дозволили розмістити текстури у внутрішній пам'яті карти та здійснювати їх обробку без участі ЦПК.

При цьому ЦПК оброблює необхідні зміни сцени, яку моделюють (переміщення, змінювання форми та розмірів об'єктів, пересування джерел світла тощо), а також фактори, що обумовлюють ці зміни (фізику об'єктів в іграх та системах моделювання реальності, команди користувачів зі змінювання зображень у системах САПР тощо). Моделювання об'єктів деталізується до обробки елементарних частин – трикутників. Визначається, які трикутники є видимими на екрані, а їх координати передаються акселератору. При цьому вказується, яку текстуру необхідно поставити у відповідність конкретному трикутнику та які ефекти потрібно здійснити над нею.

У останній час на відеоакселератор покладають ще більше задач: часткова обробка геометрії (відеокарті подається детальний опис об'єктів та відповідних текстур, а далі процесор оперує готовими об'єктами, передаючи відеокарті зміни в їх розташуванні); моделювання фізики об'єктів.

Окрім паралельної обробки, перевага від перекладання частини задач з обробки зображення на відеокарту полягає в тому, що відеокарта, завдяки вузькій спеціалізації, може виконати ці дії швидше та ефективніше. ЦПК є універсальним, тому стандартний набір його команд містить найпростіші операції над числами. Для обробки графічної інформації потрібна більша кількість цих команд, а їх виконання займає багато часу. Відеокарті для виконання тих самих дій потрібна набагато менша кількість команд, виконання яких оптимізоване. У сучасних процесорах є *SIMD*-команди (*Single Instruction Multiple Data*), що дозволяють виконувати складніші операції, наприклад, над матрицями. Першим таким набором для ПК став *MMX* (*MultiMedia eXtention*), який підтримують усі сучасні процесори. Потім фірма *AMD* розробила набір інструкцій для 3D-графіки *3DNow!*, а фірма *Intel* – його альтернативу *SSE* (він же *KNI – Klamath New Instructions*). Тобто оптимальний поділ задач між ЦПК та відеокартою є непростою проблемою.

Програмісти не повинні орієнтуватися на те, що на комп'ютері буде встановлено певний 3D-акселератор або що процесор буде підтримувати конкретний набір інструкцій. Тому для розробки програм використовують операції конкретного *API*. Драйвери цього інтерфейсу, залежно від характеристик наявного АЗ, оптимально розподіляють, яку частину операцій слід виконувати на відеокарті, яку – з використанням розширеного набору команд ЦПК, а яку – шляхом емуляції стандартним набором команд ЦПК.

**Розподіл функцій з обробки графічної інформації між складовими частинами відеокарт.** Будь-яка відеокарта

містить визначений набір складових частин. Розглянемо основні з них. **Графічний чип** є базовим компонентом відеокарти: в ньому здійснюються основні обчислення та реалізована підтримка основних 3D-ефектів. **AGP-інтерфейс (вихід)** – це графічна шина, що надає можливість зберігати текстури в ОЗП комп'ютеру та швидко їх звідтіля вилучати. **Відеопам'ять, встановлену на 3D-акселераторі**, використовують для створення кадрових буферів, зберігання текстур та інших даних, які повинні бути наготові для використання графічним процесором. **Відеовихід (VGA-конектор)** – невід'ємна складова частина відеокарти, призначена для виведення графічної інформації на екран. **Голчастий радіатор та вентилятор** підтримують нормальний температурний режим функціонування відеокарти.

Дорогі відеоакселератори, що завдяки багатофункціональності називаються графічними комбайнами, містять додаткові компоненти: роз'єми *Composite* та *S-Video* для підключення відеоапаратури; *TV*-тюнер для приймання телевізійних сигналів стандартів *PAL*, *SECAM*, *NTSC* тощо.

## Питання для самоперевірки

1. Поясніть, як здійснюється поділ функцій з обробки графічної інформації між ЦПК та відеокартою.
2. Наведіть перелік основних складових частин 3D-карти.
3. Яку роль відіграють *SIMD*-команди?

## 5.4. Засоби зберігання зображень

**Основні концепції.** У процесі роботи з графічними зображеннями задіяні всі типи внутрішньої пам'яті: як спеціалізованої графічної, так і звичайної [47, 88, 91, 104]. Розглянемо розподіл функцій та навантаження між ними.

Відзначимо, що продуктивність графічної підсистеми ЕОМ визначає швидкодія ЦПК, інтерфейсної шини, відеопам'яті та графічного контролера. Кількість інформації, що виходить з відеоадаптеру та передається до екрану, та швидкість, з якою інформація надходить до екрану, залежать від дозволу монітору, глибини кольору та частоти поновлення екрану.

Для забезпечення дозволу  $1024 \times 768$ , графічна підсистема вимагає 1 Mb пам'яті. Інформація про курсор та ярлики зберігається в буферній пам'яті дисплею (*off-screen memory*) для забезпечення швидкого доступу. Пропускна здатність пам'яті визначає співвідношення кількості Mb/s, які надходять до пам'яті та з неї. Типовий дозвіл  $1024 \times 768$ , у разі 8-бітової глибини кольору та частоти поновлення екрану 75 Hz, вимагає пропускну здатності 1118 Mb/s.

Функції обробки 3D-графіки вимагають збільшення доступної пам'яті відеоадаптеру. У відеоакселераторах для систем на базі Windows типовим є розмір встановленої пам'яті 4 Mb. Додаткову пам'ять, зверх необхідної для створення зображення на екрані, використовують для Z-буферу та зберігання текстур.

Графічний контролер – головний компонент мультимедіа-систем. Від нього вимагають найбільшої продуктивності – швидкості обміну даними між відеопам'яттю та графічним контролером. Оскільки графічний контролер здійснює обробку тих графічних функцій, які потребують інтенсивних обчислень, він розвантажує ЦПК, а тому повинен оперувати власною внутрішньою пам'яттю.

Тип пам'яті, в якій зберігаються графічні дані, називають буфером кадру (*frame buffer*). У системах обробки 3D-застосувань потрібна спеціальна пам'ять – *Z-buffer*, де зберігають інформацію про глибину зображеної сцени. Також може бути власна пам'ять текстур (*texture memory*) – для зберігання елементів, із яких формують поверхні об'єкту.

Розквіт мультимедіа та відеозастосувань, збільшення тактової частоти ЦПК не дозволяють використовувати стандартну динамічну пам'ять з випадковим доступом DRAM: мультимедіа-контролери вимагають від основної системної пам'яті більшої пропускну здатності та меншого часу доступу. Виробники покращили технології та створили нові архітектури пам'яті за допомогою традиційних та нових методів. Широкий вибір типів пам'яті ставить перед виробником відеоадаптерів проблему, для якого сегменту ринку та для яких застосувань обрати той чи інший тип.

Є два основних підходи до розташування відеопам'яті: назовні або всередині графічного контролеру.

Розташування відеопам'яті назовні графічного контролеру – традиційний підхід, коли відеопам'ять та графічний контролер використовують як незалежні компоненти. Інтерфейс графічного чіпсету забезпечує можливість використання різних

типів пам'яті: *VRAM*, *DRAM*, *3D RAM*, *CDRAM*, *SDRAM*, *WRAM* тощо. Архітектури, що підтримують пам'ять типів *DRAM* та *VRAM*, вільно доступні від різних виробників, що забезпечує постійну підтримку цих типів пам'яті та конкурентоспроможні ціни, незважаючи на те, що *VRAM* належить до класу "high-end". Чим швидшу відеопам'ять використовують, тим швидшою стає графічна підсистема.

Вбудована відеопам'ять (*embedded memory*) є частиною графічного контролера, що забезпечує зниження виробничих витрат, зменшення розмірів компонентів та збільшення швидкодії. Багато компаній активно виробляють ці компоненти, а виробники графічних чіпсетів – пристрої з вбудованими компонентами. Обробка графіки є одним із найважливіших застосувань, якому потрібна високопродуктивна пам'ять, але в менших обсягах, ніж для системної (оперативної) пам'яті комп'ютеру.

Оскільки сам по собі тип пам'яті *DRAM* не відповідає потребам графічних застосувань (особливо після того, як *3D*-графіка стала невід'ємною складовою частиною мультимедіа в ПК), було створено багатий вибір нових удосконалених типів пам'яті, доступних для використання виробникам графічних адаптерів, що є необхідним для досягнення того рівня продуктивності, якого вимагають сучасні *Windows*-акселератори – це *3D RAM*, *CDRAM*, *MDRAM*, *RDRAM*, *SGRAM/SDRAM*, *Embedded RAM*, *WRAM*, *FeRAM*, *DRDRAM*, *DDR SDRAM*, *ESDRAM*, *FCRAM*, *MRAM* [47, 104].

Стандартні архітектури пам'яті *DRAM*, *FPM DRAM*, *EDO DRAM*, *Fast EDO DRAM*, *Burst EDO*, *VRAM*.

*DRAM* (*Dynamic Random Access Memory*) є промисловим стандартом. На основі технології *DRAM* виробляють значну



частину розповсюджених типів пам'яті. Вдосконалення *DRAM* базуються на низькій вартості виробництва та суттєво збільшеній пропускній здатності. *DRAM* є енергозалежною пам'яттю. Покращення в архітектурі *DRAM* полягають в підвищенні продуктивності шляхом прискорення циклів читання та запису. *FPM DRAM* (*Fast Page Mode DRAM*), *EDO DRAM*, *Fast EDO DRAM* та *Burst EDO* є прикладами еволюції архітектури *DRAM*. Ця традиційна, але швидша архітектура *DRAM*, не вимагає значних змін в інтерфейсі графічного процесору для використання у відеопідсистемі.

***EDO DRAM* (*Extended Data Out DRAM*)** використовує стандартний інтерфейс *DRAM*, але передавання даних до/з пам'яті здійснюється з вищою швидкістю (на вищій частоті). Залежно від графічного контролера, *EDO DRAM* може мати продуктивність на рівні дорожчої двопортової технології пам'яті, такої як *VRAM*, яку використовують у графічних контролерах для систем на базі *OC Windows*.

***Burst EDO*** характеризує те, що: додатковий пакет регістрів забезпечує швидке виведення рядка з послідовних адрес; маємо довгий час очікування, якщо наступна адреса не є сусідньою в послідовності.

***VRAM*** – технологія двопортової пам'яті, що є гарним рішенням для створення буферів кадру з високою продуктивністю.

Будучи не дешевим варіантом, *VRAM* є найкращою альтернативою для застосувань, яким потрібний дозвіл  $1280 \times 1024$  при *true color* (особливо – з подвійною буферизацією).

*VRAM* є альтернативою *DRAM*, призначеною подолати обмеження з продуктивності шляхом зчитування та запису даних з банків пам'яті за один цикл.

Використання *VRAM* потребує спеціального контролеру.

Ринок цих типів пам'яті чітко сегментовано:

– *VRAM* використовують у системах, розрахованих на високий дозвіл та можливість роботи з 24-бітовим поданням кольору (наприклад, у системах *CAD*, верстання та кольороподілу);

– *DRAM* використовують у стандартних графічних підсистемах, розрахованих на офісне застосування (електронні таблиці, текстові процесори).

Але все частіше замість *VRAM* та *DRAM* в їх традиційних галузях застосування використовують нові типи пам'яті.

**Нові архітектури пам'яті *3D RAM*, *CDRAM*, *MDRAM*, *RDRAM*, *SGRAM/SDRAM*, *Embedded RAM*, *WRAM*.** Ці технології пропонують кращу продуктивність у порівнянні з *DRAM* та вигіднішу ціну, ніж *VRAM*.

***3D RAM*** – технологія робочих станцій для обробки *3D*-графіки, яку характеризують: вбудовані обчислювальні засоби та кеш-пам'ять, які реалізовано на рівні чіпу; висока оптимізація виконання *3D*-операцій. Це – комбінація кращих властивостей архітектур *VRAM*, *DRAM* та *embedded memory* (вбудованої пам'яті).

***CDRAM (Cached DRAM – кешована пам'ять)*** – попередник *3D RAM* із вбудованим до мікросхеми кешем.

*CDRAM* є ідеальною основою текстурної пам'яті та органічним доповненням до пам'яті *3D RAM* з її високою пропускну здатністю. При розміщенні елементів текстур (*texels*) у кеші, мінімізується час випадкового читання текселів та забезпечується приріст продуктивності.

**MDRAM** (*Multibank – багатобанкову DRAM*) характеризують висока пропускна здатність, малі затримки часу, малі чарунки. В середовищі *DOS* *MDRAM* досягає відмінних результатів, у середовищі *Windows* – задовільних.

**RDRAM** компанії *Rambus* є претендентом на широке розповсюдження та затвердження стандартом пам'яті з високою продуктивністю. *Rambus* використовує для *RDRAM* власну внутрішню шину для оперування потоками з великою кількістю даних.

**Синхронна DRAM** (*Synchronous DRAM – SDRAM*) та **синхронна графічна пам'ять** (*Synchronous Graphics RAM – SGRAM*) є високопродуктивними нащадками пам'яті типу *DRAM*.

**SDRAM** виробляють за стандартами *JEDEC*. Вона має більшу продуктивність, ніж *DRAM*. *SDRAM* частіше використовують як основну системну пам'ять, ніж у графічних адаптерах.

**SGRAM** теж виробляють за стандартами *JEDEC*. Вона є різновидом *SDRAM* та є однопортовою. Продуктивність *SGRAM* оптимізовано для графічних операцій. Але при цьому вона має характеристики, властиві високошвидкісній пам'яті, що дозволяють використовувати *SGRAM* для зберігання текстур та *Z*-буферизації.

*SGRAM* має унікальні властивості, що є більшими та кращими, ніж у *SDRAM*, забезпечують високу швидкість обробки графіки. *SGRAM* ідеально підходить для графічних адаптерів з одним недорогим банком пам'яті, що використовують для *2D/3D*-графіки та цифрового відео.

У реалізації *SGRAM* ще більше оптимізовано рух даних у порівнянні з *SDRAM*.

У *SDRAM* та *SGRAM* швидкість пересування даних збільшено завдяки повному контролю за часовими витратами, потрібними різним елементам чипів пам'яті для функціонування.

*SDRAM* та *SGRAM* потенційно можуть повністю замінити стандартну пам'ять *DRAM*. Ці типи пам'яті мають просту архітектуру та не вимагають надто багато спеціалізованих керувальних схем. З часом вартість *SDRAM* та *SGRAM* буде на рівні або нижче вартості чипів *DRAM*: цьому сприяють численні постачальники, масове виробництво та поява промислових стандартів та специфікацій. Безсумнівними перевагами *SDRAM* та *SGRAM*, у порівнянні з *DRAM*, є велика пропускна здатність та ємність чипів пам'яті.

***Embedded RAM*** є вбудованою пам'яттю. Архітектура *Embedded RAM* тісно пов'язує процесор та пам'ять, знижуючи необхідність у додаткових тактах та сигналах синхронізації та зменшуючи відстань, яку повинні долати дані між компонентами.

Основні переваги *Embedded RAM* – збільшення продуктивності та нижча собівартість. Недоліками *Embedded RAM* є негнучкість у використанні та можливі складності модернізації.

***WRAM (Windows RAM)*** – високошвидкісна двопортова технологія пам'яті, що за дизайном аналогічна *VRAM* та *RDRAM*. *WRAM* – нестандартний тип пам'яті, що вимагає використання спеціальної технології в контролерах. Два порти пам'яті *WRAM* забезпечують одночасне введення та виведення графічних даних. Інша перевага *WRAM* – прискорення операцій завдяки подвійній буферизації (*Double buffering*). *WRAM* має перевагу в ціні у порівнянні з *VRAM*, яку виробляє одна

компанія (*Samsung*), що не гарантує достатніх поставок для широкого застосування компонентів.

Новітні концепції архітектур відеопам'яті *FeRAM*, *DRDRAM*, *DDR SDRAM*, *ESDRAM*, *FCRAM*, *MRAM*.

*FeRAM* (*Ferroelectric RAM*) є енергонезалежним типом пам'яті, аналогічним *Flash*-пам'яті, що означає можливість зберігання даних без використання джерел енергії. Цей тип пам'яті є ефективним для використання в якості системної (операційної) пам'яті у ПК та графічних адаптерах.

*DRDRAM* (*Direct RDRAM*). Корпорація *Intel* ліцензувала архітектуру *RDRAM* у компанії *Rambus*, почавши сумісні розробки зі створення пам'яті *DRDRAM*. *Rambus* видала ліцензії на свою технологію виготовлення пам'яті найбільшим виробникам чипів *DRAM* та численним виробникам контролерів. Результатом вибору *Intel* технології *Rambus* може стати поява такої швидкої та досконалої пам'яті, яку будуть застосовувати повсюдно.

*DDR SDRAM* (*Double Data Rate SDRAM*) є покращенням архітектури *SDRAM*. Вона може передавати та приймати дані висхідним та низхідним рівнем сигналу шини, на відміну від *SDRAM*, яка передає дані тільки висхідним рівнем та має більшу ширину полоси пропускання в разі передавання довгих пакетів даних.

*ESDRAM* (*Enhanced, тобто покращена SDRAM*) – більш швидка версія *SDRAM*, розроблена згідно стандарту *JEDEC*. Продуктивність *ESDRAM* у два рази вища в порівнянні зі стандартною *SDRAM*. У більшості застосувань *ESDRAM* забезпечує більшу продуктивність, ніж *DDR SDRAM*.

Для використання всіх переваг *ESDRAM* необхідно застосувати спеціальний контролер (чіпсет). Існують чіпсети, що підтримують *ESDRAM* для застосування як системної (оперативної) пам'яті комп'ютеру.

**Fast Cycle Random Access Memory (FCRAM)** базується на принципово іншій концепції, в порівнянні з *DRAM*, завдяки якій час виконання циклу у три-чотири рази менший. За швидкістю роботи *FCRAM* нагадує *SRAM*, а за обсягом – *DRAM*.

**MRAM (Magnetic Random Access Memory)** є новою технологією пам'яті, здатною перевершити існуючі типи *DRAM* за швидкістю, обсягом та енергоспоживанням. Вона має видатні швидкісні якості (максимальний цикл читання займає 6 ns).

**Пристрої зберігання графічної інформації в зовнішній пам'яті ЕОМ.** Переважно ці пристрої використовують для зберігання масивів даних та програм, організації віртуальної пам'яті. Їх перевагою є набагато більша ємність та менша вартість, ніж у оперативної пам'яті.

За технологією запису та зчитування інформації на носій, накопичувачі класифікують на три основні категорії: магнітні, оптичні, магнітооптичні.

За функціональним призначенням, пристрої зберігання графічної інформації в зовнішній пам'яті ЕОМ розділяють на: гнучкі диски, стримери, компакт-диски, магнітооптичні накопичувачі, вінчестери.

Характеристики пристроїв зовнішньої пам'яті поділяють так:

– основні параметри всіх типів пристроїв, які фізично обмежують їх практичне застосування (ємність, швидкість зчитування та запису);

– економічні показники, що фінансово обмежують їх практичне застосування (вартість пристрою, вартість 1 Mb обсягу пам'яті, питома потужність, яка відіграє екологічне значення, ступінь надійності);

– ергономічні показники, що визначають комфортність праці людини (рівень шуму та вібрацій);

– інші показники (наприклад, час доступу для пристроїв прямого доступу).

Ці переваги зовнішньої пам'яті особливо актуальні в САПР, які є ресурсоемними застосуваннями: найчастіше застосовують жорсткі диски, що забезпечують максимальну швидкість роботи з даними та максимальний обсяг одночасно доступної інформації; *CD* широко використовують для перенесення ПЗ та резервного копіювання даних (*CD-R*, *CD-RW*); рідко застосовують стримери, гнучкі та магнітооптичні диски, що не забезпечують потрібної швидкості та надійності в умовах розвитку технології САПР.

## Питання для самоперевірки

1. Опишіть особливості розташування графічної інформації у внутрішній пам'яті ЕОМ.

2. Виконайте порівняльний аналіз основних видів архітектур пам'яті ЕОМ, які використовуються в комп'ютерній графіці.

3. Коротко охарактеризуйте пристрої зберігання графічної інформації в зовнішній пам'яті ЕОМ.

## 5.5. Пристрої передавання графічної інформації

Ці пристрої призначені для передавання графічної інформації каналами внутрішнього комп'ютерного та міжкомп'ютерного зв'язку [47, 104].

**Шина *Accelerated Graphics Port (AGP)*.** Ця шина дозволяє відображати графіку швидше та реалістичніше, ніж традиційні шини *ISA* та *PCI*, що не справляються з рядом задач.

Стандарт передавання даних *AGP* дозволяє графічній карті використовувати ОЗП ПК: за допомогою незалежної високошвидкісної шини відеокарта обмінюється даними напряму з оперативною пам'яттю, застосовуючи метод доступу *Direct Memory Execute (DIME)* фірми *Intel* (він дозволяє використовувати саме той обсяг ОЗП, який є потрібним). Щоб відповідати вимогам нової технології, стали потрібними нова материнська плата та графічна карта з шиною *AGP*, ПЗ, підтримка *AGP* операційною системою (*Microsoft* реалізувала це в *Windows*). Шина *AGP* є високошвидкісним прямим зв'язком між ОЗП ПК та графічним прискорювачем, який підключено безпосередньо до *AGP*-шини та якому не потрібно ділити загальну шину з іншими пристроями.

Таким чином, прогресивною технологією є прискорений графічний порт *AGP*, який зв'язує графічну плату безпосередньо з головною пам'яттю ПК: завдяки прямому зв'язку, графічні плати швидше переміщують екраном великі текстурні карти, підвищуючи гладкість виведення рухомих зображень.

Апаратні пристрої доступу до *Internet* та комп'ютерна графіка. *Існує три основних способи з'єднання комп'ютерів*



*для обміну інформацією: безпосередньо крізь асинхронний порт; з використанням модему; крізь локальні мережі* [38].

Попит на модеми, факс-модеми та радіомодеми є високим: вони дозволяють швидко передавати з комп'ютеру на комп'ютер пакети документів та зв'язуватися по електронній пошті, забезпечують доступ до глобальних світових мереж для встановлювання контактів із зарубіжними партнерами. Модеми стають невід'ємною частиною комп'ютеру: встановивши на ПК модем, ви перетворюєте його в ланку глобальної мережі. *За допомогою модему можна, не виходячи з дому:* отримати доступ до БД, які знаходяться на великій відстані; розташувати повідомлення на доступній для інших користувачів електронній дошці оголошень (*BBS*) або скопіювати з неї потрібну інформацію (файл); інтегрувати домашній ПК до мережі офісу, створюючи повноцінне відчуття роботи в цій мережі (окрім нижчої швидкості обміну даними); користуючись глобальними мережами (*Internet, RelCom, FidoNet*), приймати та надсилати електронні листи в будь-який кінець світу, а також приймати участь в електронних конференціях та отримувати новини за будь-якою тематикою.

За конструктивним виконанням, модеми поділяють так: внутрішній модем; зовнішній модем; модем у вигляді *PC*-карти для підключення до портативного ПК.

*Апаратні пристрої доступу до Internet поділяють на два класи* [36]: проводові та безпроводові. Перші мережні технології здійснювали передавання даних крізь проводи (модем та *Internet*-термінал, коаксіальні кабелі, оптоволоконні лінії). Їх переваги – низька ціна пристроїв та послуг, а недоліки – високий рівень перешкод та затухання переданого сигналу

(окрім оптоволоконних ліній), мала дальність. Безпроводові пристрої з'явилися в зв'язку з розробкою технологій передавання інформації без використання проводів (радіомодеми, сотові телефони, супутникові тарілки). Їх перевагами є великий радіус дії та висока перешкодостійкість, а недоліками – висока вартість пристроїв та оплати послуг.

Розглянемо особливості передавання мережами інформації графічного типу (мультимедійних даних). Специфіка цих даних полягає в їх великому обсязі та, як наслідок, низькій швидкості передавання мережами. Вимоги до АЗ полягають у забезпеченні передавання даних графічного типу з прийнятними швидкістю та якістю.

Для успішного вирішення цієї проблеми необхідно: вдосконалювати протоколи передавання/приймання інформації; збільшувати пропускну здатність каналів передавання даних шляхом побудови оптоволоконних ліній, використання супутникового передавання, вдосконалювання існуючих ліній (переходу на цифрові лінії, вдосконалювання мережного ПЗ); застосовувати компромісні методи (компресію інформації, яку передають, вдосконалювання форматів зберігання мультимедійних даних) [133].

**Пристрої інфрачервоних комунікацій.** Ці пристрої використовують у якості фізичного носія інфрачервоне випромінювання. Застосування ІЧ-світла як засобу зв'язку має багато переваг перед іншими технологіями комунікації: простота та дешевизна приймально-передавального обладнання; невеликі габаритні розміри ІЧ-пристроїв комунікації (на відміну від радіоаналогів, їм не потрібна антена для передавання даних, а тому вони можуть бути значно зменшені); незначний рівень

споживання енергії ІЧ-пристроями (оскільки ІЧ-зв'язок призначено для невеликих відстаней, енергетичні потреби залишаються в розумно малих межах, що дозволяє подовжити час роботи мобільних пристроїв без батарей); світло ІЧ-діапазону є невидимим, безпечним для людини, не корелює з освітлювальними приладами та радіоприладами. Сильною стороною ІЧ-технологій є швидке встановлювання зв'язку між комп'ютерами без наявності кабелів. Основним недоліком є те, що світло ІЧ-діапазону не проходить навіть крізь аркуш паперу.

Наприклад, фірма *Hewlett-Packard* випустила декілька сімейств лазерних принтерів, які враховують потреби друку користувачів мобільних цифрових та комп'ютерних пристроїв із вбудованими інфрачервоними портами.

### Питання для самоперевірки

1. Сформулюйте переваги *Accelerated Graphics Port*.
2. Назвіть апаратні пристрої доступу до Internet, які використовуються в комп'ютерній графіці.
3. У чому полягають особливості передавання мережами інформації графічного типу та яку роль відіграють у комп'ютерній графіці пристрої інфрачервоних комунікацій?

## 5.6. Комбіноване графічне обладнання

**Графічні робочі столи.** Ці комбіновані багатофункціональні пристрої містять: сканер; принтер; копіювальний апарат; факс; модем; потужний сервер, здатний містити великий обсяг графічних даних та обслуговувати всі робочі станції (якщо робочі столи поєднано мережею); плотер та дигітайзер (якщо це обумовлене конкретною діяльністю, наприклад, малим видавництвом).

**Безперечні переваги графічних робочих столів такі:** всі дії пристроїв, які їх утворюють (сканування, друк, копіювання, надсилання та отримання факсів) можна виконувати за допомогою єдиного, компактного та простого у використанні пристрою без плутанини проводів, який пропонує нові функції.

Найтиповішими комбінованими багатофункціональними пристроями для робочих столів є вироби фірми *Brother*, з яких ми розглянемо апарати *MFC 740* та *MFC 9550*.

*MFC 740*: переважає інші струминні пристрої якістю та швидкістю роботи; має всі важливі функції; містить телефонну трубку, автовідповідач та інтерфейс для цифрових камер; вправно працює з факсами (управління є чітким, спеціальні завдання призначаються швидко, відправка факсів з будь-якого додатку виконується безпроблемно); забезпечує зручне копіювання (наприклад, кольорові копії можна отримати без ПК, установки яскравості виконуються легко, ступені збільшення/зменшення можна обирати в межах 50-200 % із кроком 1 %); у процесі сканування, крізь інтерфейс *TWAIN*-модулю оригінали можуть потрапляти майже до будь-якого додатку обробки зображень, а додана *OCR*-програма розпізнає текст.

*MFC 9550* має гірше ПЗ, ніж у свого конкурента *HP*, але: є швидшим у друці, скануванні та копіюванні; завдяки високій швидкості роботи та гарному оснащенню, є лідером серед лазерних пристроїв; у процесі роботи з факсами надає багато функцій, наприклад, автовідповідач та шість різних способів виклику, виконання управління за допомогою численних клавіш на апараті.

**Метою комбінованих багатофункціональних пристроїв офісної спрямованості** є робота з діловими документами, не надаючи професійної якості та швидкості роботи з графікою.

**Мінітипографії. Як правило, вони містять такий набір пристроїв:** сканери; принтери; фотовивідні пристрої; кольоропробні автомати; копіювальні рами; аркушепідбиральне та брошуровочне обладнання; клейові та різальні машини. Вони дозволяють видавати та тиражувати літературу достатньо високого класу з кольоровим друком.

Визнаним лідером у виробництві мінітипографій є фірма *HAMADA*.

Модель *Hamada 234* – одна з найпопулярніших малих друкувальних машин для оперативної поліграфії. Її характеризують: висока надійність; вбудований мікрокомп'ютер, який керує процесом друку; невеликі габаритні розміри. Вона вважається ідеальною для невеликих типографій, де забезпечує відмінну якість друку.

Модель *Hamada Super 47* – малогабаритна друкувальна машина формату *A3+* легкого класу для друку невеликих тиражів кольорової продукції. Вона є цікавою для малих типографій та друкувальних салонів, а підприємства-початківці приваблює невисокою ціною та простотою обслуговування.

Модель *Hamada C248L* – серія малих друкувальних машин середнього класу для поліграфічного виробництва зі світовим рівнем якості. Їх можна використовувати двозмінно без зниження якості виробництва для середніх тиражів (1000-50000 примірників) завдяки надійності конструкції та простоті обслуговування.

Модель *Hamada B5210K* – серія малих друкувальних машин важкого класу для тризмінного виробництва продукції світового рівня якості, які можуть працювати без збоїв більше десятиріччя.

## Питання та завдання для самоперевірки

1. Назвіть складові частини та сформулюйте переваги мінітипографій та графічних робочих столів.

2. Вивчіть та проаналізуйте особливості організації та функціонування апаратного забезпечення графічних підсистем АСУ, АСУТП та САПР на прикладі декількох підприємств та установ вашого міста (району, області).

3. Дослідіть декілька найкращих вітчизняних та зарубіжних зразків упорядкування апаратного забезпечення графічних підсистем великих комп'ютеризованих виробництв (комплексів) за матеріалами фундаментальних посібників, періодичної преси, науково-практичних конференцій, *Web*-сторінок мережі *Internet*.

## Навчальний практикум до гл.4-5

У процесі виконання завдань цього практикуму необхідно: ввести до комп'ютеру графічні зображення визначених типів за допомогою заданих класів пристроїв введення; подати введені зображення за допомогою певних файлових графічних форматів, проаналізувати та порівняти отримані результати; вивести отримані зображення на вказані носії за допомогою вказаних пристроїв виведення; зберегти зображення на певних електронних носіях; передати зображення мережею за вказаною адресою; проаналізувати отримані результати та висунути альтернативні підходи до роботи з початковою графічною інформацією, обґрунтувавши їх ефективність.

Нижче подано переліки наступних значень: типів початкової графічної інформації, яку потрібно ввести до комп'ютеру; класів пристроїв введення початкової графічної інформації до комп'ютеру; файлових графічних форматів, у яких потрібно подати ту графічну інформацію, що було введено; класів пристроїв виведення графічної інформації; типів носіїв для виведення зображень; типів електронних носіїв для зберігання зображень; типів адрес для відсилання графічної інформації.

Табл.5.1 встановлює відповідність між номером варіанту індивідуального контрольного завдання студенту та набором номерів елементів наведених нижче переліків (у залежності від додаткових характеристик початкового графічного зображення, до його номеру в табл.5.1. дописують, у дужках та через кому, певні літери: "Ц" – цифрове; "Т" – традиційне; "К" – кольорове; "М" – монохромне; "О" – являє собою авторський оригінал;

"Д" – розташоване у друкованому виданні; "С" – подане описом сукупності координат точок, які його складають).

Таблиця 5.1

## Варіанти індивідуальних контрольних завдань

№ п/п	Набори номерів елементів переліків 1-7						
	Перелік 1	Перелік 2	Перелік 3	Перелік 4	Перелік 5	Перелік 6	Перелік 7
1	1(Ц,К,О)	2	2	11,15	1	3	4
2	11(Т,М,С)	1	3	2,16	2	2	2
3	2(Ц,К,О)	7	1	12,17	3	1	4
4	12(Т,К,Д)	5	8	9,15	1	3	1
5	3(Т,М,О)	7	2	11,16	3	2	3
6	13(Т,К,О)	2	4	10,17	3	1	2
7	4(Ц,К,О)	3	12	15,16	6	3	3
8	14(Т,М,О)	2	5	1,17	1	2	1
9	5(Т,К,С)	9	9	9,15	2	1	4
10	15(Т,К,О)	2	7	4,16	5	3	2
11	6(Т,М,Д)	1	6	13,17	4	2	1
12	16(Т,К,О)	2	4	6,15	4	1	3
13	7(Т,К,С)	9	10	14,16	1	3	2
14	17(Т,М,О)	2	14	12,17	1	2	4
15	8(Т,М,О)	6	2	3,15	1	1	3
16	9(Т,М,С)	9	3	5,16	1	3	2
17	10(Т,К,Д)	5	4	8,17	3	2	3
18	1(Ц,М,Д)	1	14	14,15	3	1	1
19	4(Т,К,Д)	8	13	16,17	6	3	2
20	2(Т,К,О)	7	5	12,15	3	2	4
21	5(Т,М,О)	4	15	9,16	2	1	3
22	6(Т,К,О)	1	16	10,17	1	3	1
23	7(Т,К,О)	5	17	9,15	2	2	2
24	8(Т,К,О)	6	5	7,16	3	1	3
25	9(Т,К,Д)	4	18	5,17	2	3	2
26	10(Т,К,Д)	1	11	4,15	5	2	3
27	11(Т,К,Д)	6	19	8,16	3	1	1
28	12(Т,К,О)	5	20	14,17	3	3	2
29	13(Т,К,Д)	2	8	13,15	4	2	3
30	10(Т,К,Д)	4	11	2,16	2	1	4



**Перелік 1. Типи початкової графічної інформації, яку потрібно ввести до комп'ютеру:** 1) фотографія; 2) слайд; 3) мікрофільм; 4) відеоролик; 5) ескіз; 6) векторний рисунок; 7) растровий рисунок; 8) виробниче 2D-креслення; 9) зображення каркасного 3D-тіла; 10) зображення твердого 3D-тіла; 11) карта місцевості; 12) ілюстрований текст з довільним набором таблиць, формул, рисунків, фотографій та спецефектів; 13) художня картина; 14) гравіювання; 15) вітраж; 16) мозаїка; 17) різьблення.

**Перелік 2. Класи пристроїв введення початкової графічної інформації до комп'ютеру:** 1) графічний планшет; 2) цифрова фотокамера; 3) цифрова відеокамера; 4) ручний сканер; 5) планшетний сканер; 6) барабанний сканер; 7) проекційний сканер; 8) відеодигітайзер; 9) пристрій мовного введення.

**Перелік 3. Файлові графічні формати, в яких потрібно подати ту графічну інформацію, що було введено:** 1) *BMP*; 2) *GIF*; 3) *PNG*; 4) *JPEG*; 5) *FLASHPIX*; 6) *PCX*; 7) *EPS*; 8) *PDF*; 9) *TIFF*; 10) *RAW*; 11) *VRML*; 12) *MPEG*; 13) *MJPEG*; 14) *IFF*; 15) *MAC*; 16) *PICT*; 17) *PixelPaint*; 18) *PIXAR*; 19) *Sci-text*; 20) *TGA*.

**Перелік 4. Класи пристроїв виведення графічної інформації:** 1) пір'їнний плотер; 2) олівцево-пір'їнний плотер; 3) струминний плотер; 4) електростатичний плотер; 5) *DIP*-плотер; 6) *TTP*-плотер; 7) лазерний плотер; 8) *LED*-плотер; 9) матричний принтер; 10) струминний принтер; 11) лазерний принтер; 12) *LED*-принтер; 13) термопринтер; 14) принтер зі змінюваням фази барвника; 15) ЕПТТ-монітор; 16) РКД-монітор; 17) плазмовий монітор.

**Перелік 5. Типи носіїв для виведення зображень:** 1) звичайний папір високої якості; 2) звичайний папір низької якості; 3) фотопапір; 4) термопапір; 5) електростатичний папір; 6) плівка.

**Перелік 6. Типи електронних носіїв для зберігання зображень:** 1) магнітний; 2) оптичний; 3) магнітооптичний.

**Перелік 7. Типи адрес для відсилання графічної інформації:** 1) *E-mail*; 2) *Web*; 3) інша робоча станція у локальній мережі; 4) *Fax*.

## ГЛАВА 6. ТЕХНОЛОГІЇ 3D-ГРАФІКИ

### 6.1. Введення до 3D-графіки

Роль засобів апаратного прискорення в процесі створення тривимірних зображень. Технологія 3D-графіки, розглядаючи графічне подання сцени або об'єкту відносно трьох осей (висота, ширина та глибина), дозволяє відтворити 3D-зображення на пласкому дисплеї, вносячи реалізм та інтерактивність у графічні програмні додатки.

Для відображення 3D-анімації об'єкт (машину, літак, віртуальний світ тощо) спочатку зображують опорними точками та векторними лініями в тривимірній системі координат. Векторний остов об'єкту зберігають у системній пам'яті. Потім здійснюють рендеринг об'єкту: визначають фізичний стан кожного пікселя чи групи взаємопов'язаних пікселів відносно обраної точки зору, враховують освітлення та в рамках видимих меж прорисовують зображення, що складається з зафарбованих полігонів із накладеними тінями [28, 63, 120].

Значний внесок до вирішення проблеми побудови на комп'ютері високоякісних реалістичних 3D-зображень належить засобам апаратної підтримки. З одного боку, існує велика кількість спеціалізованих комп'ютерів – графічних робочих станцій, що відмінно справляються з поставленими перед ними задачами, але мають високу ціну. Це супер-EOM та потужні станції таких виробників, як *SGI (Silicon Graphics)*, *Intel*, *Macintosh*. З іншого боку, є спеціалізовані графічні 3D-плати

(3D-прискорювачі, 3D-акселератори, 3D-відеокарти, 3D-комбайни), що доступні за ціною та забезпечують високу якість в межах своєї області функціональних можливостей та прикладних задач для широкого кола людей. Найбільшого розповсюдження набули IBM-сумісні ПЕОМ та 3D-прискорювачі для них [47, 63, 120].

Широке впровадження апаратного забезпечення 3D-графіки викликало збільшення потужності комп'ютерів без суттєвого збільшення ціни. Велика кількість нових 3D-карт дозволила користувачам бачити якісну та швидку 3D-графіку в реальному часі на своїх ПК. 3D-акселератори дозволяють додавати зображенням реалізму та прискорювати виведення графіки в обхід центрального процесору, спираючись на власні апаратні можливості [91, 120].

Найбільш широко 3D-можливості використовуються в іграх. Ділові та промислові застосування теж використовують їх все інтенсивніше. Наприклад, автоматизоване проектування гостро потребує виведення 3D-об'єктів. Особливо це стосується галузі твердотілого моделювання в машинобудівних САПР [79]. Завдяки 3D-акселераторам проектування та візуалізація 3D-об'єктів для цієї сфери діяльності можливі на якісному рівні на звичайному ПК. 3D-графіка покращує спосіб взаємодії людини з комп'ютером. Використання 3D-інтерфейсів програм робить процес спілкування з комп'ютером набагато простішим.

Найважливішими параметрами графічної підсистеми є продуктивність, якість зображення та ціна. 3D-акселератор, орієнтований на мультимедіа-ринок, повинен забезпечувати високу швидкість прорисовування анімації (частоту кадрів), поліпшувати якість та підвищувати реалістичність візуалізації в

порівнянні з *SVGA*-платами [88]. Оскільки багатьом користувачам швидка графічна система потрібна для комп'ютерних ігор, їм не підійдуть технології професійних графічних станцій (знайдеться мало охочих викласти декілька тисяч доларів за графічний процесор, геометричний співпроцесор та великий обсяг ОЗП). Тому основною задачею розробників мультимедійних *3D*-акселераторів є інтеграція максимальної кількості *3D*-функцій за умови використання мінімального обсягу тих ресурсів, які коштують дорого [104].

*3D*-акселератори поділяють на дві групи: ті, що використовують власні оригінальні технології прискорення *3D*-графіки; ті, що оптимізовано для стандартних *API*.

**Витоки становлення *3D*-індустрії.** З'явившись в другій половині 1990-х років, *3D*-акселератори спочатку не отримали широкого розповсюдження через нерозвиненість індустрії *3D*-ігор: *3D*-плати використовувалися професійними розробниками *3D*-графіки, застосовувалися в спеціалізованих графічних станціях, коштували дорого та були оптимізовані для розв'язування певного кола задач (апаратного текстурування, *Z*-буферизації, прорахунку світлових та атмосферних ефектів). Як правило акселератори використовувалися для попереднього обчислення *3D*-сцен, не встигаючи обчислювати їх у режимі реального часу. Якщо апаратну акселерацію застосовували для фінального рендерінгу, доводилося на протязі декількох годин обчислювати кадр за кадром з тим, щоб потім поєднати їх в один зв'язний фільм. Акселератор був потрібен для скорочення часу обчислень з декількох днів до декількох годин, а системи реального часу вважалися унікальними.

З появою численних 3D-ігор, в яких сцени динамічно, в режимі реального часу змінюються залежно від дій гравця, 3D-акселератори вийшли на масовий ринок. З удосконаленням ігор ускладнювалися 3D-обчислення, необхідні для візуалізації реалістичних сцен, а центральні процесори не могли забезпечити належної швидкості відображення, через що 3D-акселератори стали нагально необхідні. Змінилася головна задача акселераторів: першочерговою стала не якість згенерованого зображення, а швидкість візуалізації.

Ігровий світ, який створюють 3D-акселератори на ПК у режимі реального часу, поступається за якістю 3D-графіці, згенерованій спеціалізованими графічними пакетами з застосуванням програмного рендерінгу. Проте ті ігри, що працюють з акселераторами, набагато реалістичніші за ті, в яких для візуалізації 3D-сцен використовують програмну реалізацію та можливості ЦП. Акселератор встигає не тільки текстурувати помешкання, рослинність та персонажів, а й обчислювати спеціальні ефекти, наприклад фільтрувати текстури (з наближенням до предмету його покриття не виглядає як нагромадження різнокольорових квадратиків, а тільки стає розмитішим), застосовувати світлові ефекти (відблиски, тіні та інші), використовувати напівпрозорі матеріали та атмосферні ефекти (дим, туман) тощо.

Уперше з'явившись в ПК виключно для ігрових застосувань, 3D-акселератори, що підтримують графічний стандарт *OpenGL*, стали успішно застосовувати для розв'язування традиційних задач у пакетах 3D-графіки та анімації.

Нагадаємо термінологію, що буде першочергово потрібна для подальшого розгляду. Глибина кольору описує, як багато кольорів може одночасно виводити відеоадаптер: 8-бітова глибина відповідає 256 кольорам, 16-бітова – 65536 кольорам, 24-бітова – 16 млн. кольорів. За аббревіатурою *RAMDAC* приховано цифро-аналоговий перетворювач (ЦАП): він приймає від ПК цифрову інформацію та перетворює її в аналоговий сигнал, який відображається на моніторі. Чим швидший *RAMDAC*, тим вищою може бути частота регенерації екрану (кількість кадрів, яка перерисовується за секунду). Якщо плата використовує відеоінтерполяцію, то при збільшенні вікна з відеозображенням краї об'єктів не стають "зубчастими". Віддають перевагу інтерполяції по обох осях. У 3D-графіці під *API* розуміють спеціальну програму, що дозволяє програмному 3D-додатку взаємодіяти з 3D-відеоадаптером. Щоб 3D-гра швидше виконувалася, вона повинна бути написаною з тим *API*, що підтримується платою.

### Питання для самоперевірки

1. Що таке 3D-акселератор?
2. Як формувалася 3D-індустрія?
3. Що означає термін *RAMDAC*?

## 6.2. Методи побудови 3D-зображень

Розглянемо постановку проблеми побудови високоякісних реалістичних 3D-зображень, роль апаратної та програмної складових частин у процесі її вирішення, методи її вирішення.

**Основні поняття про 3D-конвеєр.** Тривимірне зображення – це відображення предметів у 3D-просторі. Оскільки 3D-об'єкти визначають за допомогою математичної моделі, вона є початком відліку послідовності отримання зображення на екрані ПК, яка має назву *3D-Pipeline* (3D-конвеєр) [63, 64, 79]. Схема побудови кінцевого зображення є стандартною для більшості ПЗ. Деякі програми реалізують всі стадії 3D-конвеєру, деякі перекладають частину роботи на апаратні пристрої, спеціальні бібліотеки програм (*API*), інші програми або на користувача. 3D-конвеєр складається з таких етапів [63, 120, 137-139].

**Етап 1 – визначення стану об'єктів (*Situation modeling*).** Цей етап не має прямого відношення до комп'ютерної графіки, моделюючи той світ, який буде відображатися на подальших етапах конвеєру.

**Етап 2 – визначення геометричних моделей, відповідних поточному стану (*Geometry generation*).** Цей етап створює геометричне подання поточного моменту віртуального світу.

**Етап 3 – розбиття геометричних моделей на примітиви (*Tessellation*) та побудова каркасної моделі об'єкту (*триангуляція*).** Це перша залежна від "заліза" стадія, на якій створюється зовнішній вигляд об'єктів у формі набору визначених геометричних примітивів, на основі інформації з



попереднього кроку конвесру. Найбільш розповсюдженими примітивами є трикутники (більшість сучасних програм та прискорювачів працюють з ними). В цілому, виходячи з математичної моделі об'єкту, генерується його каркасна модель, що складається з трикутників.

#### *Етап 4 – видові геометричні перетворення (Projection).*

Каркасна модель об'єкту трансформується з урахуванням положення в просторі точки спостереження 3D-сцени, зміни положення об'єкту відносно неї (пересування, обертання, змінювання розмірів тощо), перспективних викривлень, розташування джерел світла. Визначаються нові координати для всіх вершин примітивів, виходячи з положення спостерігача та напрямку його погляду. Сцена "проекується" на поверхню монітору, перетворюючись у двовимірну, хоча інформація про відстань від спостерігача до вершин зберігається для подальшої обробки.

*Етап 5 – рендерінг.* На цьому етапі формується кінцевий зовнішній вигляд об'єкту. Головною задачею є додавання до каркасної моделі об'єкту характеристик його поверхонь. Кожен трикутник зафарбовується растровим зображенням (текстурою). На етапі рендерінгу задіюють просторову координату  $Z$  – глибину (відстань від точки перегляду до об'єкту). 3D-акселератори містять  $Z$ -буфер – блок пам'яті, в якому зберігаються координати глибини точок, які потрапляють до області перегляду. На основі просторових даних здійснюється визначення видимих та невидимих частин сцени. Рендерінг можна умовно поділити на чотири складові частини: растеризація; видалення прихованих поверхонь; накладання текстур; затінення. Для рендерінгу акселератору передається набір геометричних примітивів.

*Растеризація* – це перетворення трикутників у набори пікселів. *Видалення прихованих поверхонь* – це відсікання пікселів, які є невидимими з площини спостереження (його здійснюють за допомогою Z-буферу, що визначає, який з трикутників знаходиться найближче до спостерігача та перекриє інші трикутники). Якщо карта не містить Z-буферу, то процес перекладається на ЦП. *Замінення* використовують для відтворення ефектів світла та тіні.

У цілому етап рендерінгу містить такі підетапи.

**Етап 5.1 – прив'язка текстур та освітлення (*Texture and light definition*).** На цій стадії визначається: як будуть освітлені геометричні примітиви (трикутники); які та яким чином на них будуть накладені текстури (*Textures*) – зображення, що передають зовнішній вигляд об'єкту (негеометричну візуальну інформацію).

**Етап 5.2 – відкидання невидимих примітивів (*Culling*).** На цій стадії зі списку примітивів виключаються повністю невидимі – ті, що залишаються позаду або збоку від зони видимості.

**Етап 5.3 – установка примітивів (*Setup*).** Інформація про примітиви (координати вершин, накладання текстур, освітлення тощо) перетворюється в вигляд, придатний для наступної стадії.

**Етап 5.4 – зафарбовування примітивів (*Fill*).** На цій стадії здійснюється побудова в буфері кадру зображення на основі тієї інформації про примітиви, яку було сформовано попередньою стадією конвейсру, та інших даних. Більшість характеристик прискорювачу, що можна взяти з його опису, відносяться до стадії зафарбовування примітивів (в основному

саме цю стадію конвеєру прискорюють апаратно в разі недорогих та доступних плат).

**Етап 5.5 – фінальна обробка (Post processing).** Це обробка всього підсумкового зображення як єдиного цілого деякими двовимірними ефектами.

Звичайно на апаратну частину покладають найбільш трудомісткі етапи побудови зображення: реалізація алгоритму на апаратному рівні підвищує продуктивність у сотні, тисячі та мільйони разів, залежно від складності алгоритму.

Наведемо перелік ефектів, які реалізовано на апаратному рівні в більшості 3D-прискорювачів: накладання туману, відбите світло, розмиті тіні, прозорість, напівпрозорість, затінення, атмосферні ефекти, перетворення текстур, відбиття. Ці функції є найбільш загальними ефектами, доступними при створенні зображення. Окрім них є велика кількість інших функціональних перетворень, які реалізуються на апаратному рівні (детальну інформацію про них можна отримати в описах технічних характеристик 3D-прискорювачів).

Першочерговою метою 3D-карт є вивільнення головного процесору від попиксельного прораховування 3D-сцени. ПЗ розраховує просторове положення предметів, які необхідно відобразити, та передає це положення графічній карті. При цьому об'єкти складаються з величезної кількості многокутників. Кожному многокутнику призначається певна текстура, а каркас із многокутників покривається поверхнею, що надає йому вигляду твердого просторового тіла. Роботу з текстурами бере на себе графічна карта. Вона обчислює рядок за рядком, як необхідно накласти текстуру на каркас, щоб досягти ілюзії тривимірності зображення. Іншими словами, графічна карта

проектує просторову модель на площину та передає зображення з цієї площини на екран монітору.

За допомогою таких ефектів, як *Alpha Blending*, *Depth Cueing/Fogging*, *Specular Highlights*, *Color Key*, *Frame Rate*, *Hidden Line Removal*, *MIP mapping*, Z-буфер та багатьох інших здійснюються спроби досягнути максимальної реалістичності 3D-сцени.

Основні функції й ефекти, реалізацію яких підтримують 3D-акселератори. *Texture mapping* – це операція накладання на кожен із трикутників, які складають каркасну модель об'єкту, текстури – звичайної растрової бітової карти. Окремі пікселі в цій текстурі називаються текселями – від *texel* (*textured pixel*). Наприклад, у грі *Microsoft Monster Truck Madness* у процесі пересування ігровим полем графічна плата постійно надсилає на дюни бітову карту піску. Завдяки цій технології можна відтворювати лабіринти середньовічного замку з каменю, мурмурові покої королівського палацу та дизайн сучасного житла, "оживляти" полігональну модель людини, імітувати фактуру її вбрання та природний вигляд оточуючого її середовища, а також багато іншого.

*Nearest Mipmap* – це просте накладання текстур, яке поміщає текселі на об'єкт, який текстурують, використовуючи найближчий тексель за основу для визначення кольору.

*Perspective-correct texture mapping* (*накладання текстур із корекцією перспективи*) забезпечує, щоб тривимірні текстуровані об'єкти виглядали більш реалістично під різними кутами зору та на різних відстанях. У разі відсутності цієї функції текстури будуть постійно "сповзати" або прорисовуватися під неправильним кутом, порушуючи цілісність зображення.

*Bump mapping* дозволяє наносити на поверхню трикутників рельєфні текстури.

*MIP mapping (метод багаторівневих текстур)* передбачає таке. При нанесенні текстур на полігони їх часто доводиться розтягувати або стискати залежно від того, наближається об'єкт або віддаляється. Для того, щоб покращити деталізацію та якість зображення масштабованих об'єктів, використовують метод створення багаторівневих текстур *MIP mapping*, назва якого походить від латинського *multum in parvam* (багато в малому) та від *map* (накладання). Кожна текстура складається з декількох фрагментів (*MIP*-рівнів) різного розміру. *MIP*-рівень текстури визначається розміром полігону. Ця функція може бути програмною або апаратною (другий варіант реалізується за допомогою різних алгоритмів та вимагає високої продуктивності графічної підсистеми, що впливає на ціну).

Ця функція особливо доцільна в разі руху на об'єкт з текстурою. Метод створює серію копій текстури з різним графічним дозволом. У разі віддалення або наближення об'єкту, буде застосовано ту з них, яка найкраще "вписується" в об'єкт. Якщо об'єкт наближається, то використовується текстура з вищим дозволом, а якщо віддаляється – нижчим. Завдяки цьому методу зникає необхідність переобчислювати значення текстелів кожного разу, як об'єкт змінює своє положення в просторі.

Без *MIP*-накладання краї об'єкту з текстурою, що пересувається відносно спостерігача, будуть мерехтіти чи іскритися, оскільки мікросхема 3D-акселератору буде розтягувати чи стискати карту текстури для заповнювання об'єкту. При використанні функції *MIP*-накладання контролеру відеоплати дово-

диться менше займатися розтягуванням текстур, оскільки можна швидко вибрати текстуру більшого чи меншого розміру.

Для ефективного розподілу пам'яті, при роботі *MIP mapping* застосовують різні способи упакування однотипових карт текстур. Використовуються також алгоритми *Bilinear MIP mapping* (комбінація *Bilinear filtering* та *MIP mapping*) та *Trilinear MIP mapping* (про *Bilinear* та *Trilinear filtering* дивіться далі).

**Antialiasing** надає алгоритм легкого "розмиття" діагональних ліній для отримання ефекту плавності нахилу шляхом змінювання яскравості прилеглих пікселів. Його застосовують також для отримання чіткіших обрисів об'єктів та на межах різних текстур. Цей алгоритм задіює великі обчислювальні ресурси, особливо в разі використання його в зв'язці з *subpixel positioning* – методом розбиття пікселів на підпікселі при визначенні тих пікселів, крізь які буде проходити нахилена пряма.

**Linear Mipmap** – це накладання текстур, яке усереднює кольори оточуючих текселів для визначення кольору текстури, використовуючи так звану "білінійну фільтрацію".

**Filtering (фільтрація)** – це прибирання блочної структури в 3D-зображеннях. *Point-sampled, Bilinear, Trilinear, Anisotropic filtering* – методи видалення викривлень при пересуванні або обертанні текстурованих об'єктів шляхом інтерполяції деяких пікселів. Найчастіше ці викривлення являють собою перестрибування пікселів у невідповідні позиції та появу на тлі зображення циклічних візерунчастих утворень, що псує враження від кольорового дизайну зображення та порушує його цілісність.

**Point-sampled filtering** – це найпростіший алгоритм, який тиражує колір сусідніх текселів, суттєво втрачаючи якість.

**Bilinear filtering** інтерполує текстель, обчислюючи середнє значення кольору чотирьох найближчих текстелів. Це одразу поліпшує якість зображення, але підвищує ресурсоємність.

**Trilinear filtering** дозволяє отримати ще більш якісне зображення. При обчисленні текстеля алгоритм усереднює значення двох операцій *bilinear filtering*, застосованих до сусідніх МР-рівнів бітової карти текстури.

Недоліком методів *Point-sampled*, *Bilinear* та *Trilinear filtering* є те, що вони можуть бути застосовані тільки до плоских текстурованих поверхонь.

**Anisotropic filtering** дозволяє здійснювати фільтрацію текстурованих об'єктів, обернених під відповідним кутом відносно точки зору. Фільтрація здійснюється за різними напрямками з урахуванням повороту текстури.

**Palletized textures (метод оптимізованих текстур)** передбачає вирішення такої проблеми. Коли прорисовується нова сцена, всі необхідні текстури завантажуються з системної пам'яті до кадрового буферу для подальшого використання графічним процесором. Цей процес ставить найвищі вимоги до системних ресурсів комп'ютеру та графічного акселератору. Проте розробникам ігор доводиться орієнтуватися на мінімальну користувацьку конфігурацію. Саме виходячи з обсягу ОЗП стандартного графічного адаптеру, розраховується обсяг текстур, які можна буде використовувати, що призводить до зниження детальності та якості зображення.

Проте є метод оптимізації текстур, коли для кожної з них визначається таблиця перетворень кольорової палітри *Color Look Up Table (CLUT)*. Ця технологія дозволяє розробнику використовувати не звичайне шістнадцятирозрядне подання

(65 тисяч кольорів), а чотири- чи восьмирозрядне, економлячи місце та час, необхідні для завантаження текстур. Режим *CLUT-4* (16 кольорів) застосовують для простих текстур (цегли, каміння), а режим *CLUT-8* (256 кольорів) застосовують для складніших текстур.

*Texture transparency (прозорість текстур)* – це технологія, яку використовують для забезпечення прозорості текстур. Вона схожа на технологію *Chroma Key (Color Key, Blue Box)*, що широко застосовується у відео- та кінопромисловості при відображенні складних об'єктів. Цей метод передбачає, що деякий колір текстури (наприклад, синій) обирають за фарбу. Графічний чіп у процесі відображення об'єктів з цією текстурою рисує замість фарби-штампу колір об'єкту, розташованого глибше, а тому текстура місцями здається прозорою. Метод *Color Key* дозволяє дуже просто імітувати рослинність (наприклад, замість складного багатокутника в формі дерева рисують простий плаский багатокутник з текстурою листя, а простір між листям заповнюють фарбою-штампом, яка при виведенні замінюється фоном).

*Pixel Blending* – це змішування поточного пікселя з тим пікселем, який знаходиться в буфері кадру (*frame buffer*). *Add Pixel Blending* – це складання характеристик поточного пікселя з характеристиками пікселя в буфері кадру. *Texture Blending* – це змішування кольору текстури та кольору поверхні, на яку її накладають.

*Blending (змішування текстур)* – це візуальний ефект, який дозволяє розробнику "комбінувати" текстури перед нанесенням їх на об'єкт, досягаючи цікавих результатів та спецефектів. Найпростіше реалізувати цю функцію, задавши



ступінь прозорості верхньої текстури та прорисувавши тільки частину пікселів, які складають її зображення. Цей метод звичайно використовують у недорогих графічних акселераторах. У професійних системах застосовують метод *alpha blending* (альфа-змішування).

Метод *alpha blending* дозволяє створювати повністю чи частково прозорі об'єкти. Це спосіб змішування двох текстурних карт, які перекриваються, так, що одна текстура стає прозорою. Колір підсумкового зображення обчислюється як середнє між значенням кольору перекритого (розташованого далі) об'єкту та значенням кольору розташованого перед ним прозорого об'єкту.

У алгоритмі введено додатковий параметр – альфа-значення (коефіцієнт прозорості). Якщо перед одним об'єктом розташовано інший з ненульовим значенням коефіцієнту прозорості, то яскравість пікселів першого об'єкту зменшується або їх колір змінюється відповідно до кольору того об'єкту, що його закриває.

*Alpha blending* застосовується в разі використання 16-бітового кольору для створення складних ефектів, наприклад, для реалістичної візуалізації моря або зображення лазерного променя на фоні стіни (щоб промінь був схожий на дійсний, колір стіни потрібно змішати з кольором променя). Метод *alpha blending* застосовують у професійних системах: зчитування та обчислення з урахуванням перспективи значень обох початкових текстур вимагає від графічної підсистеми високої продуктивності.

Поняття *апаратного Z-буферу* визначає віддалення кожної точки зображення від спостерігача (її просторову глибину).

Так можна визначити, чи закриває один об'єкт інший. Використання *Z*-буферу (виділеного буферу в ОЗП графічної плати для зберігання даних про *3D*-об'єкти) особливо необхідне тоді, коли два об'єкти в *3D*-світі перетинаються відносно певної точки зору. Ця функція дозволяє визначити, яка частина одного об'єкту затулена іншим, тобто – як саме потрібно прорисувати зображення.

Для *Z*-буферу потрібний певний обсяг пам'яті в кадровому буфері, тому його застосування навіть у разі базових дозволів ( $320 \times 240$ ,  $640 \times 480$ ) можливе за умови наявності як мінімум 4 Мб ОЗП. Витрати на додаткову пам'ять виправдовують себе: все більша кількість ігор використовує *Z*-буфер, а без нього продуктивність суттєво знижується (оскільки обчислення зображень здійснюється програмно, через центральний процесор).

***Hidden Line Removal*** дозволяє вирішувати таку проблему: закриті об'єкти не повинні бути видимими (за винятком того випадку, коли розташований перед ними об'єкт є прозорим), оскільки рисувати їх означає марно витратити час та ресурси процесору.

Розпізнавання закритих ліній здійснюється або на графічній карті за допомогою *Z*-буферу, або шляхом обчислень у ЦП. Перший метод розвантажує ЦП, але не шину (приховані об'єкти передаються на карту). Тому виробники ігор часто обирають випробувані *Hidden-Line* алгоритми часів *2D*.

***Lighting (освітлення) та shading (затінення)*** відіграють велику роль у візуальному сприйнятті *3D*-світу, дозволяючи оптично згладити кутистість *3D*-моделей, акцентувати увагу на певних деталях та приховати непотрібні фрагменти. Апаратна реалізація функцій освітлення та затінення дозволяє досягти вищої якості зображення без участі ЦП.

Метод *Gуро* обчислює значення освітленості для вершин трикутника, а значення всередині нього інтерполюються.

Методи *Flat Shading* та *Gourand Shading* створюють ефекти затінення об'єктів. З їх допомогою зображення 3D-об'єктів стають реалістичнішими завдяки ефектам світла та тіні. Перший метод є простішим у реалізації, але не дає потрібної якості, на відміну від *Gourand Shading*.

Алгоритми *Lighting effects Fogging/Depth cueing* (використання туману, вуалювання) перетворюють вигляд текстуrowаних трикутників згідно з інформацією про джерела освітлення 3D-сцени.

Зі збільшенням відстані об'єкту від спостерігача, кольори об'єкту змінюються. Якщо вони поступово змінюються в напрямку білого кольору, то створюється уява, що в зображеному просторі стоїть туман, у якому зникають ті об'єкти, що знаходяться далі. Ефект *fogging* створює ілюзію димки (туману) різної насиченості.

Намагаючись розвантажити процесор, розробники застосовують спеціальні прийоми зниження обсягу даних, необхідних для рендерінгу сцен. Одним із цих прийомів є "напустити туману" – "приховати" частину зображення за пеленою, не прописуючи її, шляхом змішування кольорових значень текстур з білим кольором. Цей трюк часто використовують для побудови ландшафтів в авіасимуляторах. Апаратна реалізація цієї функції економить до 10-15 % ресурсів системи.

У загальному випадку вуалювання – це змішування кольору 3D-об'єкту з фіксованим кольором по мірі віддалення цього об'єкту, що на екрані комп'ютеру виглядає як зникнення об'єкту в тумані (темряві) або поява його звідтіля. За допомогою *Depth*

*cueing* змінюються колір та яскравість об'єктів, які віддаляються від точки перегляду або джерел освітлення.

Метод *Dithering* (*dithering*) надає реалістичності кольоровим переходам при використанні 16-бітових буферів. При визначенні проміжного кольору, що згладжує перехід, беруться значення сусідніх кольорів.

Метод *Specular Highlights* враховує те, що яскраві джерела світла (наприклад, сонце) створюють на гладких поверхнях точкові відображення. Використання ефекту *Specular Highlights* підсилює сприйняття реальності 3D-сцени, дозволяючи обчислювати інформацію про відбите від об'єктів світло та вносити відповідні корективи до 3D-сцени.

*Transformation and Lighting (T&L)* – це обчислення геометричних перетворень та освітленості. Трансформація – це той етап створення 3D-сцен, на якому 3D-координати вершин полігонів (трикутників) перетворюються в 2D-координати площини екрану. В традиційному для непрофесійних систем варіанті ці обчислення виконував ЦП. Цей ефект підтримується тільки високопродуктивними прискорювачами.

Функція *Bus mastering* (*управління шиною*) є необхідною для досягнення високої частоти кадрів. Існують два режими управління шиною: 1) графічний процесор запитує текстурні карти напряму з системної пам'яті, не звертаючися до ЦП; 2) графічний процесор обробляє команди синхронно з ЦП (інформація про полігони, з яких складається об'єкт, може бути обчислена ЦП та передана на обробку графічному процесору, а тим часом ЦП почне обчислення нового кадру). Ці режими суттєво підвищують продуктивність акселератору.

Примітка: не плутайте параметр *Frame Rate* (кількість кадрів, які відтворюються за секунду – *Frames per second, fps*) з частотою повторювання кадрів (вертикальною частотою), що вказує, скільки разів за секунду графічна карта передає зображення на монітор.

Можна ще довго перелічувати 3D-функції та технології, але: більшість з них зустрічається тільки в професійних графічних системах (ми їх торкатись не станемо); інші ефекти рідко підтримуються акселераторами, а тому їх перекладають на процесор.

3D-карти підходять до "віртуальної реальності" ближче, ніж їх 2D-попередники, але серйозно говорити про "інтерактивне відео" та "віртуальну реальність" ще рано. Сценарій ще покриває нальот штучності, але перевага ігрових приставок перед PC вже не така очевидна. Зображення предметів (у тому числі об'єктів, які пересуваються), дуже реалістичне.

Основною проблемою є природне зображення людей. Спочатку вони були грубо растрованими через малу кількість кольорів та низький дозвіл. З підвищенням продуктивності комп'ютерів та графічних карт, можливостей CD-ROM діючі особи ігор за допомогою методу *Motion Capture* з технологій виробництва відеофільмів стали відтворюватись живими акторами. Але більшість результатів були або низькоякісними, або повільними.

В області 3D ситуація змінюється. Типовий дозвіл може становити 1280×1024 (якщо карта має достатній обсяг пам'яті). При цьому "ривків" зображення практично не помітно. Але є інша проблема: фігури виглядають як роботи, оскільки для підвищення швидкості здійснюється спрощення зображень.

**Визначальні параметри 3D-прискорювачів та відеокарт.** 3D-графіка розвивається дуже стрімко: щопівріччя змінюються покоління графічних чіпів та відеокарт на більш продуктивні, якісні та реалістичні. Але створення по-справжньому реалістичної графіки стримується недостатньою продуктивністю графічних акселераторів. Тому поява продукції з вищою швидкістю та додатковими функціями знаходить споживачів ще на стадії проекту: для неї негайно починають розробляти застосування, розраховані на вищий рівень продуктивності.

Розвиток ринку настільної 3D-графіки визначають компанії *nVidia Corporation*, *ATI Technologies*, *Matrox Graphics*, *S3 Incorporated*, *Intel*, *BitBoys*, *Real3D*, *Trident*. Також потрібно відзначити фірми *Video Logic*, *TriTech*, *Artist Graphics*, *Fujitsu*, *GigaPixel*, *Oak tech*, *Philips*, *Silicon Reality*, *Stellar*, *3Dlabs*.

Компанія *nVidia* є найпродуктивнішою на новинки, задаючи такий темп, з яким важко змагатись: розробка повністю нової продукції щопівроку. Цю продукцію, що є швидкою та багатофункціональною, розраховано на різні сегменти ринку. Якщо в одному півріччі випускається акселератор на ринок десктопів та професійних *CAD*-станцій, то в наступному півріччі, модифікувавши його (підновивши, підвищивши частоту, покращивши архітектуру), випускається продукція на ринок мобільних систем. *nVidia* чітко розділила продукцію по сегментах ринку. При цьому користувач має великий вибір на будь-який смак. Забезпечені відлагоджені драйвери, робота в професійних пакетах моделювання та повна сумісність з ними. Є акселератори, розроблені в двох номінаціях: ігровий та професійний з відносно низькою ціною.

Повний опис продукції провідних фірм за її технічними характеристиками, переліком підтриманих 3D-функцій, характеристиками 2D-частини та підтримки відео, можна знайти в матеріалах Web-сайтів цих фірм. Ми ж акцентуємо увагу на найголовніших питаннях.

**Розглянемо методи та способи тестування 3D-прискорювачів.** Для порівняння відеоадаптерів та об'єктивного оцінювання їх продуктивності, здійснюють їх тестування на базі ряду програм, які пройшли певний контроль якості (графічні програми, спеціалізовані програми з тестування чіпів, комп'ютерні ігри). Безсумнівним лідером для тестування є графічні студії з редагування 3D-композицій об'єктів, такі як *3D Mark Max* та *True Space* фірми *Caligari*. В області спеціалізованого ПЗ можна відзначити *Winbench*. З ігрових програм найбільш популярними для тестування стали *Unreal Tournament (Epic MegaGames)*, *Hostile Waters*, *Expendable* й *Incoming (Rage SoftWare)*, *Croc*, *Alien Resurrection and Kanaan (Agronaut Software)*, *Quake Arena (id Software)*, *Half-Life (Valve)*, *Anachronox* та інші.

Поточні результати тестування популярних чіпів та акселераторів за різними параметрами та на різному тестовому полігоні можна побачити в матеріалах відповідних Web-вузлів. Найчастіше графічні чіпи тестують за такими характеристиками: частота чіпу, *MHz*; частота пам'яті, *MHz*; максимальний обсяг відеопам'яті, *Mb*; розрядність шини пам'яті, *bit*; підтримка *True Color* в 3D; частота *RAMDAC*, *MHz*; підтримка *3D API*; кількість конвексів рендерінгу; підтримка 3D-ефектів (трилінійна фільтрація, анізотропна фільтрація, мультитекстурування, *EMBM – Environment Mapped Bump Mapping*); карта, яку було використано при тестуванні; інтерфейс; *Video-in*; *Video-out*; орієнтовна ціна.

*Розглянемо характеристики 3D-прискорювачів очима користувачів.* Основний параметр прискорювачу – суб'єктивна якість зображення для майбутнього покупця з точки зору конкретних практичних застосувань. Цей параметр не піддається чисельній оцінці, а тому не може бути використаний для порівняння. Але він прямо залежить від числових параметрів, які наводять виробники в своїх специфікаціях (швидкість побудови зображення, глибина кольоропередавання, використані для побудови алгоритми та технології тощо).

Існує два основних кількісних показника швидкості роботи акселератору: *Fill Rate* – швидкість заповнювання елементів 3D-сцени пікселями з певною текстурою, яку вимірюють в текселях (мільйонах текселів) за секунду; *Triangle Rate* – максимальна пропускна здатність геометричної установки акселератору по виведенню на екран трикутників за секунду.

Важливу роль відіграє параметр *frames per second (fps)*, який показує, скільки разів за секунду апарат рендерінгу оновлює 3D-сцену на екрані. Чим вищим є значення *fps*, тим природнішими виглядають анімація та рух об'єктів: 30 *fps* є прийнятним показником у 3D-сценах; підвищення з 30 *fps* до 60 *fps* веде до суттєвого підсилення реалістичності відображеної 3D-графіки; у разі значень, які перевищують 75 *fps*, неможливо розрізнити подальше покращення якості (досягається межа, що обумовлена особливостями зору людини).

Швидкість побудови зображення оцінюють ще одним параметром – залежністю від потужності процесору ЕОМ. Якщо процесор буде недостатньо потужним для повного завантаження прискорювача процесом обробки трикутників, то значного підвищення швидкодії досягнуто не буде.



Глибина подання кольору, як правило, становить 16 бітів на піксель (*Hi-color*), 24 або 32 біти/піксель (*True-color*). Режим 16 бітів на піксель знижує вимоги до обсягу пам'яті буферу кадру прискорювача, але не здатен передавати кольори без помітних для ока викривлень. Для компенсування цих викривлень (наприклад, для усунення помітної дискретності передавання кольору), використовується технологія *Dithering* (растрування). Вона покращує передавання кольору за рахунок деякої втрати точності геометричної картини. Конкретна реалізація механізму дізерінгу суттєво впливає на якість зображення.

Не менш важливими є набір підтриманих прискорювачем режимів згладжування (інтерполяції та фільтрації текстур, згладжування країв полігонів) та *trip-map* режимів (техніки вибору різних текстур для різних дистанцій спостереження), а також точність виконання геометричних обчислень.

Багато з недоліків зображень, які виникають через геометричні недоліки чіпів (наприклад, нестиківку трикутників, із яких побудоване зображення), можуть бути скореговані завдяки якісним драйверам. Прискорювач повинен підтримувати як мінімум один із двох найбільш розповсюджених API – *Microsoft DirectX* або *SGL OpenGL*.

До обсягу відеопам'яті, встановленої в 3D-акселераторах, не можна застосовувати стандартну формулу розрахунку: певну її частину використовують для додаткових цілей, наприклад, відводять під буфер текстур або кадрів, Z-буфер тощо.

У разі високоякісного монітору, велику роль у виборі відеокарти відіграє максимальна частота *RAMDAC*: від неї залежать граничні частоти кадрів у різних режимах та якість відеосигналу, що надсилають до монітору.

*Акцентуємо увагу на 3D-ефектах, які забезпечують чіпи та 3D-акселератори провідних компаній у першу чергу.* По-перше, це антиаліасінг, який прибирає сходинкову структуру ліній при їх візуалізації на моніторі. Цю технологію обов'язково підтримує продукція компанії *nVidia*. Для її здійснення потрібна значна швидкість графічного чіпу та велика ширина каналу пам'яті. Чим вищий дозвіл ви використаєте, тим непомітнішим буде аліасінг. У залежності від потрібної якості, для згладжування 3D-сцени знадобиться різна кількість ресурсів.

Компанія *nVidia* реалізує антиаліасінг так: спочатку – рендерінг зображення в більш високому дозволі, потім – зниження дозволу. Через недостатньо швидку пам'ять, якою споряджають акселератори *nVidia*, антиаліасінг суттєво впливає на продуктивність, особливо в разі високого дозволу, 32-бітового кольору та швидкості під 100 кадрів/с.

По-друге, це ефекти стискання текстур та об'ємних текстур, які побутові акселератори підтримують давно. Ефективні алгоритми для них з'явилися недавно. Одним із вузьких місць при використанні стискання текстур є оперативна пам'ять. Метод об'ємного текстуровання допомагає працювати з вокселями, при накладанні текстур нестандартної форми, при створенні деформованої геометрії. Оскільки об'ємна текстура займає більше місця в пам'яті, ніж пласка, критичною є підтримка спеціальних ефективних методів стискання об'ємних текстур.

По-третє, це ефекти *T&L*, *TCL* та інші. Для професійної техніки прийнято, що процесор займається роботою операційної системи, звуковий чіп – звуком, графічний чіп – графікою, геометричний чіп – геометрією тощо. Саме від профе-

сійних ЕОМ надійшли в світ побутових акселераторів апаратна трансформація та освітлення, націлені на розвантаження ЦП від роботи над геометрією. Оскільки використання в ПК такої ж кількості процесорів, що і в професійному, призводить до невиправданого збільшення вартості системи, то ядро *T&L* інтегрують до графічного процесору.

Першим кроком з підтримки *T&L* стало впровадження компаніями інтегрованих наборів логіки з підтримкою *T&L*. Сучасне покоління продуктів використовує складніше *T&L*-ядро, що правильніше називати *TCL*-ядром (*Transform, Clipping, Lighting*). Використовуючи *TCL*, можна візуалізувати на екрані ПЕОМ фотореалістичні моделі автомобілів, пішоходів, фантастичних космічних сюжетів тощо. Але наявні програмні продукти, в тому числі комп'ютерні ігри, не використовують потенціал *TCL* повністю.

### Питання для самоперевірки

1. Коротко охарактеризуйте основні стадії *3D*-конверсу.
2. Наведіть перелік *3D*-ефектів, реалізованих на апаратному рівні в більшості *3D*-прискорювачів.
3. Як тестують *3D*-прискорювачі та за якими визначальними характеристиками їх порівнюють користувачі?

### 6.3. Технології графічних робочих станцій

Присвяtimo розгляд продукції широко відомої фірми *SGI* (*Servers, Graphics, Insight*) або по-старому *Silicon Graphics*. Вона створює комп'ютерні системи з використанням новітніх нетрадиційних технологій, які дозволяють користувачу швидко побачити та інтуїтивно зрозуміти розв'язок найскладніших задач. Сервери та суперкомп'ютери *SGI* надають феноменальну обчислювальну потужність для розв'язування цих задач. Графічні станції *SGI* дозволяють бачити, маніпулювати та використовувати величезні обсяги інформації. В поєднанні з широким спектром глобального сервісу, користувач отримує закінчене рішення для дослідження та втілення будь-яких складних ідей. З продукцією від *SGI* значно швидшими стають прорив на ринок телекомунікацій, створення оригінального дизайну, отримання нових науково-технічних відкриттів.

Донедавна продукція *SGI* була в повній ізоляції. Ситуацію змінило сімейство робочих станцій *SGI Visual Workstation* на процесорах *Pentium*, під управлінням *Windows NT*. Маючи вартість *IBM*-сумісних ПК, вони виконують функції дуже потужних комп'ютерів. Ті візуальні ефекти, що на звичайному *Intel*-комп'ютері виглядали б дивом, для робочої станції *SGI Workstation* є звичайною справою. Розглянемо причини незрівняно вищої швидкості, торкнувшись питань архітектури ПК.

Відкрита архітектура, основана на загальній шині та роз'ємах розширення для підключення додаткових пристроїв, свого часу стала основою *IBM PC*. Надалі фірма *IBM* розвинула ідею загальної відкритої шини, доповнивши її можливістю

широкої кооперації виробників комплектуючих. Було створено новий сектор промисловості, куди націлилися виробники апаратного та програмного забезпечення.

Конкуренція між численними виробниками підсистем забезпечила головну перевагу – доступність, що виправдовувала недоліки архітектури: універсальність архітектури призвела до погіршення характеристик (оскільки тільки процесор та пам'ять знаходяться на одній системній шині, а інші пристрої обмінюються з ними інформацією через загальну магістраль, її пропускна здатність є вузьким місцем обробки інформації); оскільки більшість зовнішніх пристроїв має власну буферну пам'ять, яку не може запитати процесор або інші підсистеми, має місце непродуктивне копіювання великих обсягів інформації та нерациональне використання апаратних ресурсів (у разі програмування виводу графічних зображень на екран ПК, основний час витрачається не на обчислення, а на пересилання даних до графічної пам'яті).

Але розробники *IBM PC* і не планували обробляти величезні обсяги графічної інформації (наприклад, "живого" відео), що потрібні для виведення на екран. Процесор ПК не обробляв дані відразу по їх надходженні, оскільки мав справу з їх миттєвою копією, оригінал якої надійшов до комп'ютеру раніше. До цієї затримки додавався ще й час реакції ОС.

Перед розробниками *IBM*-сумісних ПК постали жорсткі вимоги: підвищуючи продуктивність, забезпечити двійкову сумісність з попередніми моделями, не набагато збільшуючи ціну, а поступово зменшуючи її. Суворе дотримання вимоги сумісності в архітектурі викликало додаткові проблеми. Оскільки програми мали справу не тільки з апаратурою процесору, а й з *BIOS* та напряму з підсистемами (в разі потреби

підвищити швидкодію), то при найменшій модифікації архітектури все ПЗ довелося б переписувати наново.

Розробники суперЕОМ могли заради продуктивності не піклуватися про двійкову сумісність та вартість. Але потреби масового ринку не дозволяли значно підвищувати ціну та вимагали чекати здешевлення електронних компонентів, особливо мікросхем пам'яті. Поява дешевих мікросхем *SDRAM* дозволила без особливих витрат забезпечити процесорам відповідні переваги. Це продемонструвала архітектура комп'ютеру *SGI O2*, в якій за допомогою *UMA* (*Unified Memory Architecture* – уніфікованої об'єднаної пам'яті) вдалося досягти компромісу між продуктивністю, можливостями та ціною. Загальне поле пам'яті склали з чотирьох банків модулів *SDRAM*, об'єднаних мультиплексором. А всі підсистеми отримали можливість доступу до цього поля пам'яті крізь високошвидкісні порти зі швидкістю до  $2,1 \text{ Гб/с}$  та використання загального поля пам'яті як буферу, позбувшись окремих тимчасових сховищ. Схематично це зображено на рис.6.1.



Рис.6.1. Архітектура *SGI O2*

Велику кількість операцій пересилання даних від підсистем та до них було зведено до змінювання значення вказівника на масив з інформацією, що виконувалося за лічені такти. Сама інформація нікуди не пересувалася. Коли необхідність в деякому блоці пам'яті загального сховища відпадала, пам'ять віддавалася під інші потреби. Процесор міг завжди оперувати даними, які тільки-но надійшли до пам'яті від підсистем, не очікуючи їх пересилання. Стали непотрібними дорогі мікросхеми пам'яті, які встановлювали на графічних платах. У підсумку було створено робочу станцію, в якій процесор розвивав максимальну продуктивність, не гальмуючи через нестачу даних.

Ефективність системи відображення графічної інформації в архітектурі *O2* набагато вища, ніж у звичайних ПК. А використані в ній процесори *R 10000* та *R 5000* фірми *MIPS* можуть не тільки паралельно виконувати операції з цілими та дійсними числами, а й допомагати в побудові геометричних примітивів. Наприклад, в *R 5000* передбачено інструкції для переобчислення освітлення об'єктів та різних трансформацій зображення, що прискорюють побудову *3D*-сцен.

Підсистемою прискорення обслуговування аудіо- та відеопотоків є блок обробки та стискання зображень, що компресує в реальному часі відеоінформацію в формат *JPEG*. При підтримці процесору введення/виведення, він дозволяє обробляти одразу два відеосигнали. Наявність спеціального обчислювача для рендерінгу настільки розвантажує ЦП, що можна використовувати один із відеосигналів як "живу" текстуру поверхонь при *3D*-модельованні. Поява та успіх відносно дешевих комп'ютерів відкрили новий ринок доступних робочих станцій для обробки відеоданих.

Логічним продовженням цього процесу та результатом швидкого вторгнення ПК в області панування *SGI* стало створення *SGI*-установок на базі більш масових процесорів *x86* (починаючи з першого *SGI*-комп'ютеру, всі вони використовували виключно *RISC*-процесори фірми *MIPS Computers*).

Інтенсивне впровадження фірмою *Intel* технології *Visual Computing* на звичайних ПК стимулювало *SGI* на ліцензування в *Intel* низки архітектурних рішень та створення *IVC* (*Integrated Visual Computing*) – інтегрованої архітектури для візуальних обчислень. Її було реалізовано в сімействі робочих станцій *SGI Workstation 320, 540* як нащадка архітектури, розглянутої вище (було використано ідею об'єднаної пам'яті *UMA*). Схематично основні концепції цієї архітектури зображено на рис.6.2.

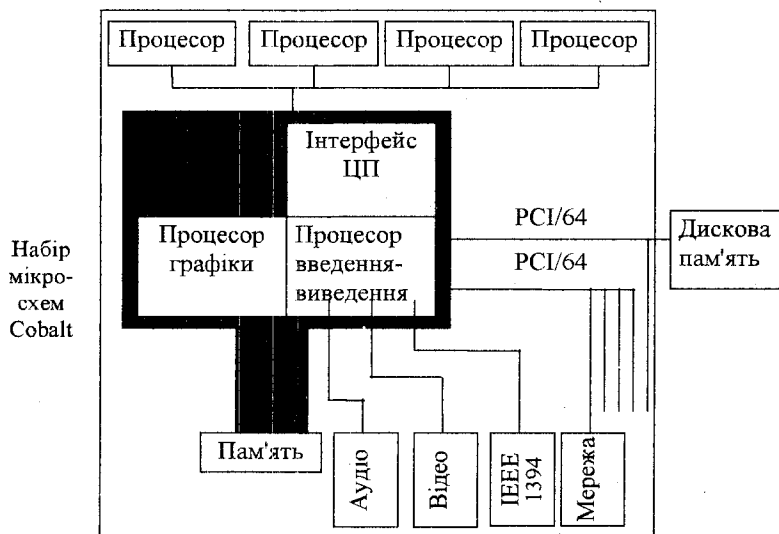


Рис.6.2. Архітектура *SGI Visual Workstation 540*, яка може містити до чотирьох процесорів



Для забезпечення загального доступу підсистем до пам'яті на основі напівзамовлених мікросхем було розроблено системний набір із інтегральних схем *Cobalt*, *Arsenic* та *Litium*, під загальною назвою *Cobalt*.

У кристалі під назвою *Cobalt*, який за кількістю транзисторів не поступається *Pentium*, розташований пристрій *Cobalt Memory Controller*. Він організує та синхронізує роботу численних підсистем комп'ютеру та центрального процесору з загальним полем пам'яті. Цим пристроєм керують операційна система та драйвери *OpenGL*. В ОЗП використовуються звичайні мікросхеми динамічної пам'яті з часом доступу 50 нс, а для оптимізації продуктивності їх конструктивно об'єднано по шість на одному модулі так, як у блокнотних комп'ютерах.

У кристалі *Cobalt* інтегровано апаратні прискорювачі побудови геометричних примітивів, 2D-зображень та накладання текстур.

Розташування на одному кристалі графічного прискорювача з керуючими ОЗП схемами дозволяє виконувати операції з графічними об'єктами максимально ефективно. 256-розрядна шина, що зв'язує процесор графіки з загальним полем пам'яті, може пропускати до 3,2 Гб/с (у 6 разів більше за *AGP*).

Кристал *Arsenic* відповідає за виведення на екран. У інтегральній схемі, що керує виведенням, реалізовано різноманітні можливості, наприклад: можна використовувати аналоговий дисплей з традиційним *VGA*-інтерфейсом (у моделі 540 є цифровий вихід для спеціально розробленого плоскопанельного монітору) або порт *IEEE 1394* із пропускнуою здатністю 400 Mb/s; передбачено канал *Ultra DMA/33* для підключення

CD- або DVD-приводів та традиційний інтерфейс для дисководу на гнучких дисках; є все необхідне і для введення, і для оцифрування сигналу з мікрофону та стереолінії, а також виведення в неї, на колонки та навушники. На стандартних мікросхемах фірми *Intel* реалізовано два *USB*, послідовний *RS-232* та паралельний *IEEE 1284* порти.

Кристал *Litium* обробляє вхідні та вихідні дані. Він має гарну здатність отримувати та віддавати інформацію. Можна одночасно вводити два нестиснутих потоки відеоданих у форматах *PAL* та *NTSC* та стільки ж виводити. Секрет такої швидкості полягає в тому, що всі основні шляхи руху інформації пов'язані індивідуальною магістраллю передавання даних із однією або декількома шинами *PCI* (їх у *SGI Workstation* – дві) та не залежать один від одного. Також апаратно реалізований алгоритм стискання відеоданих (можлива і програмна компресія).

За процесори можна брати різні покоління *Pentium*. Підтримуються процесори високої частоти. Будучи встановленими в *SGI Workstation*, вони продемонстрували гарні результати з обробки великих обсягів інформації, обумовлені високими піковими пропускними здатностями каналів передавання аналогового та цифрового відеосигналу в пам'ять та з пам'яті до монітору, а також наявністю двох 64-розрядних шин *PCI*. Швидкість обміну даними між процесором та ОЗП визначається процесором.

*SGI Workstation* працює під управлінням модифікованої ОС *Windows NT*. Вона здатна робити те, чого звичайна версія для ПК не може, наприклад, підтримувати інтерфейс *FireWire IEEE 1394*. Усі програми, що працюють під звичайною *NT*,

можуть функціонувати з модифікованою версією, але для повноцінного її використання вони повинні бути відповідно оптимізовані.

Серед *Unix*-комп'ютерів є кращі за швидкісними якостями, ніж *SGL Workstation*. Але незрівняною перевагою останніх є співвідношення "ціна – продуктивність": переробити ПК у графічну робочу станцію, що має подібні характеристики, дорожче, ніж придбати станцію *SGL*.

*SGL Visual Workstation* є новим підходом до вирішення проблеми обробки великих обсягів інформації в реальному часі та новим рівнем інтеграції всіх підсистем: весь базовий набір функцій реалізовано на одній платі, що підвищує надійність та швидкість їх виконання.

Але в разі неполадок доведеться замінити всю плату. Можливості необмеженої модернізації, що є звичайними для прихильників ПК, у *SGL Visual Workstation* обмежені. Замість плати розширення, є дві швидкісні шини *PCI* та шість роз'ємів, але спроби замінити діючі підсистеми найчастіше призводять до погіршення характеристик.

Архітектура *SGL Visual Workstation* виявила альтернативний шлях розвитку традиційної відкритої архітектури ПК. *SGL* планує продовжувати, паралельно з вдосконалюванням власного процесору *MIPS*, розвиток *Linux*. Ці тенденції є позитивними кроками до співпраці архітектурних шкіл.

Відзначимо, що матеріал цієї глави є авторським оглядом, аналізом та узагальненням численних відомостей про графічні чіпи, 3D-відеокарти та графічні робочі станції, що систематично наводяться на *Web*-сайтах відповідних фірм-виробників та в комп'ютерному журналі "*Chip*".

## Питання та завдання для самоперевірки

1. Виконайте аналіз переваг, недоліків та галузей застосування 3D-акселераторів та професійних графічних станцій.

2. Складіть та подайте в електронній формі реферативний огляд по сучасним моделям та технологіям 3D-відеоакселерації, використовуючи комп'ютерну періодику за останнє півріччя.

3. Доповніть реферативний огляд, який було виконано в попередньому завданні, наочними зображеннями типових представників відповідного апаратного забезпечення. Для цього використовуйте різноманітні можливості: здійсніть сканування готових зображень; зробіть фотографічні знімки та відеоролик за допомогою цифрових фотокамери та відеокамери; виконайте оцифровування традиційного фото та відео, помістивши їх надалі на відповідні носії інформації.

## ГЛАВА 7. ТЕХНОЛОГІЇ МУЛЬТИМЕДІА, ГІПЕРМЕДІА, ЦИФРОВОГО ФОТО ТА ІНТЕРАКТИВНОГО ВІДЕО

### 7.1. Визначення мультимедіа та гіпермедіа

Концепція мультимедіа здійснила революційні зміни в сферах освіти, комп'ютерного тренінгу, професійної діяльності, науки, мистецтва, комп'ютерних ігор та розваг [137-139].

*Мультимедіа-системи* – це інтерактивні системи, що забезпечують роботу з текстом, мовою, високоякісним звуком, статичними графічними зображеннями (технічними рисунками, художніми малюнками, фото тощо), анімованою комп'ютерною графікою, динамічним "живим" відео [28].

Мультимедіа-ПК нагадує побутовий стереофонічний *Hi-Fi* комплекс, об'єднаний з дисплеєм-телевізором та вкомплектований активними стереофонічними колонками, мікрофоном, дисководом для оптичних компакт-дисків, звуковою картою (для прослуховування чистих стереофонічних звуків крізь активні акустичні колонки) [47].

*Основною технічною проблемою мультимедіа є сумісна обробка різномірних даних (цифрових та аналогових, динамічного відео та статичних зображень тощо):* в комп'ютері дані зберігаються в цифровій формі, але теле-, відео- та більшість аудіоапаратури працює з аналоговим сигналом.

Оскільки вихідні пристрої комп'ютеру (монітори, динаміки) мають аналоговий вихід, то найпростішою та найекономічнішою технологією побудови перших мультимедіа-

систем стала: стиковка різnorodної апаратури з комп'ютером та надання комп'ютеру можливостей керування цими пристроями, суміщення вихідних сигналів комп'ютеру та відео- й аудіо-пристроїв, забезпечення їх нормальної сумісної роботи.

Подальше вдосконалювання мультимедіа-систем здійснюється в напрямку об'єднання різnorodних типів даних у цифровій формі на одному середовищі-носії, в межах однієї системи.

Повноцінне спорядження мультимедіа-ПК передбачає підключення численних зовнішніх пристроїв: аудіо- та відео-адаптерів, теле- та радіотюнерів, дисководів *CD-ROM*, джойстиків, клавіатури *MIDI* тощо.

Вони обслуговуються безліччю програмних утиліт-драйверів та нерідко конфліктують між собою [133]. Тому провідні розробники ПК об'єднали зусилля в створенні стандарту *P'nP* (*Plug and Play*, вмикай та грай), який передбачає обширний комплекс програмних та апаратних засобів для автоматичного настроювання конфігурації комп'ютеру відповідно до використаного з ним обладнання.

Технологія *P'nP* передбачає, що після вмикання комп'ютеру всі апаратні та програмні засоби автоматично оптимально налагодяться та почнуть працювати без збоїв та конфліктів [47].

**Гіпермедіа** – це поєднання технологій мультимедіа, гіперзв'язків, *World Wide Web* [63].

Передавання тексту, звуку, статичної графіки, анімації та відео в *WWW*, телефонні розмови та відеоконференції крізь *Internet* – це реальність, для здійснення якої повинні бути виконані такі процедури:

– оцифрування (*digitizing*) звуку та відео, яке виконують звукова карта (*sound card*) та плата для введення відео (*video capture board*);

– кодування (*encoding*) файлів із оцифрованим звуком та відео за допомогою алгоритмів стискання аудіо- та відеоінформації з припустимими втратами для скорочення потоку інформації (чим більше стискання, тим більші втрати);

– вбудовування (*embedding*) отриманої мультимедіаінформації до *Web*-сторінки;

– інтеграція (*integrating*) мультимедіа-файлів на *Web*-сервері, який повинен розпізнавати різні формати цих файлів.

Щоб отримати мультимедіаінформацію в *WWW*, потрібно виконати: інсталяцію (*installing*) медіаплеєру до *Web*-браузера; запит (*requesting*).

## Питання для самоперевірки

1. Сформулюйте поняття мультимедіа та гіпермедіа.
2. Назвіть галузі застосування мультимедіа та гіпермедіа в комп'ютерній графіці.
3. У чому полягає основна технічна проблема мультимедіа та як її вирішують?
4. Розшифруйте скорочення *P'n'P*.
5. Які технології застосовують для передавання та приймання мультимедійної інформації в *WWW*?

## 7.2. Відео- та аудіо-технології мультимедіа

**Відеотехнології мультимедіа.** У процесі змішування сигналів основні проблеми виникають з відеозображенням. Застосування різних *TV*-стандартів (*NTSC*, *PAL*, *SECAM*), моніторів та відеоконтролерів вимагає різних підходів до вирішення цих проблем. Але в будь-якому випадку потрібна синхронізація зображень. Пристрій *genlock* дозволяє сумістити на екрані монітору зображення, згенероване комп'ютером (текст, титри, анімована або нерухома графіка) та "живе" відео. Пристрій *encoder* (кодер) дозволяє перетворити комп'ютерне зображення в форму *TV*-сигналу та записати його на відеошлівку.

Настільні відеостудії, що є одним із прикладів застосування систем мультимедіа, дозволяють готувати суміщені відео-комп'ютерні кліпи, титри для відеофільмів та допомагають в процесі монтажу кінофільмів. Але вони не дозволяють обробляти та редагувати аналогове зображення.

Для того, щоб обробка та редагування аналогового зображення стали можливими, його необхідно оцифрувати та ввести до пам'яті комп'ютеру. Для цього призначені плати захоплення (*capture board*, *frame grabbers*).

Оцифрований кадр можна змінювати в процесі інтерактивної екранної обробки (в межах дозволу відеоадаптеру), наприклад, редагувати графічним редактором (видаляти або додавати деталі, змінювати кольори та масштаби, додавати спецефекти мозаїки та інверсії тощо). Оброблені кадри можна записати на диск у графічному форматі, а потім взяти за реалістичний нерухомий фон для комп'ютерної анімації.



Можлива покадрова обробка початкового зображення та виведення його назад на відеоплівку для створення псевдореалістичного мультфільму.

Оцифровування аналогових сигналів або породжує величезні масиви даних, або призводить до втрати якості зображення.

Запис послідовності кадрів у цифровому вигляді потребує великих обсягів зовнішньої пам'яті: оскільки частота кадрів у *NTSC* – 30 кадрів/с, а в *PAL/SECAM* – 24 кадри/с, то для запам'ятовування 1 с повнокольорового екранного відео потрібно 20-30 *Mb*. До того ж послідовність кадрів потрібно вивести на екран у відповідному темпі – близько 30 *Mb/s*. А такої швидкості передавання інформації не має жодний з існуючих зовнішніх запам'ятовуючих пристроїв. Щоб виводити на екран комп'ютеру оцифроване відео, доводиться йти на зменшення обсягу переданих даних (виведення зменшеного зображення в невеликому вікні, зниження частоти кадрової розгортки до 10-15 кадрів/с, зменшення кількості біт/піксель), тобто – на погіршення якості зображення.

Більш радикально проблеми пам'яті та пропускну здатності розв'язують за допомогою методів стискання/розгортання даних, які дозволяють стискати інформацію перед записом на зовнішній пристрій, а потім зчитувати та розгортати в реальному режимі часу при виведенні на екран.

Для рухомих відеозображень існуючі адаптивні різничні алгоритми можуть стискати дані з коефіцієнтом 100:1-160:1, що дозволяє розмістити на *CD-ROM* біля години повноцінного озвученого відео. Робота цих алгоритмів основана на тому, що як правило наступний кадр відрізняється від попереднього ли-

ше декотрими деталями: обираючи деякий кадр за базовий, для наступних кадрів можна зберігати тільки відносні зміни. При значних змінах кадру (монтажному склеюванні, наїзді або панорамуванні камери) автоматично обирається новий базовий кадр. Для статичних зображень коефіцієнт стискування є нижчим – 20:1-30:1. Для аудіоданих застосовують свої методи компресії.

*Існують такі базові схеми стискування даних.*

**Асиметрична** – коли інформацію стискають в автономному режимі. Одну секунду відео стискають на протязі декількох секунд або хвилин потужними паралельними комп'ютерами та поміщають на зовнішній носій. На машинах користувачів встановлюють дешеві плати декодування, що відтворюють мультимедіа-інформацію в реальному часі. Це збільшує коефіцієнт стискування та покращує якість зображення, але користувача позбавлено можливості розробляти власні продукти мультимедіа.

**Симетрична** – коли стискування та розгортання здійснюють у реальному часі на машині користувача, що може виробляти власну комерційну продукцію, не виходячи з дому. При цій схемі падає якість зображення (з'являються "змазані" кольори, розфокусованість), але з розвитком технології ця проблема відходить.

**Змішана** – коли розробник готує, відлагоджує та випробовує мультимедіа-продукт на своїй машині з симетричною схемою. Цей напівфабрикат у стандартному форматі надсилається до фірми, де його піддають стискуванню на потужному комп'ютері з використанням досконаліших алгоритмів та поміщають на *CD-ROM*. Через проблеми симетричної схеми, іноді обирають змішану схему.

Багато фірм веде розробку алгоритмів стискання відео з більшим коефіцієнтом. В їх основу покладено різні адаптивні варіанти: *DCT (Discrete Cosine Transform)*, *DPCM (Differential Pulse Code Modulation)*, а також фрактальні методи. Алгоритми реалізують апаратно (в вигляді мікросхем), як "firmware" (програми, записаної до ПЗП) або чисто програмно.

Різничні алгоритми стискання можуть бути застосовані до комп'ютерної графіки, що дає можливість реалізувати на звичайних ПЕОМ покадровий запис рисованих мультфільмів великої тривалості. Їх можна зберігати на диску, а при відтворенні – зчитувати, розпаковувати та видавати на екран у реальному часі, забезпечуючи 24-30 кадрів/с, необхідні для плавного зображення.

♦

У разі використання спеціальних відеоадаптерів (відео-бластерів), мультимедіа-ПК стає центром побутової відеосистеми, що конкурує з найдосконалішим телевізором. Новітні відеоадаптери мають засоби зв'язку з джерелами телевізійних сигналів та вбудовані системи захоплення кадру (компресії/декомпресії відеосигналів) у реальному масштабі часу (миттєво). Відеоадаптери мають швидку відеопам'ять 2-4 Mb та спеціальні графічні прискорювачі-процесори, дозволяючи отримувати до 30-50 кадрів/с та виведення повноекранних зображень.

**Аудіотехнології мультимедіа.** Мультимедіа-ПК містить плату-аудіоадаптер (саундбластер): це дає стереофонічне звучання та можливість запису на зовнішні носії звукових сигналів.

Дискові накопичувачі ПК не підходять для запису звичайних (аналогових) звукових сигналів: вони розраховані на запис цифрових сигналів, які практично не викривляються при

передаванні лініями зв'язку. Аудіоадаптер має АЦП, який періодично визначає рівень звукового сигналу та перетворює цей відлік до цифрового коду, що записують на зовнішній носій як цифровий сигнал. Цифрові вибірки реального звукового сигналу зберігаються в пам'яті комп'ютеру, наприклад, як *WAV*-файли. Зчитаний з диску цифровий сигнал подається на ЦАП, який перетворює цифрові сигнали в аналогові, які після фільтрації можна підсилити та подати на акустичні колонки для відтворення.

Важливими параметрами аудіоадаптеру є: частота квантування звукових сигналів (показує, скільки разів/с беруть вибірки сигналу для перетворення в цифровий код); розрядність квантування (кількість кроків квантування, що вимірюється ступенем числа 2). Якщо казати про приблизну оцінку якості звучання та квантування, то маємо такі набори "частотний діапазон, вид сигналу, частота квантування": "400-3500 Hz, ледве розбірливе мовлення, 5,5 KHz"; "250-5500 Hz, мовлення середньої якості, 11,025 KHz"; "40-10000 Hz, якість звучання УКХ-приймача, 22,040 KHz"; "20-20000 Hz, звук високої якості, 44,100 KHz". Що стосується розрядності квантування, то 8-розрядні аудіоадаптери мають  $2^8 = 256$  ступенів, що явно недостатньо для високоякісного кодування звукових сигналів. Тому застосовують в основному 16-розрядні аудіоадаптери, що мають  $2^{16} = 65536$  ступенів квантування, як в аудіо-*CD*.

Інший спосіб відтворення звуку полягає в його синтезі: при надходженні до синтезатору керувальної інформації, на її основі формується вихідний сигнал. Аудіоадаптери синтезують музичні звуки двома методами: частотної модуляції (стандартний *FM*-синтез має середні звукові характеристики, тому

на картах встановлюють складні системи фільтрів проти звукових перешкод); хвильового синтезу, з вибором звуків із таблиці звуків *WT* (цей метод забезпечує натуральніше звучання).

Сутність технології *WT*-синтезу така: на звуковій карті встановлюють модуль ПЗП із "защитими" семплами (зразками звучання музичних інструментів), а *WT*-процесор за допомогою спеціальних алгоритмів за одним тоном інструменту відтворює всі інші звуки. Звукові карти дозволяють не тільки записувати довільні семпли, а й підвантажувати нові інструменти.

Керувальні команди для синтезу звуку можуть надходити до звукової карти не тільки від комп'ютеру, а й від іншого пристрою, наприклад, від *MIDI*-пристрою. *MIDI* визначає протокол передавання команд стандартним інтерфейсом. *MIDI*-повідомлення містить не запис музики, а посилення на ноти. Коли звукова карта отримує подібне повідомлення, воно розшифровується (які ноти й яких інструментів повинні звучати) та відпрацьовується на синтезаторі. Комп'ютер може через *MIDI* керувати різними "інтелектуальними" музичними інструментами з відповідним інтерфейсом. Для електронних синтезаторів звичайно вказують кількість тих інструментів, що звучать одночасно, та їх загальну кількість (як правило, в діапазоні від 20 до 32).

Важливою є програмна сумісність аудіоадаптеру з типовими звуковими платформами: *SoundBlaster*, *Roland*, *Adlib*, *Microsoft Sound System*, *Gravis Ultrasound* тощо.

Аудіоадаптер є складним технічним пристроєм, побудованим на основі використання останніх досягнень в аналоговій та цифровій аудіотехніці. Новітні звукові карти містять цифровий сигнальний процесор *DSP* або розширений сигнальний

процесор *ASP*. Вони використовують досконалі алгоритми для цифрової компресії та декомпресії звукових сигналів, розширення бази стереозвуку, створення луни та забезпечення об'ємного (квадрофонічного) звучання. Процесор *ASP* використовують в разі звичайних двоканальних стереофонічних запису та відтворення звуку. Його застосування не завантажує акустичні тракти мультимедіа-комп'ютерів.

**Мультимедіа-носії інформації.** Важливою проблемою мультимедіа є забезпечення адекватних засобів збереження та розповсюдження мультимедіа-інформації. Носії повинні: вміщувати великі обсяги різномірної інформації; дозволяти швидкий доступ до її компонентів та якісне їх відтворення; бути дешевими, компактними та надійними. Ця проблема отримала достойне вирішення з появою оптичних дисків різних типів.

У перших системах мультимедіа були використані аналогові диски – "відеодиски" діаметром 12 та 8 дюймів. Найвідоміші 12-дюймові диски відповідають стандарту *LV (Laser Vision)*, підтриманому компаніями *Sony, Philips, Pioneer*. Існують два базових способи запису інформації на лазерні диски.

Перший спосіб – *Constant Angular Velocity (CAV)* – з постійною кутовою швидкістю. Диск *CAV* вміщує на доріжці один відеокадр – сукупність двох напівкадрів, які містять парні та непарні рядки кадру: телевізор працює в інтерлейсному режимі, поперемінно висвічуючи парні та непарні рядки кожного кадру.

Диск *CAV* обертається з постійною швидкістю 30 об/с, забезпечуючи необхідні для *NTSC* 30 кадрів/с. Кожна сторона

диску вміщує 30 хвилин відео *NTSC* або 37 хвилин *PAL*. Кожен кадр має свій номер (адресу), за якою можливий прямий доступ до нього. Інформацію на дисках *CAV* можна розбивати на частини – до 80 на кожній зі сторін. Керувальну інформацію, номери кадрів та частин розташовують у "бланкових" (невидимих) частинах кадрів. Кадри можуть трактуватись як нерухомі зображення: після завершення зчитування доріжки, пристрій не переходить до наступної, а знову зчитує ту ж саму. Можливе програвання з різними швидкостями та в зворотному напрямку. Разом із зображенням записують дві звукові доріжки, що доступні при перегляді кадрів у режимі відео.

Другий спосіб – *Constant Linear Velocity (CLV)* – з постійною лінійною швидкістю. *CLV*-диски вміщують по одній годині відео на кожній зі сторін. Інтерактивні можливості дисків *CLV* є обмеженими, тому в системах мультимедіа вони використовуються рідко, частіше їх застосовують для запису фільмів.

Проміжний аналогово-цифровий формат лазерних дисків *LVRM*, або *AIV (Advanced Interactive Video)* – покращене інтерактивне відео) дозволяє сполучати на одному диску аналогове відео з цифровим звуком та даними.

Існують різні типи чисто цифрових дисків. Наприклад, *CD-ROM*, як і цифрові аудіо компакт-диски *CD-DA*, мають діаметр 5.25", містять 500-600 *Mb* інформації та є наймасовішим цифровим засобом доставки мультимедіа-інформації; є диски *WORM*; є такі диски, що піддаються стиранню тощо.

Наведемо перелік найвідоміших форматів цифрових дисків: *CD-Audio* – найстаріший формат *CD* (майже всі дисководи *CD-ROM* можуть програвати звукові компакт-диски); *CD-Inter-*

*active* – власний формат *Philips* для "інтерактивних", в основному, ігрових *CD* для домашніх програвачів; *CD-ROM/XA* – поєднує стиснуті дані та звук, а також змішаний режим (запис іде з чергуванням для рівного відтворення, є найкращим форматом для мультимедіа); *Mixed mode* – комбінація звуку в форматі *RedBook* та даних *CD-ROM* (перша доріжка повинна містити дані, за нею можуть слідувати доріжки *CD-Audio*); *CD-Plus* – подібний до режиму *Mixed mode*, але запобігає звертанням звукового програвача до доріжок з даними для уникнення пошкодження динаміків; *ISO-9660* – стандартний формат та структура каталогів для *CD-ROM*; *HFS* – формат даних для *Macintosh*; *Hybrid discs* – містить системи *HFS* та *ISO*; *Photo CD* (фірма *Kodak*) – призначений для запису фотографій високої якості (для відтворення необхідний пристрій *CD-ROM/XA* та *CD-Interactive*); *Video CD* – відеоінформація в форматі *MPEG* та звук (призначений для відтворення фільмів).

Звичайний *CD-ROM* містить близько 700 *Mb*, а нові типи дисків мають на порядок більший обсяг та припускають запис інформації користувачем.

На відміну від жорстких магнітних дисків, *CD-ROM* можна замінити в лічені секунди. Програвачі комп'ютерних компакт-дисків (драйви *CD-ROM*) розділяють на два класи: зовнішні (зі своїм корпусом); внутрішні (вбудовані до системного блоку комп'ютеру).



## Питання для самоперевірки

1. Назвіть найвідоміші технології мультимедіа.
2. Які вимоги висувають до мультимедіа-носіїв інформації?
3. Що розуміють під терміном *MIDI*?

### 7.3. Технології гіпермедіа в комп'ютерній графіці

**Передавання звуку (*streaming audio*).** Спочатку для прослуховування звукового файлу, вказаного на *Web*-сторінці, використовували технологію "*download-and-play*": мультимедіа-файл починали програвати після того, як завантажували його цілком.

Технологія "*continuous-delivery*" дозволяє програвати файл одразу після отримання його початку. Її було реалізовано в програмах *InternetWave*, *RealAudio*, *StreamWorks*, *ToolVox for the Web*, *TrueSpeech Internet* та інших, які підтримують функції магнітофону (швидкий пошук, перемотування, пауза, стоп, регулятор гучності тощо). Розробники цих пакетів розв'язали складні задачі, головною з яких є обмежена швидкість та великі обсяги передавання інформації мережею. Вони стискають звук, забезпечуючи потрібну швидкість передавання, за допомогою алгоритмів стискання з втратами (на слух ці втрати є прийнятними) [38].

Пакети для передавання звуку мають два безкоштовних компоненти: кодер – для стискування; плеєр – для відтворення. Кодери стискають звукові файли форматів *WAV*, *AU*, *AIFF*, які мають заголовок з інформацією про частоту дискретизації, розрядність та спосіб оцифрування, кількість каналів (моно, стерео), довжину файлу. Протокол *UDP* є ефективнішим для передавання звуку, ніж *TCP*: в останньому нема корекції помилок, викривлені або втрачені пакети ігноруються. Проте при використанні *UDP* можуть виникнути проблеми з корпоративними брандмауерами (*firewalls*, які розраховано на *TCP*). Усі плеєри інсталиуються до *Web*-броузеру та добре працюють з найвідомішими броузерами *Microsoft Internet Explorer*, *Netscape Navigator*, *Quarterdeck Qmosaic*, *NetManage WebSurfer* та броузером служби *CompuServe*.

**Передавання відео (*streaming video*).** Потік інформації при передаванні відео є значно більшим, ніж при передаванні звуку. Навіть при застосуванні алгоритмів стискування з втратами цей потік до потрібної швидкості *bps* не стиснути. Технології *streaming video* надають два способи вирішення проблеми: передавати менше, ніж 24 кадри/с, що нагадує демонстрування слайдів; стискати потік сильніше на шкоду якості, оскільки зображення йде неперервно, але якість його погіршується.

Програми, що базуються на технологіях *download-and-play*, тобто завантажуй файл повністю та тільки потім дивись та слухай (приклад – стандарт *QuickTime*), дозволяють вирішити проблему в тому разі, якщо потрібні і гарна якість, і 24 кадри/с.

*Netscape Communications* ліцензувала технології *QuickTime* та *QuickTime VR* – крос-платформові (*Macintosh System*, *OS/2*, *Windows*, *UNIX*) технології зберігання, редагування та перегля-

ду синхронного відео, звуку, графіки та тексту компанії *Apple*. Ці технології інтегровано до броузера *Netscape Navigator*, який дозволяє вбудовувати до застосувань для *Internet (applets)* звукові та відеоефекти, програвати сценарії *QuickTime* та керувати ними. Цей броузер дозволяє дивитись анімацію та відеоролики, одночасно програвачи *MIDI*-файли в фоновому режимі. Як *Netscape Navigator*, так і *Microsoft Internet Explorer* підтримують відеоформат *AVI* та аудіоформати *WAV*, *AU*, *AIFF*, *MIDI*. *Internet Explorer* також може працювати з аудіо- та відеоформатами *MPEG*.

Коротко оглянемо відомі алгоритми стискання відео.

Алгоритм *JPEG* стискає з втратами окремі зображення, з можливістю зміни ступеня стискання. Цей алгоритм застосовують для компресії нерухомих зображень. Він забезпечує високу якість при стисканні 10:1. *JPEG* адаптовано і для відео: *MJPEG (Motion JPEG)* стискає кожен кадр відеопослідовності за методом *JPEG*. У підсумку якість зображення є гарною, але потік інформації є дуже великим.

Алгоритм *MPEG* стискає цифрове відео приблизно в 150 разів. До цифрового потоку додається синхронізований стереозвук. Відео майже дорівнює за якістю *VHS* (стандарту побутової відеотехніки). У *MPEG*-стисканні використовують алгоритм *JPEG*, але в процесі формування потоку даних передбачається, що два сусідніх кадри відеопослідовності мало відрізняються. Тому в потоці даних, стиснутих за технологією *MPEG*, є три типи зображень: *Intra (I-frame, опорні)* – *JPEG*-стиснуті зображення, що використовують для відновлення інших зображень (в типовому випадку їх задають два рази/с зі стисканням 12:1); *Predicted (P-frame, передбачувані)*,

що містять відмінності поточного зображення від попереднього (*Intra* або *Predicted*) та передбачають високий рівень компресії; *Bidirectional* (*B-frame*, двонаправлені), що містять відмінності поточного зображення як від попереднього, так і від наступного (при відновленні результат усереднюється) та передбачають максимальний рівень компресії. Якщо не вистачає пропускної здатності, кадри *Predicted* та *Bidirectional* опускають.

**Телефонні розмови по *Internet*.** За наявності ПК, модему та повнодуплексної (*full duplex*) звукової плати, що дозволяє здійснювати одночасно запис/відтворення, можна розмовляти з абонентом іншої країни вільно за вартістю *on-line* (швидкість з'єднання з *Internet* повинна бути не менше 9600 *bps*). Можна обійтися звичайною напівдуплексною звуковою картою та спілкуватися зі співрозмовником по черзі. Також потрібні мікрофон та динамік або мікротелефонна гарнітура. Проект *FreeWorld Dial-Up* компанії *Netscape Communications* націлений на те, щоб користувач *Internet* вів розмови з абонентами загальнодоступної телефонної мережі.

Програми, що дозволяють вести телефонні розмови крізь *Internet*, називають *Internet*-телефонами (*IP*-телефонами, *Web*-телефонами). Ці програми стискають мову та передають її через *Internet* у масштабі реального часу. Є можливість настроїти чутливість мікрофону та гучність динаміку, створити "телефонну книгу" з *IP*- або *E-mail*-адресами. *Internet*-телефони дозволяють передавати файли, обмінюватися текстовими повідомленнями (*chat*), рисувати на загальному планшеті (*whiteboard*) [100]. Наприклад, програма *WebPhone* підтримує

селекторні ради з чотирма співрозмовниками, шифрування мовлення, голосову пошту та автовідповідач тощо.

Зв'язок зі співрозмовником організують одним із двох способів: через *IRC*-сервер телефонного зв'язку; напямку за *IP*-та *E-mail* адресами (останній спосіб не завжди є можливим, оскільки деякі *Internet*-провайдери присвоюють при кожному підключенні нову *IP*-адресу). Завдяки стандарту *ITU H.323* на низькошвидкісні аудіо- та відеоконференції, *Web*-телефони мають можливість бути сумісними та працювати один з одним.

*Netscape Communications* включила до броузеру *Netscape Navigator* засіб *CoolTalk* фірми *InSoft* та вбудувала технології цієї фірми до своєї платформи *LiveMedia*, що підтримує роботу з відео- та аудіоінформацією в *Internet* у реальному часі. В броузері *Microsoft Internet Explorer* також є можливості *Internet*-телефонії.

Є програми, що дозволяють обмінюватися голосовими повідомленнями через *Internet* або корпоративні мережі *intranet*, наприклад, *CoolWare* (*CoolWare Technology*).

*Internet*-телефони можна розцінювати як *Internet collaborative tools* (засоби для сумісної роботи в *Internet*), в яких основним є планшет (*whiteboard*) – графічний редактор, який дозволяє працювати над рисунком декільком співрозмовникам, виконуючи такі основні операції: вставка файлу з текстового редактору або електронної таблиці; збереження в файлі змісту екрану та його друк; синхронізація екрану, яка дозволяє всім учасникам бачити одне й те ж місце на планшеті; стискання зображення, яке передають, що прискорює обмін даними між учасниками обговорення; використання вказівника. Деякі засоби для сумісної роботи в *Internet* надають такі можливості:

сумісне використання застосування (*Application Sharing*), наприклад, *Microsoft Word*; обмін текстовими повідомленнями (*Chat*); звуковий зв'язок (*Audio Conferencing*); передавання файлів (*File Transfer*). Усе це реалізоване в *Microsoft NetMeeting*; менші можливості мають *CoolTalk*, *Internet Phone*, *PowWow*. *Microsoft NetMeeting* – перший продукт, який став підтримувати стандарт *ITU T.120* на обмін даними між декількома користувачами, що працюють сумісно (*multiuser data conferencing*). *T.120* регламентує роботу планшету, передавання двійкових файлів, звуку, зображення.

**Відеоконференції в Internet.** Для того, щоб проводити *Internet*-відеоконференції, потрібні: модем, адаптер цифрової лінії, повнодуплексна звукова плата, мікротелефонна гарнітура, відеокамера, плата введення відео (*video capture card*), ПЗ. Наведемо приклади такого ПЗ: *Cine Video/Direct (CineCom)*, *Connectix VideoPhone (Connectix)*, *CoolView (InSoft)*, *Enhanced CU-SeeMe (White Pine Software)*, *VDOPhone Internet (VDOnet)* тощо.

Доступ до *Internet* повинен бути здійснений крізь модемне з'єднання *28800 bps*, а краще – крізь цифрову лінію *ISDN* (*128-256 Kbps*) або швидкісні лінії з *1544 Kbps* та вище (наприклад, *T1* або *E1*): відеоконференції через *Internet* за допомогою модему можливі, але якість їх є вельми посередньою, тобто *0,25-2,0 кадрів/с (frames per second, fps)* та часте пропадання звуку; з'єднання крізь *ISDN* дає *4-9 fps* та нормальний звук; з'єднання через швидкісні лінії дає *8-10 fps* із дозволом *160×120* пікселів та чистий звук.

Справжня відеоконференція – це коли кожен співрозмовник бачить та чує кожного іншого співрозмовника (*true*

*multipoint conferencing*). Можливі спрощені варіанти відео-конференцій: *point-to-multipoint* (усі бачать тачують тільки одного); *point-to-point* (спілкуються тільки двоє) [100]. У цілому ПЗ для відеоконференцій в *Internet* тестують за такими можливостями: обмін текстовими повідомленнями, передавання файлів, планшет.

### Питання для самоперевірки

1. Назвіть найвідоміші технології передавання звуку та відео.
2. Чому *Internet*-телефони характеризують як *Internet collaborative tools*?
3. Охарактеризуйте поняття *Internet*-відеоконференції.

## 7.4. Переваги та недоліки цифрових фотокамер

Цифрові технології професійної фото- та відеозйомки розвивають з 1980-х років. Побутові цифрові фотоапарати з'явилися на початку 1990-х років: це були чорно-білі пристрої, що за якістю поступалися плівочним фотоапаратам, а за ціною наближались до гарної кольорової відеокамери. З середини 1990-х років почалося масове виробництво любительських та напівпрофесійних цифрових фотокамер. Серед професіоналів стали користуватися популярністю цифрові приставки до традиційних фотоапаратів. А побутові цифрові камери перетво-

рилися в універсальні прилади для розв'язування великого переліку задач, зокрема в нетрадиційних для класичної фотографії галузях. У гонку цифрових технологій включилися всі провідні виробники фотообладнання. Ці технології розвиваються настільки стрімко, що цифрові фотокамери наздоганяють традиційні плівочні за якістю зображень та ціною. Цифрові фотокамери наполегливо завойовують загальний інтерес, але потрібного успіху ще не досягли: навіть найпростіша з них коштує дорого; до того, як цифрова фотокамера стане першим помічником фотографа, потрібно вирішити ряд проблем, основні з яких ми розглянемо нижче.

**Проблема якості зображення.** Студійний цифровий фотоапарат професійної якості коштує надто дорого для пересічного фотографа, не кажучи про любителя. В усіх інших випадках результати зйомки значно гірші за ті, що можна отримати за допомогою звичайної 35-міліметрової плівки: традиційні фотокамери, маючи сторічну історію вдосконалення технологічних процесів, гарно відтворюють неперервні кольори; цифрові камери є новачками та зв'язані дозволом зображень. Так, 35-міліметровий негатив можна без помітної втрати якості спроектувати на достатньо великий формат (залежно від його світлочутливості). На відміну від цього, більшість недорогих цифрових фотокамер надають зображення з таким малим дозволом, який годиться лише для перегляду зображень на моніторі.

**Проблема кількості фотознімків.** У процесі зйомки звичайним фотоапаратом, потрібно періодично замінити касету з відзнятою фотоплівкою. З цифровим фотоапаратом ситуація не краща: доводиться замінити дискети 3,5", які вміщують



30-40 кадрів розміру 640×480 у форматі *jpg* з кольоровою гамою 16 млн. кольорів. Якщо ж користуватися дорогим картою флеш-пам'яті, то зйомка закінчиться дуже швидко: в розпорядженні буде невелика кількість знімків, а для продовження зйомки після заповнення пам'яті необхідно підключити фотоапарат до комп'ютеру.

**Проблема заміни джерел живлення.** Однією з найпривабливіших властивостей цифрових фотокамер є відсутність необхідності постійно купляти нову плівку, але доведеться так само часто міняти батарейки.

**Переваги цифрових фотоапаратів.** Визначальною перевагою, завдяки якій цифрові фотоапарати успішно продають, є зручність у користуванні. Якщо важлива швидкість, то за допомогою цифрової фотокамери можна вже через декілька хвилин після зйомки кадру мати його в себе на моніторі.

Більшість цифрових фотокамер споряджено *LCD*-екранами (рідинно-кристалічними дисплеями – РКД), за допомогою яких можна: налаштуватися на зображення; переглядати відзняті кадри, викидаючи непридатні або нецікаві кадри та вивільняючи місце в пам'яті фотоапарату. Цифрові фото- та відеокамери, обладнані *LCD*-екранами, можна підключати до принтеру та відправляти знімки на друк, обминаючи комп'ютер: попередній перегляд відзнятого матеріалу виконують на *LCD*-екрані. Цифрові фотокамери – гарна допомога тим власникам *Web*-сторінок, які бажають запропонувати *online*-спільноті свій каталог фотографій: дозвіл відзнятих зображень ідеально підходить для *Internet*, а швидкість та простота зйомки й обробки зображень додають ще одну перевагу.

Цифрова камера не просто заміняє плівочну, а й виконує додаткові функції медіа-колектору, мультимедійного носія інформації: можна записувати звук та рухомі об'єкти, супроводжувати знімки анотаціями, записувати текст та пов'язувати його з зображеннями, тобто формувати мультимедіа-продукт (ведуться роботи над технологією отримання фотозображень, що за якістю переважають знімки на 35-міліметровій плівці). Оскільки цифрові фотокамери розраховані на декілька типів даних та їх комплексну обробку, в масового споживача змінюються уявлення про видавничу справу та носії інформації. Навіть у найдешевших моделях цифрових фотокамер є можливість керувати пам'яттю камери: переглядати, відбирати та видаляти непотрібні кадри.

Таким чином, цифрову фотокамеру можна охарактеризувати, як комп'ютер з проєкційним пристроєм. Цифрова камера є пристроєм, який керує пам'яттю та виведенням, записує звук та відео. До такої камери вбудовують процесор та операційну систему з ПЗ, наприклад, таким, що дозволить запрограмувати камеру на зйомку прискореного руху або буде з'єднувати окремі кадри в панораму, записувати анімацію, здійснювати монтаж та вставляти спецефекти. Такі компанії, як *Sharp* та *Nikon*, пропонують продукти, що являють собою комбінацію цифрових фотокамер та *PDA* – персональних цифрових секретарів.

## Питання для самоперевірки

1. Які переваги надають цифрові фотокамери?

2. З якими проблемами стикаються користувачі цифрових фотокамер?

3. Чому цифрову фотокамеру можна охарактеризувати як медіа-колектор?

## 7.5. Технології цифрового фото

Плівоchnу якість можна отримати тільки на деяких професійних сканувальних камерах та багатопрорідних апаратах високої вартості. Проте до них усе більше наближаються недорогі камери. В сфері дешевих фотокамер любительського рівня здійснюються значні зміни. Присвятимо ним подальший розгляд.

Перші цифрові камери, що зберігали зображення в електронній формі, захоплювали його за допомогою чутливих елементів *CCD* (прилад із зарядовим зв'язком – ПЗЗ). Цей монолітний чіп являє собою матрицю мікроскопічних сенсорів, здатних реагувати на рівень освітлення. Кожен сенсор породжує електричний заряд певної сили, який перетворюється в числове значення, що кодує колір та тон у заданій точці. Оскільки *CCD*-матриця породжує чорно-білі зображення, то для отримання 24-бітового кольору перед нею розміщають червоний, синій та зелений фільтри: кожного разу сенсор породжує один із основних кольорів, які накладаються, створюючи повнокольоровий піксель. Ці складні маніпуляції з даними є однією з причин невисокого дозволу цифрових фотокамер. У *CCD*-приладів є й інші недоліки: споживають багато енергії, вимагаючи великих батарей; їх ланцюги живлен-

ня відрізняються від активних та живильних ланцюгів іншої електроніки в цифровій камері, тобто комбінувати елемент *CCD* з іншими ланцюгами на одному кристалі кремнію складно. Таким чином, *CCD*-технологія вимагає застосування трьох різних рівнів напруги, великого часу та потужних джерел живлення.

На передові позиції за кількістю пікселів, яку може видавати недорогога фотокамера, виходить *CMOS*-технологія активних сенсорів: чутливим елементом є дуже мала мікросхема, вироблена за масовою технологією *CMOS APS*. Ця технологія, як і *CCD*, застосовує пласкі матриці світлочутливих елементів, які передають образ до пам'яті. Але розвиток напівпровідникових технологій дозволив створювати підсилювачі в кожному елементі зображення, генеруючи значні токи, що можна передавати крізь звичайні провідники. Лідером у виробництві *CMOS*-елементів є компанія *Photobit*.

*CMOS*-активні сенсори споживають набагато менше енергії, що є для цифрових камер великою перевагою. Але *CCD*-технологія забезпечує низький рівень шумів. Ось чому *CMOS*-камери певний час використовувалися там, де важливішою є не якість, а дешевизна. На даний час вдалося знизити рівень шумів у схемах *CMOS APS* до рівня *CCD*. До того ж технологія *APS* видаляє такі недоліки *CCD*, як погана чутливість в умовах слабкого освітлення та низьких температур, складність виробництва матриць великого обсягу. *CMOS*-технологію світлочутливих елементів вдосконалено настільки, що можна знімати сигнали з кожного окремого пікселя.

Першими про випуск для масового ринку цифрових камер за *CMOS*-технологіями оголосили корпорації *Toshiba* та *Vivitar*.

Застосування технології *CMOS* дозволяє: випускати чутливі елементи на підприємствах-виробниках чіпів; суміщати всі функції камери на одному чіпі та вбудовувати нові чіпи в усі ПК. Так, компанія *Creative Laboratories* продемонструвала систему відеоконференцій, в якій роль телекамери виконував чіп виробництва *VLSI Vision*. Відомими є також розробки компаній *Eastman Kodak*, *National Semiconductor*, *Polaroid*, *Hewlett-Packard*, *IBM*, *Intel*, *Omnivision Technologies*, *Suni Imaging Microsystems*, *Matsushita* (розвиває чіп, який суміщає світлочутливий елемент з електронікою стискання).

Але не всі обирають відеочіпи *CMOS*: представники *NEC* вважають, що *CCD*-пристрої більше відповідають запитам ринку, забезпечуючи незрівняно вищу якість, а корпорація *Sony* планує розширювати їх виробництво; фахівці дослідницького відділу *CMOS* компанії *Olympus Optical* вважають, що технологія *CCD* досягла розквіту та не дозволяє повністю від неї відмовитись, але що майбутнє – за технологією *CMOS*.

Цифрові камери фірми *Olympus* відрізняються від попередників вищим рівнем якості: об'єктиви невисокої якості з низькою світлосилою замінені класичним об'єктивом *TTL* з видошукачем крізь лінзу; камеру можна пригвинтити до штативу та відкорегувати видошукач відповідно до свого зору; цифрові камери суттєво легші за плівочні; дозвіл зображення в цій цінній категорії поза конкуренцією та дозволяє досягти гарної якості відбитків на сублимаційному принтері (10×14 см).

Серед дорожчих фотокамер вартістю до 1000 \$ з дозволом 640×480 та 800×600 потрібно відзначити моделі фірм *Agfa*, *Apple*, *Canon*, *Casio*, *Epson*, *Fuji*, *Kodak*, *Nikon*, *Olympus*, *Polaroid*, *Ricoh*, *Sanyo*, *Sony*, *Toshiba*, *Vivitar*, *SoundVision*, які характеризують за такими параметрами: комп'ютерними (ка-

мера, *min/max*, матриця, пам'ять, формат, кадри, звук, відеовихід, платформа), оптичними (оптика, діафрагма, витримка, спалах, фокус, *min/max*, видошукач, розмір, вага). Аналізуючи значення цих характеристик, можна обрати потрібну модель та створити узагальнений образ оптимальної цифрової камери. Розглянемо його.

Цифрова камера повинна мати всі можливості класичного фотоапарату. Цінують гарну оптику, трансфокатор з дво- та трикратним збільшенням, можливість перемикання фокусування з автоматичного режиму на ручний, макрозйомку. Для зручності зйомки камеру споряджають *LCD*-панеллю, що може обертатися. Камери мають вбудований спалах, який забезпечує до п'яти режимів роботи. Бажана функція автоспуску.

Чутливість цифрових фотокамер за стандартом *ISO*, як правило, за фотографічними мірками є низькою. Вони споряджені функціями автоматичного встановлювання балансу білого, виставляють час експозиції, регулюють витримку.

Зйомка може вестись як окремими кадрами, так і в неперервному режимі з частотою до 5-15 кадрів/с. Кожен кадр можна супроводжувати текстовими та звуковими коментарями, супутніми параметрами (дата та час зйомки, рівень компресії тощо). Для запису та прослуховування звуку є вбудовані мікрофони та динаміки.

Для оперативного перегляду зображень можна використати *LCD*-панель розміром 4-5 см. Кадри, що не сподобалися, можна видалити, звільняючи місце для подальшої зйомки. Передбачено різні режими перегляду: окремий кадр, декілька кадрів одночасно, слайд-шоу всіх кадрів. Камера може мати стандартний відеовихід для перегляду зображень на екрані телевізору, запису на відеомагнітофон або друку на принтері.

Перевагою цифрового фотоапарату є простота зв'язку з комп'ютером. Зображення переписують в одному зі стандартних графічних форматів крізь інтерфейсний кабель по паралельному або послідовному порту, а також крізь адаптер для змінних *PCMCIA*-карт пам'яті. Всі операції виконуються під управлінням доданої програми, що має можливості редагування зображень. Цифрові фотокамери дозволяють організувати відеоконференцію або презентацію під керуванням комп'ютеру. Серед додаткових функцій цифрових фотоапаратів – зйомка в режимі підвищеної контрастності з наступним розпізнаванням тексту.

Цифровий фотоапарат за зовнішнім виглядом, розмірами, вагою та зручністю використання не поступається аналоговому.

Фірми *Toshiba*, *Panasonic*, *Yashica*, *Sharp*, *Sony* та інші постійно пропонують ринку цифрових пристроїв новинки.

### Питання для самоперевірки

1. Порівняйте технології *CCD* та *CMOS APS*.
2. За якими параметрами оцінюють цифрові фотокамери?
3. Охарактеризуйте оптимальну цифрову фотокамеру.

## 7.6. Технології інтерактивного цифрового відео

Першим методом передавання відеосигналів став аналоговий, а одним із перших відеоформатів на його основі – композитний відеосигнал.

*Композитне аналогове відео* комбінує всі відеокомпоненти (яскравість, колір, синхронізацію та інші) в один сигнал, якість якого є далекою від досконалості: неточне передавання кольору, недостатньо "чисте" зображення тощо.

Для усунення цих недоліків було розроблено *компонентне аналогове відео*, в якому різні відеокомпоненти подано незалежними сигналами. Подальші вдосконалення цього формату привели до появи: *S-, RGB-, Y-, Pb-, Pr-Video* та інших.

Аналогові відеоформати мають суттєвий недолік – втрату якості при копіюванні відеоматеріалу: дубль поступається оригіналу за чіткістю, яскравістю та рядом інших характеристик.

Недоліки аналогового методу відтворення відео стали каталізаторами розробки цифрового відеоформату: кожна копія цифрового відео є ідентичною оригіналу.

Комп'ютерне цифрове відео являє собою послідовність цифрових зображень та пов'язаний з ними звук. Елементи відео зберігаються в цифровому форматі. В сфері професійного відео застосовують декілька цифрових відеоформатів: *D1, D2, Digital BetaCam* тощо. Склалося багато способів захоплення, зберігання та відтворення відео на комп'ютері. При цьому завдяки *ISO* вироблено єдині стандарти відеоданих.



Сучасний відеоряд базується на цифровій основі, але всі цифрові відеоформати використовують в якості носія початкового сигналу плівку з послідовним доступом. Тому більшості професіоналів сфери відео звичніше працювати з плівкою, а не з комп'ютером. Плівка як джерело даних залишається переважнішою за жорсткий диск комп'ютеру, оскільки може містити значно більший обсяг даних.

Використання комп'ютерів надає ряд суттєвих переваг для цифрового відеомонтажу: забезпечує прямий доступ до будь-якого відеофрагменту (при роботі з плівкою, дістатися до необхідних ділянок можна лише шляхом послідовного перегляду відеоматеріалу); пропонує широкі можливості обробки зображення (редагування, стиснення) [90].

Для розуміння специфіки сфери професійного відеомонтажу потрібно усвідомити, що визначення та характеристики професійного цифрового відео якості мовлення для телебачення суттєво відрізняються від того відео, що реалізують на ПК: на протязі десятиріч на телебаченні відпрацьовували професійні стандарти на високоякісне відео та жорсткі вимоги до нього, що привело до появи багатьох технологічних нововведень.

Базовими характеристиками цифрового відео є: частота кадру; екранний дозвіл; глибина кольору; якість зображення. Що стосується частоти кадру (*frame rate*), то стандартною швидкістю відтворення відеосигналу є 30 кадрів/с, для кіно – 24 кадри/с. Оскільки кадр складається з рядків, які прорисовують через один, отримують два напівкадри (поля), а кожна секунда аналогового відеосигналу складається з 60 полів (цей процес називають *interlaced*-відео).

Монітор ПК використовує для прорисовування екрану метод "прогресивного сканування" (*progressive scan*): рядки кадру формують послідовно, згори вниз, а повний кадр прорисовують 30 разів/с (цей метод називають *non-interlaced* відео).

Екранний дозвіл (*spatial resolution*) – це кількість точок, із яких складається зображення на екрані. Прямого зв'язку між дозволом аналогового відео та комп'ютерного дисплею (наприклад, звичайним для моніторів *PC* та *Macintosh* базовим дозволом 640×480) нема. Стандартний аналоговий відеосигнал надає повноекранне зображення без обмежень розміру, властивих комп'ютерному відео. Телевізійний стандарт *NTSC*, розроблений у США (використовується в Північній Америці та Японії), пропонує дозвіл 768×484. Стандарт *PAL*, розповсюджений в Європі, пропонує дозвіл 768×576. Оскільки дозволи аналогового та комп'ютерного відео різні, в процесі перетворення аналогового відео в цифровий формат доводиться масштабувати та зменшувати зображення, що призводить до певної втрати якості.

Глибина кольору (*color resolution*) – це комплексний показник, який визначає кількість тих кольорів, які одночасно відображаються на екрані. Комп'ютери обробляють колір у *RGB* (*red, green, blue*)-форматі. Відео використовує й інші методи. Однією з розповсюджених моделей колірності для відеоформатів є *YUV*. Кожна з моделей, *RGB* та *YUV*, може бути подана різними рівнями глибини кольору (максимальної кількості кольорів). Для кольорової моделі *RGB* є характерними режими глибини кольору 256 кольорів (8 біт/піксель), 65535 кольорів (16 біт/піксель), 16,7 млн. кольорів (24 біт/піксель). Для моделі *YUV* застосовують режими 7 біт/піксель

(4:1:1 або 4:2:2, приблизно 2 млн. кольорів) та 8 біт/піксель (4:4:4, приблизно 16 млн. кольорів).

Якість зображення (*image quality*) – це найважливіша характеристика. Вимоги до якості зображення залежать від конкретної задачі: іноді достатньо зображення розміром у 1/4 екрану з палітрою в 256 кольорів (8 біт/піксель) при швидкості відтворення 15 кадрів/с; іноді потрібне тільки повноекранне відео (768×576) з палітрою в 16,7 млн. кольорів (24 біт/піксель) та повною кадровою розгорткою (24 або 30 кадрів/с).

Без стискання: важко забезпечити неперервне передавання відео зі швидкістю 21 Mb/s, яка відповідає вимогам визнаного світового стандарту цифрового телебачення CCIR 601; обсяги та вартість зберігання відеоданих на дисках не дозволяють застосовувати PC для чорнового монтажу. Якість стискання варіює в широких межах. Звичайними для відеосистем є коефіцієнти стискання 4:1-100:1. Для цифрового обладнання нелінійного монтажу відео з якістю мовлення (4:1 та менший коефіцієнти стискання), вплив стискання особливо помітний.

Коефіцієнт стискання (цифрове висловлення співвідношення між обсягом стиснутого та початкового відеоматеріалу) є особливо важливим для професіоналів, які працюють з цифровим відео на комп'ютерах.

Чим вищим є коефіцієнт стискання, тим гіршою є якість відео, проте багато залежить від використаного алгоритму. Наприклад, для MPEG стандартом вважають коефіцієнт стискання 200:1 при непоганій якості відео. Варіанти Motion JPEG працюють з коефіцієнтами стискання 5:1-100:1, але в разі 20:1 вже важко отримати нормальну якість зображення. Окрім

алгоритму стискання (*MPEG* або *Motion JPEG*), якість відео залежить від параметрів цифрової відеоплати, конфігурації комп'ютеру, ПЗ.

Методи стискання даних використовують математичні алгоритми для видалення, групування та/або усереднення подібних даних, присутніх у відеосигналі. Існує велика кількість алгоритмів стискання: *PLV*, *Compact Video*, *Indeo*, *RTV*, *AVC*, *MJPEG*, *MPEG* тощо. *Motion JPEG (MJPEG)* та *MPEG* є визнаними міжнародними стандартами для стискання відео. Вибір алгоритму стискання залежить від поставленої мети.

Постійно розробляються досконаліші алгоритми стискання зображення та відеопотоку. Але незмінним залишається одне: чим вищий коефіцієнт стискання, тим гірша якість. Методи стискання зводяться до аналізу зображення, на основі якого роблять передбачення про зображення в цілому, а це припускає можливість похибки. Застосування інтегральних оцінок до різних зображень у процесі стискання дає різні результати: чудові – на зображеннях з плавними переходами та невеликими шумами, погіршені – на різких та зашумлених зображеннях.

Велика кількість апаратних та програмних рішень для оцифрування та монтажу відео ставить перед необхідністю вибору. Головним критерієм є дотримання загальноприйнятих стандартів: це не гарантує найкраще рішення, але забезпечує впевненість у завтрашньому дні.

Як правило, відеоплати побудовано на основі одного з двох методів компресії – *MJPEG* або *MPEG*, сфери застосування яких різні: технологія *MPEG*-кодування та монтажу до останнього часу була дорожчою та складнішою, ніж *MJPEG*; велику роль зіграло аносування специфікацій *MPEG*, який

стає основою нових відеотехнологій не тільки на комп'ютерах, а й на телебаченні та в кіно; *MPEG* у сукупності з компакт-дисками високої щільності *CD-MLR* та *CD-DVD* суттєво змінив звичний відеоринок.

Чим нижчі продуктивність системи та пропускна здатність шини, тим важливіша можливість контролю параметрів цифрового відео. Гарна система оцифровування та стискання відео дозволяє задавати найважливіші параметри для апаратної та програмної частини відеосистеми. Так, у певних застосуваннях вирішальну роль відіграє швидкість відтворення відео, але доводиться відмовлятися від повноекранного зображення. В інших випадках достатньо 15 кадрів/с, але якість кадрів повинна бути високою.

### Питання для самоперевірки

1. Дайте порівняльний аналіз аналогового відео та комп'ютерного цифрового відео.
2. Які характеристики цифрового відео є базовими?
3. Назвіть прогресивні методи компресії цифрового відео.

### 7.7. Галузі застосування *MPEG*-відео

У першу чергу з цифровим відео пов'язують можливість дивитися на комп'ютері повнометражні повноекранні відеофільми: використання ПК як відеомагнітофону та теле-

візору стає звичайним явищем, а асортимент *Video-CD* фільмів є великим та динамічно оновлюється. З кожним роком цифрове відео ширше застосовується в різних сферах життя: у відеобібліотеках, у навчанні, на телебаченні, в відеокіосках тощо [63, 90, 91].

*MPEG-1* за якісними параметрами багато в чому є аналогічним звичайному *VHS*-відео.

Його застосовують у першу чергу там, де незручно або непрактично використовувати стандартні аналогові відеоносії: у відеобібліотеках, відео за вимогою (*video on demand*), відео по телефону, навчанні, відеокіосках (інформаційних кіосках), корпоративних презентаціях.

Організації, що мають великі відеоархіви (музеї, бібліотеки, державні підприємства та наукові організації, рекламні фірми та інформаційні агенства), переходять на цифрове відео. Вони можуть суттєво виграти, перекодувавши відеоархіви до цифрового формату та помістивши їх на *CD*-носії або на спеціальний сервер: на відміну від аналогових носіїв, гарантується тривале зберігання, багаторазове програвання без втрати якості та швидкий доступ до будь-якого фрагменту; можна легко відкрити віддалений доступ до відеоматеріалу крізь локальну мережу (*intranet*) або *WWW (Internet)*.

Термін "відео за вимогою" з'явився недавно, а такий сервіс практикувався в дорогих готелях. На даний час здійснюється практична реалізація ідеї про інтерактивну цифрову телевізійну систему широкого вжитку, крізь яку користувач може зробити запит на будь-який фільм або передачу на певний час та безпосередньо додому.

Деякі телефонні компанії розробляють системи, що дозволяють замовляти відеофільми крізь звичайні телефонні лінії. Основною проблемою є обмежена пропускна здатність наявних телефонних комунікацій, але повсилюдне впровадження *ISDN* та інших нових технологій зв'язку допомагає вирішити її.

Інтенсивно розвивається ринок тренажерів та інтерактивних навчальних комплексів. *MPEG* став ідеальною альтернативою аналоговим відеосистемам та лазерним відеодискам, надаючи, при більш низьких витратах, ряд переваг: транспортабельність, компактність, легкість модернізації, можливість працювати в мережі.

Відеокіоски (інформаційні кіоски) надають можливість по-новому організувати та автоматизувати інформаційний сервіс у рамках організації, що важливо для роздрібних магазинів, автомобільних салонів та банків: відеокіоск завжди в розпорядженні клієнтів, коли продавець не може приділити достатньо уваги одразу декільком клієнтам, детально розповісти про всі особливості та специфікації товару, наочно та ефективно його продемонструвати.

У відеокіоску можна розташувати не тільки детальну інформаційну базу про продукти та послуги, а й інтерактивні відеофільми, що дозволять просто та наочно відповісти на багато питань (про асортимент продукції, її особливості та характеристики тощо). Інформація, яку раніше виводили в вигляді слайдів та супроводжувального тексту, завдяки впровадженню повноекранного відео стає доступнішою та ефектнішою. Використовуючи *MPEG*, розробник може регулярно та без особливих витрат оновлювати зміст відеокіоску.

Оскільки корпоративний ринок стає вимогливішим до якості та технічних можливостей презентаційного обладнання, більшість презентаційного ПЗ підтримує роботу з відео. Багато менеджерів впевнилися, що не можна недооцінювати можливостей мультимедіа-систем: без ефектного супроводження та інтерактивних ілюстрацій ваші ідеї, добре написаний та аргументований проект або доклад можуть залишитися незапитаними.

*Специфікації MPEG, починаючи з MPEG-2, передбачають використання високих дозволів для досягнення максимальної якості зображення. Тому цей формат у першу чергу застосовують у професійних сферах, таких як кабельне телебачення (CATV), направлене супутникове мовлення (DBS), телебачення високої чіткості (HDTV).*

Що стосується кабельного телебачення, то переведення телебачення на цифрове мовлення надає користувачу реальну можливість приймати телепередачі зі стерео- та *Dolby Surround*-звуком. Що стосується направленого супутникового мовлення, компанія *Thomson* виробляє на замовлення консорціуму *Hughes/USSB* спеціальні декодери для приймання до 150 каналів. Що стосується телебачення високої чіткості, то створено консорціум *U.S. Grand Alliance*, що займається новим стандартом *HDTV* для телебачення високого дозволу з підтримкою режимів  $1440 \times 960$  (30 Гц),  $1280 \times 720$  (60 Гц).

## Питання для самоперевірки

1. У яких галузях *MPEG*-відео ефективно заміняє стандартні аналогові відеоносії?



2. Розшифруйте терміни *CATV*, *DBS*, *HDTV*?
3. Як *MPEG*-відео застосовують у професійних сферах?

## 7.8. Методи стискання цифрової відеоінформації

У процесі визначення оптимального ступеня стискання відеоінформації потрібно знати, яку ціну доведеться сплатити за якісне зображення (як базові характеристики відео вплинуть на його обсяг та якість): чим більшими будуть частота кадру та глибина кольору, вищим екранний дозвіл та кращою якість зображення, тим більша продуктивність комп'ютеру знадобиться, стануть потрібними величезні обсяги дискового простору. Так, 24-бітове кольорове відео  $640 \times 480$  з частотою 30 кадрів/с вимагає передавання приблизно  $26 \text{ Mb/s}$ , призводячи до перевантаження шини та переповнювання дискового простору.

Для професійного відео найкращим рішенням є досягнення найнижчого коефіцієнту стискання.

Найчастіше для зменшення обсягу відеоданих достатньо оптимізувати один із базових параметрів відеосигналу.

Такі застосування, як ігри, комп'ютерні тренажери, відеоігроски та деякі ділові пакети часто не потребують повноекранного відео. Вони відтворюють відео в вікні, а тому не потрібно оцифровувати цілий кадр. Для цього випадку зменшують один параметр відеосигналу – дозвіл. Наприклад, при зменшенні дозволу до  $320 \times 240$  суттєво зменшиться обсяг даних, які передають за одиницю часу (приблизно до  $3 \text{ Mb/s}$ ). Можна досягти додаткового зниження обсягу даних (до

1,5 Mb/s) завдяки подальшому зменшенню розміру вікна, зниженню якості зображення та переходу з RGB-формату до YUV (4:1:1), але цього теж буде недостатньо. Тому застосовують спеціальні методи стискування.

Стискування відео потрібне для зменшення цифрових відеофайлів при максимальному збереженні якості оригіналу. Виділяють такі класи методів стискування: звичайне та в режимі реального часу; симетричне та асиметричне; з втратою якості та без її втрати; стискування відеопотоку та покадрове стискування.

Багато систем оцифровують відео та одночасно стискають його, паралельно здійснюючи зворотний процес декомпресії та відтворення. Для якісного виконання цих операцій потрібні потужні спеціальні процесори, тому більшість плат введення та виведення відео для PC побутового класу не здатні оперувати з повнометражним відео та часто пропускають кадри. Недостатня частота кадрів є однією з основних проблем для відео на PC: при продуктивності нижче за 24 кадри/с відео перестає бути плавним, порушуючи комфортність сприйняття; пропущені кадри можуть містити необхідні дані зі синхронізації звуку та зображення.

Основна відмінність методів симетричного та асиметричного стискування пов'язана зі співвідношенням способів стискування та декомпресії відео. Симетричне стискування передбачає можливість програти відеофрагмент із дозволом  $640 \times 480$  при швидкості в 30 кадрів/с, якщо оцифрування та запис були виконані з тими ж параметрами. Асиметричне стискування – це обробка 1 с відео за значно більший час. Ступінь асиметричності стискування задають у вигляді відношення  $N:1$ . Зручнішим та ефективнішим для отримання якісного відео та оптимізації

швидкості його відтворення є асиметричне стискання. На жаль, кодування повнометражного ролику може зайняти забагато часу. Якщо ж відсилати початковий матеріал на кодування до спеціалізованих фірм, це збільшує матеріальні та часові витрати на проект.

Усі методи стискання призводять до певної втрати якості відео. Навіть якщо це не помітно оку, різниця між початковим та стиснутим відео завжди є. Існує всього один алгоритм – різновид *Motion JPEG* для формату *Kodak PhotoCD*, який виконує стискання без втрат, проте: його оптимізовано для фотозображень; він працює з коефіцієнтом стискання 2:1.

Покадровий метод передбачає стискання та зберігання кожного відеокадру як окремого зображення. Стискання відеопотоку враховує те, що задній план у більшості відеосцен залишається постійним: створюють початковий кадр, а кожен наступний кадр порівнюють з попереднім та наступним зображеннями, фіксуючи розбіжність між ними. Метод стискання відеопотоку суттєво підвищує коефіцієнт стискання, зберігаючи початкову якість. Але можуть виникнути труднощі з покадровим монтажем відеоматеріалу, який закодовано таким чином.

## Питання для самоперевірки

1. Для чого призначене стискання цифрового відео?
2. Як базові характеристики цифрового відео впливають на його обсяг та якість?
3. Охарактеризуйте провідні методи стискання цифрової відеоінформації.

## 7.9. Програмне забезпечення відеомонтажу

Програмні пакети для відеомонтажу – це додатки, які використовують для редагування та монтування уривків у завершений продукт. Ці пакети містять велику кількість вбудованих спецефектів. Наприклад, найпопулярніший із них – *Adobe Premiere* – нараховує більше півсотні спецефектів.

Спецефекти (блискучі переливчасті логотипи, виштовхування з екрану одних зображень іншими, імітація атмосферних ефектів та природних явищ тощо) є невід'ємними складовими частинами виробництва відеопродукції будь-якими розробниками: робітниками телебачення, продюсерами рекламних та навчальних роликів, любителями *Quick Time* тощо. Системи відеомонтажу старшого класу, які постачають з власним ПЗ, такі як сімейство *Adobe Media*, окрім програмної реалізації ефектів, часто містять спеціальні пристрої для прискорення рендерінгу ефектів.

Відеомонтажери постійно знаходяться в пошуку ефектів, які збагачують можливості інструментарію їх програм. Для них є велика кількість програм зі спецефектами: окремих пакетів; додаткових модулів, які розширюють можливості застосування ефектів програм відеомонтажу.

Фірма *Macintosh* домінує у виробництві відеопродукції, а для її комп'ютерів створено більше програмних ефектів, ніж для *Windows*-машин.

Найбільш потужний спеціалізований пакет ефектів *Adobe After Effects* складається з базової версії та модуля *Production*

*Bundle* з додатковими можливостями та ефектами. Цей пакет є особливо сильним при виконанні ефектів, пов'язаних з рухом (створення фрагментів, які в процесі швидкого руху вздовж екрану перевертаються та змінюють розміри). Можливості *Adobe After Effects* можна розширювати за допомогою додаткових модулів.

Для визначення кращих додаткових модулів створення спецефектів, їх колекції тестують. Наочним прикладом є тестування колекцій для *Adobe After Effects*, *Premiere* та *Media* за такими критеріями: рейтинг (базується на можливостях, реалізації, передових ідеях, продуктивності, надійності, легкості в опануванні, вартості); ціна; підтримка *Adobe After Effects*, *Premiere*, *Media*; переходи; рух на фоні; комбінування часток; генерація; природні ефекти; освітлення носіїв; імітація; об'ємні ефекти; викривлення.

Були отримані такі висновки: *Boris Effects* (*Artel Software*) – чудові ефекти руху; *Aurorix* (*DigiEffects*) – багата збірка ефектів, кращі імітації носіїв; *Berserk* (*DigiEffects*) – відмінний туман, віхола, спалахи лазерів та зоряне небо; *Cyclonist* (*DigiEffects*) – потужний генератор часток; *TransFX* (*DigiEffects*) – економічні та цікаві переходи; *Dynamic Effects* (*Gryphon Software*) – добротні фільтри та переходи; *Lens Flare Pack* (*Knoll Software*) – найкращі ефекти відблисків на об'єктиві; *Final Effects AP* (*MetaTools*) – найкраща збірка фільтрів та переходів для пакету *Premiere*; *KPT Final Effects* (*MetaTools*) – чудові ефекти переготання сторінки та витіснення, гнучкий генератор часток; *Studio Effects* (*MetaTools*) – унікальні генератори часток, чудові витіснення та імітації театрального освітлення; *HoloDozo* (*M.M.M. Software*) – передовий інстру-

мент для створення об'ємних ефектів, цікавий інтерфейс користувача; *Video SpiceRack (Pixelan Software)* – ретельно продумана збірка переходів; *Ultimate for Macintosh (Ultimate)* – складний та дорогий, але не має конкурентів у сфері комбінування; *TypeCaster (Chaos tools)* – потужний генератор об'ємних текстів.

При тестуванні додаткових модулів, які містять не менше десятка ефектів, поєднаних у різні набори, для полегшення пошуку необхідних ефектів, їх розділяють на категорії. Усередині кожної категорії визначають найкращі реалізації ефекту та обирають найкращі набори, що містять різноманітні спеціальні ефекти.

У практичній роботі основну увагу слід звернути на ті ефекти, що є для вас життєво важливими: при роботі з *Adobe Premiere* або *Adobe Media* першочерговим є пакет *Boris Effects*, який містить найбагатший набір ефектів, особливо 3D-переходів та ефектів руху; при розв'язуванні задач генерації часток та ефектів викривлення неможливо обійтися без пакету *Final Effects AP*; для розв'язування задач комбінування на блакитному тлі неперевершеним є пакет *Ultimate*; при роботі з ефектами для пакету *After Effects*, перевагу слід віддати пакетам *Aurorix* та *Berserk*, які містять багату добірку корисних ефектів, а їх ціна вдвічі нижча за вартість пакетів *KPT Final Effects* та *Studio Effects*; продукція компанії *DigiEffects* є цінною додатковими модулями, демонстраційними роликами, інтерактивною документацією, набором утиліт та умовно-безкоштовних програм; пакет *Hollywood FX* (більше 100 3D-переходів і фільтрів для пакету *Premiere* та системи *Media*) та пакет *Hollywood FX Take* (більше 30 переходів) від компанії *Sinergy* підтримують багатопроцесорну обробку.

## Питання для самоперевірки

1. Назвіть найвідоміші програмні пакети для відеомонтажу та реалізації спецефектів.
2. Наведіть приклади спецефектів.
3. Як визначають кращі додаткові модулі спецефектів?

### 7.10. Апаратне забезпечення цифрового відео

**Анімаційні контролери та системи нелінійного відеомонтажу.** Традиційна технологія роботи на ПК із цифровим відео передбачає, що для запису та відтворення використовують програмно керований відеомагнітофон, який позиціонує стрічку з покадровою точністю. Це призводить до ряду недоліків: скидання на стрічку є тривалим (4 кадри/хв); режим покадрового скидання підвищує рівень шумів на стрічці; оброблене відео записують на майстер-стрічку, при дублюванні якої втрачається якість; зношування механіки магнітофону в покадровому режимі є значним; програмно керований магнітофон має високу вартість.

Ці недоліки усуває використання анімаційних та відео-контролерів, які забезпечують відтворення цифрового відео в режимі реального часу безпосередньо з диску комп'ютеру та запис з відеострічки на диск. Ця технологія надає такі переваги: не потрібен дорогий відеомагнітофон із покадровим записом;

режим використання відеотехніки є помірним; швидкість скидання відео на стрічку є високою (процес займає стільки ж часу, скільки йде кліп); майстер-стрічку можна отримати будь-яку кількість разів. Прикладами сучасних цифрових анімаційних та відеоконтролерів є: *Matrox Animation Xpress (MAX)*; *Truevision TARGA*; *DPS Personal Animation Recorder (PAR)*; *DPS Perception Video Recorder (PVR)*; *MIRO VIDEO DC/DC+*.

**Професійні відеоконспекти.** Розглянемо їх на прикладі двох провідних систем.

*Alladin Media Printer* – відкрита інтерактивна система для роботи з відеоефектами в режимі реального часу та з чотиришаровим відеомікшером *Pinnacle Alladin* від *Pinnacle Systems*. *Alladin* дозволяє обирати ефекти з сотні запрограмованих цифрових 3D-кліше та графічних переходів або формувати власні відеоефекти. Пересування та змінювання зображення можна виконувати за довільними траєкторіями. Використання субпіксельних обчислень забезпечує плавне змінювання та пересування при повній відсутності зернистості з будь-якими значеннями параметрів ефектів. Використання ключів та масок дозволяє задавати прозорість будь-якого елемента композиції.

*Alladin* забезпечує чотири відеовходи та два буфери для зображень, що дозволяє створювати відеоефекти, використовуючи до чотирьох незалежних шарів. На двох шарах можна керувати прозорістю та використовувати силует-генератор за яскравістю або кольоровістю. ПЗ для керування ефектами та мікшером працює в середовищі *Windows*, що забезпечує: простий та зручний інтерфейс; взаємодію з графічними, титрувальними та анімаційними програмами. *Alladin* можна



керувати від зовнішнього монтажного контролеру. Всі входи повинні бути синхронізовані.

Характерними особливостями цієї системи є: високо-якісна графіка; знакогенератор; чотиришаровий відеомікшер; 3D-відеоефекти в режимі реального часу; довільний розмір, положення й обертання зображення; розрізання, перегортання аркушів та хвилі; підсвітлювання та тіні; побудова нових ефектів; робота в середовищі *Windows*; мінімальні вимоги до системи (*CPU*: 80486DX2/66 або швидше; 16 Mb *RAM*; 250 Mb вільного дискового простору; відеоадаптер *Hi-Color SVGA* 800×600).

*Matrox Studio* – інтеграція всіх функцій традиційної відеостудії в єдиній комп'ютерній системі. *Matrox Studio* містить комп'ютерний монтажний пульт та поєднує в собі систему монтажу з вбудованими аудіо- та відеомікшерами, цифровою системою відеоефектів над графікою та декількома відеоканалами. Ця система реалізує найбільш завершений та цілісний підхід до виробництва відеопродукції.

*Matrox Studio* побудовано за модульним принципом. Вона дозволяє сформувати будь-яку конфігурацію, відповідну конкретному відеовиробництву, та надалі розширювати її можливості. Набір модулів *Matrox Studio* містить комутатор відеоефектів, коректор часових викривлень, відеомікшери, відеографічний контролер, контролер управління професійними відеомагнітофонами, аудіомікшер із аудіопроектором.

Мінімальний комплект постачання *Matrox Studio* забезпечує: вихідний сигнал якості мовлення; можливість мікширувати звук; отримання цифрових відеоефектів; інтерактивну обробку відеоматеріалів; керування професійними відео-

магнітофонами. До комплекту *Matrox Studio* входить спеціальний відеографічний адаптер. Доповнивши студію засобами нелінійного монтажу, можна самостійно будувати сценарій програми, використовуючи цифрове відео з диску (не потрібно передбачати час на підмотування стрічки на магнітофонах).

Характерними особливостями *Matrox Studio* є: якість мовлення; повноцінна цифрова обробка; відео (композитний сигнал, *Y/C (S-Video)*, компонентний сигнал *Y/U/V*); вбудований коректор часових викривлень (*TBC*); п'ять шарів комбінування відео та графіки; *2D*- та *3D*-відеоефекти; рух; соляризування; мозаїка; масштабування; монохром; відеомікшер (комбінування графіки та відео); силует-генератор за яскравістю та довільним кольором; десятиканальний цифровий аудіомікшер; керування будь-якими професійними магнітофонами; лінійний та нелінійний монтаж в одній системі; високоякісні титри.

*Matrox Studio* є гарно продуманою та реалізованою системою для відеомікширування.

**Обладнання для мультимедіа.** В цьому класі обладнання слід відзначити продукцію серії *AVER* від компанії *AVER Media Technologies*:

– *AVER* – кольоровий захоплювач кадру реального часу – "живе відео" на *SVGA*-моніторі, 3 входи *PAL* або *NTSC*, масштабування, панорамування, ПЗ для *Video for Windows*;

– *AVER VIDEO COMMANDER* – кольоровий захоплювач кадру реального часу – "живе" відео на *SVGA*-моніторі, *True Color*, 3 входи *PAL* або *NTSC*, масштабування, панорамування, відеоефекти, ПЗ для *Video for Windows*;

– *AVER KEY* – перетворювач *SVGA*-сигналу на відеосигнал *PAL* (композит, *S-Video*, *RGB*), фільтр пригнічування ме-

рехтінь черезрядкової розгортки, режими 640×480 та 800×600, *Overscan/Underscan*; дистанційне керування; масштабування; панорамування; стоп-кадр;

– *VGA AVER PRO* – дозволяє комбінувати графіку *SVGA*-адаптеру та стандартний композитний або *Y/C (S-Video)* сигнал; вбудований *SVGA*-адаптер (*Tseng ET*) підтримує режими до 800×600 при 64536 кольорах одночасно, забезпечує створення та обробку високоякісних графічних матеріалів, дозволяє використовувати практично будь-яке ПЗ у середовищі *MS-DOS* та *Windows*; можливе встановлювання додаткового модуля *SVGA AVER HQO Pro*, який забезпечує компонентне та *RGB/SYNC* введення та виведення на професійні відеопристрої (відеомагнітофон типу *Betacam* та інші), ефекти типу "*Fade In/Out*" та "*CrossFade*" між будь-якими двома відеоджерелами; всі відео-сигнали мають якість мовлення.

Використання ПК як домашньої відеостудії не вимагає особливих витрат та спеціальних знань: можна почати редагування домашніх відеороликів у будь-який момент.

### Питання для самоперевірки

1. Яку роль відіграють анімаційні контролери та відео-контролери?
2. Які можливості надають професійні відеокомплекси?
3. Наведіть приклади обладнання для мультимедійного відео.

## ГЛАВА 8. КОМП'ЮТЕРНА ГРАФІКА В МЕРЕЖІ *INTERNET*

### 8.1. Особливості *Internet*-графіки

**Загальні зауваження.** Популярність *Internet* та *World Wide Web* значно підсилює можливість графічного оформлення документів. Усі сучасні броузери (окрім тих, що працюють виключно в алфавітно-цифровому режимі) можуть відображати графіку в різних форматах. У процесі підготування графічного матеріалу для *Internet* слід: звертати увагу на вибір формату, розміри растру зображень, вибір палітри; враховувати, що графіку будуть передавати мережею, а відображати в основному на дисплеях ПЕОМ; пам'ятати, що в графічних системах із глибиною кольору в 8 біт (256 кольорів) використовують фіксовану палітру з 216-ти кольорів, які обрано так, щоб у палітрі були рівномірно присутні кольори спектру [27, 70, 78, 175].

**Основні типи *Internet*-графіки.** Графічне оформлення *Web*-сторінки може бути різноманітним, але декілька типів графічних вставок набули на практиці найбільшого розповсюдження [27, 70, 78]: лінії та смуги розділу (*lines, rules*), фонові зображення (*back-grounds*), заглавні зображення (*titles*), маркери списків (*bullets*), кнопки (*buttons*). В *Internet* є великі колекції цих зображень, але часто простіше нарисувати їх самому.

Нескладні прийоми швидкої розробки ілюстрованих *Web*-сторінок, які доступні звичайному користувачу, надають програми, що базуються на використанні мови гіпертекстової розмітки *HTML* [78, 82, 168]. Найчастіше для створення ілюст-

рованих *Web*-сторінок застосовують такі оператори (теги) [168]: `<html>`, `</html>` – починають та закривають сторінку; `<body background = "ім'я графічного файлу">` – заповнює сторінку фоновим рисунком, який знаходиться у файлі з вказаним іменем; `<table border = 0>` – відрисовує таблицю, в колонках та стовпчиках якої буде розташовано потрібні зображення та пояснювальні тексти (застосування цієї таблиці є зручним, а за умови `border = 0` рамок таблиці не буде видно); `</table>` – закриває таблицю; `<tr>`, `<td>`, `</tr>`, `</td>` – відкривають та закривають рядки та стовпчики вказаної таблиці; `<img>` – вставляє зображення в рядки та стовпчики таблиці, що розмічає *Web*-сторінку; `fixed` – параметр, який вказує, чи буде фон нерухомим при пересуванні тексту, чи буде рухатися (це зручно в разі використання за фон маленьких рисунків); `<pre>` – дозволяє вклеювати текст "один до одного" (це зручно, коли необхідно розташувати текст на сторінці в потрібній формі); `<hr align = "..." color = "..." noshade size = "n" width = "n">` забезпечує розташування лінії ліворуч, праворуч або по центру (*align*), зображення її певним кольором (*color*), з певною товщиною (*size*) та довжиною (*width*).

**Розповсюджені формати для передавання графіки мережею *Internet*.** Лідером серед цих форматів є *Graphics Interchange Format (GIF)* фірми *CompuServe*, що надає: гарний коефіцієнт стискання зображення (це дуже важливо при передаванні мережею); можливість створення *interlaced*-зображень (таких зображень, що поступово проявляються); просту анімацію. Проте *GIF* має суттєві обмеження, особливо на кількість кольорів у зображенні: не більше 256 унікальних кольорів (більшість сучасних відеоадаптерів здатні відображати як мінімум 32768 кольорів) [35,107].

Другим найпопулярнішим форматом подання графіки для WWW є JPEG, орієнтований на зберігання зображень фотографічної якості, що містять велику кількість кольорів. JPEG дозволяє досягти великого коефіцієнту стискання (це важливо в процесі передавання повільними каналами зв'язку). Проте при стисканні втрачаються деякі деталі зображення. Для фотографічних зображень ця втрата непомітна. Але при стисканні зображень, що містять текст або схеми, втрата якості стає відчутною [35, 107].

Зупинимося детальніше на тих можливостях, які роблять GIF та JPEG особливо привабливими для WWW [107, 167, 180].

GIF дозволяє створювати: прозорі зображення (*transparent*); зображення, що проявляються поступово, в процесі підкачування (*interlaced*). Ефект прозорості полягає в тому, що один із індексів палітри оголошують прозорим: при відображенні замість цього кольору буде використано колір під зображенням, наприклад, колір фону документу. Ефекту поступового проявлювання досягають завдяки черезрядковому відрисовуванню зображення за чотири прийоми: спочатку – кожні четверті рядки растру, потім – кожні четверті рядки зі зсувом на один рядок униз і так далі. Існує багато програм, які дозволяють надати GIF-зображенням властивостей прозорості та черезрядковості: *GIF Construction Set by Alchemy Mindworks* для MS Windows, *Image Alchemy* та *PBMPLUS* для UNIX тощо.

JPEG також дозволяє створювати файли, що відображають черезрядковим способом. Це *метод progressive JPEG*, який використовують багато програмних пакетів (наприклад, *Adobe Photoshop* має під'єднуваний модуль, що створює *progressive JPEG*).

**Вставка ілюстрацій до HTML-документів.** Для цього призначено тег `<IMG>` (без парного тегу для закриття) з декількома параметрами [78, 168]: *SRC* (обов'язковий параметр) – вказує URL-адресу (уніфікований локатор ресурсу) файлу зображення; *ALT* – рядок, що замінює зображення в процесі його завантаження та в броузерах без підтримки графіки; *WIDTH* та *HEIGHT* – горизонтальний та вертикальний розміри растру зображення (в процесі завантаження зображення браузер може відображати на його місці в документі рамку відповідного розміру); *BORDER* – задає товщину рамки навколо зображення, що є частиною активної зони посилання (якщо він дорівнює 0, рамку відображено не буде); *HSPACE* та *VSPACE* – задають вільний простір навколо зображення; *ALIGN* – визначає положення тексту відносно зображення (*TOP*, *BOTTOM*, *MIDDLE*) та положення зображення відносно вікна броузеру (*LEFT*, *RIGHT*); *USEMAP* – вказує на наявність у зображенні активних зон, описаних за допомогою тегу *MAP* у документі; *ISMAP* – вказує на наявність у зображенні активних зон та на те, що інформацію про координати обраної точки слід передавати серверу.

**Програми перетворення форматів.** Для багатьох популярних форматів даних існують програми-конвертери в HTML-формат, які дозволяють швидко та ефективно публікувати в Internet наявну інформацію, не застосовуючи повільного ручного перетворення [107]. Прикладом цих конвертерів є серія *Microsoft Internet Assistants* для програм зі складу пакету *Microsoft Office*.

**Анімація WWW-документів.** Серед різних способів цієї анімації найбільш розповсюдженими є: **анімовані GIF-файли**

(*animated-GIF*); програми, написані мовою *Java* [31, 32, 78, 107, 135, 137-139, 162]. Механізм анімованих *GIF*-файлів базується на властивості зберігати в одному файлі декілька зображень та задавати порядок їх відображення. Для створення анімованих *GIF*-файлів для *MS Windows* можна застосовувати програму *Gif Construction Set for Windows*, а останні версії багатьох ілюстраційних пакетів мають відповідні вбудовані можливості. Сучасні броузери (*Netscape Navigator*, *Microsoft Internet Explorer* та інші) коректно відображають анімовані *GIF*-файли [78, 107].

**Згладжування *Web*-зображень.** Растровим зображенням властивий ефект зазубреності на нахилених прямих та дугах, викликаний дискретністю пристроїв виведення. Він псує враження від зображень. Для усунення цього недоліку застосовують метод *anti-aliasing*, реалізований у більшості графічних пакетів: у місцях утворення зазубреностей вставляють точки проміжних кольорів.

**Рекомендації щодо стилю оформлення *Web*-документів, які містять графіку.** Наведемо основні правила, яких потрібно дотримуватися в процесі створення ілюстрованих *Web*-сторінок для *Internet*-користувачів [70, 78, 107]:

- не вставляйте до документу великих зображень, оскільки в разі відсутності доступу до швидких ліній зв'язку довге очікування завантаження зображення (10-15 секунд) призводить до втрати інтересу до *Web*-сторінки;

- не робіть довгих *Web*-сторінок, які містять багато тексту та обмаль ілюстрацій, стаючи непривабливими;

- у процесі анімації *Web*-сторінок сконцентруйте увагу користувача на тому, що він повинен для себе відзначити, не



призводячи до його постійного відволікання на миготіння анімованих зображень;

– при створенні *interlaced GIF*-зображень з прозорістю врахуйте, що деякі броузери не можуть їх коректно відображати;

– використовуйте параметр *ALT* тегу *IMG*, допомагаючи тим користувачам броузерів, які працюють в алфавітно-цифровому режимі з повільними каналами зв'язку;

– використовуйте параметри *WIDTH* та *HEIGHT* тегу *IMG* для вказування точкових розмірів зображень, допомагаючи броузерам коректніше відображати сторінки по мірі їх отримання через мережу та полегшуючи користувачам оцінку часу появи зображення (більшість броузерів відображає рамку навколо того місця, де буде розташоване зображення);

– пам'ятайте, що неправильне застосування рамок ускладнює навігацію, робить неможливим коректний запис *URL* до бази закладок та ускладнює друк сторінок на принтері;

– при перевизначенні кольорів для документу, посилання повинні виділятися на фоні іншого тексту;

– якщо ви хочете зробити *Web*-сторінку з темним фоновим зображенням та світлим текстом, обов'язково задайте (за допомогою параметру *BGCOLOR*) темний фоновий колір на той випадок, якщо фонове зображення з якихось причин не буде відображене;

– оскільки не всі користувачі мають екрани високого дозволу, уникайте широких зображень та таблиць;

– розташовуйте найважливішу інформацію на початку *Web*-сторінок, щоб вона одразу потрапляла в поле зору, без необхідності її пошуку по всьому документу;

- постійно оновлюйте *Web*-сторінку свіжою інформацією, щоб користувачі не втрагли інтересу до неї;
- основний акцент робіть на інформаційне наповнення *Web*-сторінки, а не на надто яскраве оформлення;
- ретельно вичитуйте *Web*-сторінки, перевіряйте правильність *HTML*-документів та коректність посилань;
- піклуйтеся про навігацію на *Web*-сторінках (передбачайте зміст, горизонтальні та вертикальні посилання в ієрархії сторінок, плануєте їх топологію);
- у процесі складання ієрархії *Web*-сторінок піклуйтеся про те, щоб імена файлів та каталогів мали смислове навантаження та були зрозумілими;
- вказуйте внизу *Web*-сторінки її *URL* (це допоможе користувачам передавати один одному інформацію про цю сторінку).

**Тестування *Web*-сторінок із залученням користувачів.** Образ мислення професійних дизайнерів *Web*-сторінок та програмістів суттєво відрізняється від образу мислення користувачів. Тому часто *Web*-сторінки не відповідають призначенню доносити до користувачів інформацію. Цьому можна зарадити, якщо розглядати *Web*-сторінки як інтерфейси користувача до даних, які повинні мати такі особливості [8,101,107]: інтуїтивну зрозумілість, тобто очевидність для користувача місця розташування потрібної інформації (якщо дизайнер розташував на зображенні область, яка приховує важливу інформацію, але не виділив її на тлі зображення очевидним способом, наприклад, привабливою для натискання кнопкою, то ніхто на цю інформацію не натрапить); зручність, яка забезпечує низькі затрати зусиль та часу користувача на отримання потрібної інформації.

Для досягнення максимальної зручності користувачів, *Web*-сторінки перед публікуванням потрібно тестувати [38, 78]. Наприклад, у процесі створення великих *Web*-документів важливо розподілити всю інформацію між гілками ієрархії сторінок, і до цього доцільно залучити користувачів.

Цікавою є методика розробки та тестування *Web*-сторінок компанії *Sun Microsystems*: на першому етапі користувачам пропонують набір карток із назвами матеріалів, які вони повинні розкласти групами за критерієм наближеності за змістом; на другому етапі користувачам демонструють початкові, віддруковані на папері варіанти оформлення *Web*-сторінок, а також пропонують розпізнати, що стоїть за посиланнями, та розповісти про загальні враження від дизайну; на третьому етапі користувачам демонструють варіанти *Web*-сторінок у робочому вигляді (користувачі сідають за комп'ютер із запущеним броузером та починають мандрувати по сторінках, виконуючи видані їм завдання з пошуку конкретної інформації), а інструктори спостерігають за тим, на що користувач звертає більшу увагу.

### Питання для самоперевірки

1. Назвіть ті елементи графічного оформлення *Web*-сторінок, які набули найбільшого розповсюдження на практиці.
2. Які можливості графічного оформлення *Web*-сторінок надає *HTML*?
3. Наведіть перелік правил, яких слід дотримуватися при створенні ілюстрованих *Web*-сторінок.

## 8.2. Програми для перегляду *Web*-сторінок

Структуру та методи відображення *Web*-інформації постійно вдосконалюють, сторінки стають динамічнішими. Нові застосування вимагають розвинутіших та гнучкіших засобів подання інформації та взаємодії з користувачем. Проста та звична, стандартна мова гіпертекстової розмітки *HTML* вичерпала свої можливості. Її витісняють динамічний *HTML*, *XML* (*eXtensible Markup Language*), *CSS* (каскадні аркуші стилів), *VRML* та пов'язані з ними нові технології.

У першу чергу, заміною стандартного *HTML* може стати один із варіантів мови структурованої розмітки *XML*: це дозволить подавати наповнення *Web* у вигляді розподіленої чітко структурованої БД з швидким доступом до будь-якої інформації.

Існує багато комерційних та безплатних програм для перегляду *Web*-сторінок (броузерів). Кожна з них має свої особливості, які важливо враховувати в процесі створення *Web*-сторінок. Найбільш широко відомими на практиці стали такі броузери для *Microsoft Windows* та *UNIX/XWindow*: *LYNX* (для більшості *UNIX*-систем та *OS VMS*) – працює в алфавітно-цифровому режимі; *Mosaic* (для *MS Windows*, *UNIX/Xwindow/Motif*) – є одним із перших графічних броузерів; *Arena* (для більшості *UNIX*-систем) – розроблений як експериментальний броузер комітету *W3 Consortium* для стандарту *HTML 3.0*; *Amaya* – новий проект *W3 Consortium*, що відповідає проекту стандарту *HTML*, починаючи з *V3.2*; *Cello* – доволі простий броузер для *MS Windows*; *Emacs W3 mode* – *Web*-броузер для

інтегрованого середовища *Emacs*; *HotJava* – браузер компанії *Sun Microsystems*, написаний мовою *Java*; *Netscape Navigator* (для *MS Windows*, *Sun Solaris*, *SGI IRIX*, *BSDI*, *Linux*, *IBM RS6000*, *DEC Digital UNIX*, *HP-UX*) – майже для всіх систем підтримує мову *Java*, здійснив прогресивні зрушення в світі *WWW*, першим надав такі можливості, як фонові кольори та зображення, підтримка *JPEG* тощо; *Microsoft Internet Explorer* (для *MS Windows/NT*) – підтримує мову *Java*. До популярних браузерів сьогодення слід віднести: *Internet Explorer* від *Microsoft*, *Netscape* від *America OnLine (AOL)*, *Mozilla*, *Opera*, *Amaya* (в певній мірі) [35, 36, 51, 100, 105, 121].

У цілому вказані браузери мало відрізняються один від одного. До переліку функціональних можливостей кожного з них входить підтримка: мультимедіа (відображення рисунків, відтворення анімації, звуків, фото та відео); *FTP*; групи новин та електронної пошти (окрім *Amaya*); пошуку інформації; можливості зберігати, друкувати, переглядати в автономному режимі та позначати закладками потрібні *Web*-сторінки; протоколів безпеки та можливості використання обмеженого доступу. Є й відмінності. Кожен браузер є гарним по своєму, тому перед користувачем постає задача вибору.

Браузер – складний та комплексний продукт, а тому оцінювати його складно. Важливу роль відіграють зручність користування програмою, інтеграція з іншими *Internet*-додатками, стійкість роботи, вимогливість до ресурсів, додаткові можливості. Головна задача браузеру – швидко та коректно відображати *Web*-сторінки, а для цього необхідна підтримка сучасних *Web*-стандартів. До основних типів тестів браузерів належать: сумісність з *CSS*; групування, успадковування,

загальні функції; відображення в кольорі ліній, символів та фону; каскадування, вертикальне та горизонтальне форматування, відображення різних стилів шрифтів, вирівнювання та розташовування тексту; відображення полів навколо тексту та рамок, ширина рамок, кольори та стилі рамок; відображення та позиціонування графіки, підтримка абсолютних та відносних координат; сумісність з прогресивними стандартами.

Розглянемо специфічні можливості популярних браузерів.

**Internet Explorer.** Підтримує такі можливості: пересування мережею за допомогою клавіатури; виділення посилань (при переході до посилання або рисунку навколо них з'являється рамка, яка дозволяє бачити, що саме виділено); налаштування шрифтів, формату, кольору та контрастності тексту; налаштування панелі інструментів; автозаповнювання; заміна домашньої сторінки; пошук інформації; запам'ятовує параметри з'єднання з конкретним провайдером, включаючи адреси та номери портів його *proxy*-серверів (після встановлення зв'язку необхідні настройки активізуються автоматично); попередній перегляд перед друком; підтримка нових стандартів *Web*, таких як динамічний *HTML*, *XML* та *CSS* (хоча казати про повну сумісність зі стандартами *W3C* рано, є суттєвий прогрес). Залежно від конфігурації комп'ютеру, *Internet Explorer* може працювати краще або гірше.

**Netscape Navigator.** Підтримує такі можливості: провідні технології пошуку; зміна інтерфейсу; обмін миттєвими повідомленнями; дзвінки в будь-яку країну світу крізь службу *Internet*-телефонії; бокова панель *My SideBar*, яка нагадує панель каналів *Internet Explorer*, але, окрім новин з *Internet*, на ній можна відобразити список закладок та посилань, результати

пошуку та іншу корисну інформацію, що завжди повинна бути під рукою. AOL прагне того, щоб *Netscape Navigator* витіснив *Internet Explorer*.

Необхідно додати, що: *Netscape6* переорієнтувався з *Windows* на *X-Window* та став невідрізнимим від браузеру з відкритим кодом *Mozilla*, популярного серед прихильників *Unix*; передбачена можливість мобільного змінювання інтерфейсу браузеру за допомогою "оболонок" – *skins* (наприклад, у якості заготовки є оболонка в дусі *Netscape4*); використано механізм візуалізації з проекту *Mozilla*, з яким браузер *Netscape Navigator* працює швидше, але в користувачів та розробників *Web*-сторінок можуть виникнути проблеми через намагання авторів браузеру максимально слідувати стандартам *W3C*, які часто не дотримувалися компанією *Netscape* та іншими розробниками (в першу чергу – *Microsoft*); з браузером тісно інтегровані програми *AOL Instant Messenger* (дозволяє обмінюватися миттєвими повідомленнями) та *Net2Phone* (дозволяє дзвонити в будь-яку точку світу крізь службу *Internet-телефонії*); помітним недоліком *Netscape Navigator* є відсутність підтримки підключення до декількох провайдерів.

В цілому *Netscape Navigator* – це *Mozilla*, інтегрований з *Windows*-службами AOL та прикрашений деякими додатками.

**Mozilla.** Підтримує: провідні технології пошуку; зміну інтерфейсу; обмін миттєвими повідомленнями; дзвінки до будь-якої точки світу крізь службу *Internet-телефонії*; можливість одразу отримувати працездатний браузер. У 1998 році компанія *Netscape* опублікувала початковий код свого браузеру, а організація браузеру *Mozilla* взяла на себе його супроводження. За обома браузерами стоїть одна команда розроб-

ників, однаковий механізм візуалізації та однакова поведінка в плані відображення *Web*-сторінок. *Mozilla* вміє зчитувати настройки *Netscape* та відповідно настраювати власні параметри. Але у цих броузерів є відмінності щодо процедури встановлення та видалення.

**Opera.** Має низку особливостей, що зменшують потреби в системних ресурсах та підвищують продуктивність. Броузер надає спеціальну панель *Hotlist*, яка за функціональними можливостями нагадує *My SideBar* з *Netscape*б, але є зручнішою: список закладок, адресна книжка та модуль роботи з електронною поштою оформлені в вигляді однієї панелі з вкладками, що дозволяє швидко переходити від однієї програми до іншої в межах основного вікна; те саме стосується відкриття *Web*-сторінок, оскільки, на відміну від багатьох інших броузерів, в *Opera* використовується концепція *MDI* (багатоментного інтерфейсу); забезпечене автоматичне завантаження останньої переглянутої *Web*-сторінки при повторному запуску програми.

У порівнянні з розглянутими броузерами, дистрибутив *Opera* ні в чому принципово не поступається. Деякі важливі можливості в основному дистрибутиві відсутні (мультимедійні можливості, підтримка *Java* тощо), але для подолання цього достатньо завантажити з *Web*-вузла розробників відповідний *plug-in*, сумісний з архітектурою *Netscape*. Щодо недоліків: залишає бажати кращого підтримка кирилиці; *Opera* є платним броузером.

**Amaya.** Цей броузер було створено групою розробки *Web*-стандартів консорціуму *W3C* для використання за еталон при тестуванні підтримки *Web*-стандартів, а також для тестування, відпрацьовування та демонстрації нових *Web*-технологій.



Він одночасно є і *Web*-клієнтом, і засобом розробки *Web*-сторінок. Проте сам *Amaya* не до кінця підтримує стандарт *CSS*. Цей браузер підтримує *HTML*, *XML*, *CSS*, *MathML* (дозволяє створювати складні математичні вирази та одразу поміщати їх на *Web*-сторінку). Для форматування документів можна користуватися каскадними аркушами стилів, а для їх публікування на *Web*-сервері в *Amaya* реалізовано метод *PUT* протоколу *HTTP*. *Amaya* забезпечує редагування та публікування аркушів стилів, їх зв'язування з конкретними документами, вибірку активізацію та деактивізацію з можливістю миттєвого спостереження отриманого результату. Усе це зручно для розробників. Але в браузері *Amaya* відсутні вбудовані модулі для роботи з електронною поштою та мережними новинами, та й навряд чи він сподобається кінцевим користувачам.

## Питання для самоперевірки

1. Що таке браузер?
2. Які типові функціональні можливості властиві браузерам?
3. Назвіть найпопулярніші браузери.

### 8.3. Програми-редактори Web-сторінок

*Web*-сторінки описують за допомогою спеціальної мови – *HTML*, яка стала основною мовою опису документів в *Internet* [78, 105, 107, 169].

*HTML* (мова розмітки гіпертекстової інформації) є підмножиною універсальної стандартної мови розмітки документів *SGML*, яка є стандартом для обміну документами між різними платформами. Синтаксис *HTML* повністю описується за допомогою *SGML DTD (Document Type Definition)*, а тому програми, сумісні з *SGML*, можна використовувати при підготовці *HTML*-документів [168].

Будь-який *HTML*-документ можна створити за допомогою текстового редактору, але широко застосовують і спеціалізовані програми редагування документів для *Web* [78].

Більшість *HTML*-редакторів – це *ASCII*-редактори, доповнені можливостями, що полегшують працю розробників *Web*-документів, наприклад, діалоговими вікнами вставки зображень. До них належать програми *Webber*, *DarekWare HTML Author* для *Windows*, *asWedit*, *xHTML* для *UNIX* тощо.

Друга категорія *HTML*-редакторів – це програми класу "напів-*WYSIWYG*", що забезпечують перегляд кінцевого вигляду документу по мірі його підготування (документ відображають вбудованим броузером у додатковому вікні). До них належить, наприклад, *WebEdit* для *Windows*.

Третя категорія *HTML*-редакторів, найпопулярніша серед *Web*-дизайнерів, які є початківцями – це повні *WYSIWIG*-редактори, що забезпечують максимальне наближення зовнішнього

вигляду редагованого документу до його кінцевого вигляду в браузері. При цьому користувачу не потрібно розбиратися в тегах мови *HTML*. Часто ці редактори вбудовують до *Web*-браузерів. До цієї категорії належать, наприклад: *Netscape Navigator Gold* (для всіх систем), який дозволяє набирати документ у вікні браузера; *Microsoft FrontPage*, що забезпечує створення документів у режимі *WYSIWYG*; *SoftQuad HoTMetal* (для *Windows*), який, незважаючи на уявну бідність, добре зроблений; *Microsoft Internet Assistant for Word* (з серії *Microsoft Internet Assistants*) є незамінним для тих, хто звик працювати з *Microsoft Word* або хоче перенести в *Web* існуючі документи в форматі *Word*; *W3 Amaya* є новою розробкою *W3 Consortium* – спаркою редактору та браузеру, подібною *Netscape Navigator Gold*.

У кожного з підходів до організації *HTML*-редакторів є свої переваги та недоліки. Так, шанувальники системи *LaTeX* та досвідчені розробники документів для *Web* не будуть використовувати *WYSIWYG*-редактори: за зовнішньою простотою та красою візуальних засобів розробки приховані недостатність функціональних можливостей та надмірність інформації, доданої до документу, втрачається контроль над використанням доповнень. Розумною достатністю вважають: використання досвідченими користувачами редакторів першої категорії, сервіс яких не є нав'язливим; використання тими, хто інтенсивно працює в *Microsoft Word*, *Microsoft Internet Assistant for Word*.

Зручним редактором для створення *Web*-сторінок є програма ***FrontPage Express***, яка дозволяє: використовувати всі можливості *HTML*; створювати та оформлювати *Web*-сторінки на *HTML* у процесі роботи в режимі точного відображення

*WYSIWYG* (це дозволяє одразу оцінити розмітку та оформлення), працювати з настроюванням папок (легко можна задати спосіб їх відображення) [107]. Прості у використанні, засоби *FrontPage Express* дозволяють створювати різноманітні складні сторінки *Web*.

Наприклад, можна: відкривати існуючу сторінку прямо з *Web* (якщо встановлений майстер видань *Web*) або з файлу, що знаходиться на вашому комп'ютері чи в локальній мережі; зберігати документ в *Web* (використовуючи майстри видань *Web*) або в файлі; вставляти графічні зображення з завчасно підготованих наборів (набори не входять до комплекту постачання *FrontPage Express*); застосовувати теги *HTML* зі списку на панелі інструментів, не набираючи їх вручну; вставляти доступні теги *HTML*, використовуючи команду *Розмітка HTML*; переглядати та виправляти теги *HTML* на сторінці *Web* (кольорове оформлення тегів дозволяє легко знаходити їх у тексті); виділивши текст або окремі абзаци, оформити та розташувати їх, натискаючи кнопки на панелі інструментів; додавати маркери (прокрутку тексту); додавати фонові зображення та підложку; розширювати можливості *Web*-сторінок, застосовуючи елементи управління *ActiveX*; не застосовуючи комплексного програмування, збагатити свої можливості завдяки використанню компонентів *WebBot* – "динамічних об'єктів" на *Web*-сторінці, що запускаються в той час, коли розробник зберігає сторінку на *Web*-сервері або користувач переглядає сторінку.

Комплекст *FrontPage Express* містить компоненти *WebBot*, які дозволяють поміщати часові мітки, а також засоби вставки та пошуку на сторінки *Web*. Для отримання повного набору

компонентів *WebBot* (управління формами, навігаційні панелі та інші), потрібно замінити програму на *Microsoft FrontPage*.

*DreamWeaver* також є одним із найзручніших редакторів для створення *Web*-сторінок: дозволяє редагувати *HTML*; споряджений зручним інтерфейсом, який дозволяє найкращим чином організувати послідовність дій користувача; дозволяє в процесі розробки *Web*-сторінки використовувати *JavaScript*; з легкістю надає об'єктам обтічну форму, ставлячи інтерфейс сторінки на професійний рівень; слідує за кожною дією користувача, що дозволяє відмінити, копіювати та повторювати його дії; підтримує анімацію та відтворює зображення, що використовують *QuickTime*; містить вбудовану анімацію та вбудовані звуки; працює з графічними файлами будь-яких форматів [107].

*Microsoft Word* дозволяє створювати *Web*-сторінки так само, як і звичайні *Word*-документи, за допомогою таких способів.

Перший спосіб – за допомогою майстра *Web*-сторінок, який дозволяє: створити *Web*-сторінку та *Web*-вузол; розмістити на *Web*-вузлі існуючі *Web*-сторінки та документи *Word* (щоб зробити *Web*-сторінки привабливішими, використовують рамки).

Другий спосіб – за допомогою шаблону *Web*-сторінки.

*Word* автоматично відключає можливості, що не підтримані використаним броузером (можна створювати *Web*-сторінки, не піклуючись про підтримку застосованих параметрів форматування в *Web*-броузері). Можна вибрати інший броузер або вимкнути можливість вибору броузера. Щоб зробити *Web*-сторінки динамічнішими, використовують рамки. Щоб

використати додаткові можливості форматування без майстра *Web*-сторінок, створюють сторінку рамок для розроблюваної *Web*-сторінки.

Третій спосіб – шляхом створення *Web*-сторінки з існуючого документу *Microsoft Word* (щоб швидко підготувати документи *Word* для використання в *Internet* або *intranet*, зберігають існуючий документ як *Web*-сторінку).

### Питання для самоперевірки

1. Класифікуйте програмний інструментарій редагування *Web*-сторінок.
2. Опишіть можливості програм *FrontPage Express* та *DreamWeaver*.
3. Якими способами можна створити *Web*-сторінку у середовищі текстового процесору *Word*?

## 8.4. Оптимізація ілюстрованих *Web*-сторінок

Низька пропускну здатність каналів *Internet* примушує жорстко слідкувати за розмірами елементів *Web*-сторінок, особливо графіки та *multimedia*-матеріалів.

Для зниження розмірів цих ресурсів є декілька стандартних методів, але в цілому підхід є індивідуальним: кожне зображення або будь-який інший матеріал мають свої особливості, враховуючи які, можна досягти додаткового зниження їх розміру [107].

У процесі розробки *Web*-сторінки на швидкому комп'ютері та зі швидкісним каналом зв'язку не завжди помітне те, від чого потерпають користувачі масових модемів зі звичайною швидкістю, де погано оптимізовані *Web*-сторінки неприйнятні.

Розглянемо основні прийоми оптимізації різноманітних *WWW*-матеріалів [107, 167, 180].

**Оптимізація графічних зображень.** Графічні вставки є найобсяжнішими елементами *Web*-сторінок, тому більшість прийомів оптимізації орієнтовано на них.

*Першим основоположним правилом оптимізації розміру Web-графіки є ретельний вибір формату зберігання зображення.*

В *WWW* використовують в основному два типи форматів зберігання графіки, *GIF* та *JPEG*, кожен із яких має свої переваги, недоліки та дещо різні галузі застосування.

*GIF* має обмеження на кількість кольорів, які можна одночасно використовувати в одному зображенні (не більше 256), але забезпечує непоганий коефіцієнт стискування без втрат (стиснене зображення майже повністю відповідає оригіналу).

*JPEG* використовує "повну" кольорову модель (зазвичай багата кольорова гама необхідна для зберігання цифрових фотографій) та має характерну особливість – використання алгоритму сильного стискування з втратами (зображення буде змінене для досягнення потрібного коефіцієнту стискування).

При зберіганні багатокольорових зображень типу фотографій *JPEG* є переважним, оскільки забезпечує високий ступінь стискування.

Таким чином: якщо потрібна точність зберігання кожного пікселя зображення, а обмеження на кількість кольорів не є

важливим, використовуйте формат *GIF*; якщо необхідно мати в зображенні більше 256 кольорів, не використовуючи технологій змішування, зупиніться на форматі *JPEG*, забезпечивши правильним вибором коефіцієнтів стискання оптимальне співвідношення якості та байтового розміру.

***Другим основоположним правилом оптимізації розміру Web-графіки є зберігання зображень з таким кольоровим дозволом, який відповідає кількості кольорів.***

Чим більше кольорів має зображення, тим більший обсяг воно займає. Часто при зберіганні навіть найпростіших зображень у форматі *GIF* обирають палітру з 256 кольорів (8 бітів на точку), що не завжди виправдано: якщо зображення містить менше кольорів, воно може бути збережене з нижчим кольоровим дозволом.

Наприклад, якщо використано 50 кольорів, зображення може бути збережене з палітрою в 64 кольори (6 бітів на точку), а це майже 30 % економії розміру файлу.

На жаль, цей підхід не можна застосувати до зображень в форматі *JPEG*, який завжди використовує повну кольорову модель.

***Третім основоположним правилом оптимізації розміру Web-графіки є уникання складних кольорових ефектів, які можуть викликати значне зростання обсягу файлів.***

Оскільки найважчими для алгоритмів стискання є плавні кольорові переходи, намагайтеся ними не зловживати.

Існує також проблема обмеження кількості одночасно відбитих кольорів на пристроях виведення (не всі мають потужні графічні адаптери з 16 Mb відеопам'яті та відповідними прискорювачами 3D-графіки). До того ж, оскільки те, що на



одних системах виглядає чудово, на інших може бути непривабливим, перевіряйте *Web*-сторінки на найширшому спектрі систем з різними можливостями.

**Четвертим основоположним правилом оптимізації розміру *Web*-графіки є використання спеціальних програм оптимізації *GIF*-файлів.**

Не всі графічні пакети добре оптимізовано, тому *GIF*-файли мають надлишок інформації (неоптимальні стискання, коментарі), видалити який допомагають спеціальні утиліти.

Наприклад, програма *giflite* для *DOS* може зменшувати *GIF*-файли на 10-50 %, не змінюючи зображень. Її аналогом для *UNIX* є утиліта *gifblast*.

**П'ятим основоположним правилом оптимізації розміру *Web*-графіки є перевірка того, чи не втратите ви ефекту прискорення завантаження, використовуючи *interlaced GIF*-файли.**

Створення *GIF*-файлів такими, що поступово з'являються (*interlaced*), дозволяє прискорити завантаження.

Але файли стають більшими, що викликане принципом роботи алгоритму *LZW*, використаного у форматі *GIF*: у процесі кодування черезрядкового зображення, рядки обробляють в порядку їх наступного відображення (1,4,7), що еквівалентно стисканню іншого зображення, яке складається з тих самих рядків, що й початкове, але в іншому порядку. Зображення стає складнішим через наявність елементів, які повторюються, а внаслідок – збільшенням за розміром.

**Шостим основоположним правилом оптимізації розміру *Web*-графіки є врахування того, що параметр *LOWSRC* не є порятунком від повільних каналів з'язку.**

У *Netscape Navigator* є параметр тегу `<IMG>` під назвою *LOWSRC*, який дозволяє задати зображення низького кольорового дозволу (як правило, чорно-біле), що повинне швидко передаватися та відображатися в той час, коли браузер підвантажує дійсне зображення великого розміру.

Але, додаючи до *Web*-сторінки ще одне зображення, ви збільшите її обсяг. Тому подекуди краще зробити інформативний запис у параметрі *ALT* тегу `<IMG>`.

**Сьомим основоположним правилом оптимізації *Web*-графіки є використання параметрів *WIDTH* та *HEIGHT* тегу `<IMG>` для масштабування зображень, економлячи таким чином на розмірах.**

Якщо доводиться вставляти до документу однотонове зображення (наприклад, однокольорову смугу розділу), можна зекономити в розмірах, використовуючи параметри *WIDTH* та *HEIGHT* тегу `<IMG>`: якщо вказати в них розміри, які відрізняються від реальних розмірів растру зображення, то зображення буде промасштабоване (збільшене або зменшене) до цих розмірів. Складні зображення при цьому дуже викривляться, тоді як прості та однотонні не постраждають.

Таким чином можна створити невелике зображення та, вказавши потрібні розміри, досягти необхідного ефекту з меншими втратами.

**Оптимізація мультимедіа-матеріалів.** Звуки, відеовставки та інші *multimedia*-матеріали, що використовують *Web*-сторінки, мають великий розмір та передаються мережею довго. Універсального засобу їх оптимізації не існує: все залежить від основної мети поставленої задачі. Але в першу чергу виникає питання про формати зберігання даних: *MIDI*, *WAV* (*Windows*

*Audio*), *AU* (*Sun AUdio*), *AIFF* (*Mac Audio*) – для звуку; *AVI* (*Microsoft Video for Windows*), *MPEG* (*Motion Picture Experts Group*), *MOV* (*Apple QuickTime movies*) – для відеовставок та інші.

Відеовставки – найобсяжніший елемент *Web*-сторінок. Намагайтеся не вставляти їх прямо до документу (*inline*): краще вставте один із кадрів у вигляді статичної картинки та зробіть посилання на весь фрагмент. Ретельно обирайте формат та спосіб компресії, досягаючи оптимального співвідношення розмір/якість. Як правило, в *WWW* застосовують такі формати зберігання відеофрагментів: *AVI*, *QuickTime MOV*, *MPEG*.

Формат *AVI*, який використовують в системах сімейства *Windows*, побудовано за схемою застосування "кодеків" (модулів стискування/розпаковування). Формат *QuickTime MOV* має повсюдне розповсюдження, а програми його відтворення (деяка множина кодеків) існують для більшості систем. Формат *MPEG* експертної групи з кінематографії є дуже розповсюдженим та популярним.

Розглянемо формати зберігання аудіоданих.

*MIDI* є форматом зберігання команд для різних електронних музичних інструментів, наприклад, синтезатору на музичній карті комп'ютеру.

Всередині *MIDI*-файлу зберігається нотний запис музичного фрагменту, що дозволяє досягти мінімального розміру файлу при максимальній складності мелодії.

Файли формату *MIDI* є оптимальними для запису мелодій, якщо не потрібні гарантії абсолютно ідентичного звучання інструментів на різних системах.

*WAV* – формат зберігання звуків, який використовують у системах сімейства *Windows*: він містить запис звуку в вигляді сигналу, тобто гарантує точність відтворення, якщо це дозволяє пристрій виведення.

*AU* – формат подання звуку, що використовують у системах компанії *Sun Microsystems*: він має вбудовану компресію та за принципом схожий на *WAV*.

*AIFF* – формат подання аудіоданих, який використовують у системах *Apple Computer* та *Silicon Graphics*: він має вбудовану компресію та за принципом схожий на *WAV*.

При оптимізації звукових файлів потрібно звільнитися від надлишкової інформації.

Для цифрового звуку є характерними два параметри, що впливають на розмір даних:

- частота оцифровування, що визначає, скільки разів за секунду заміряли рівень сигналу (скільки разів за секунду він буде змінюватися при нормальному відтворенні);

- розрядність рівня сигналу, що показує, скільки бітів виділялося на запис цього рівня.

Головні правила оптимізації звукових файлів такі:

- обирайте мінімальну частоту дискретизації та розрядність, за яких якість залишиться прийнятною;

- обирайте формат із максимальним показником стискування даних.

**Оптимізація тексту.** Текстові файли займають найменше місця в порівнянні з зображеннями або звуком, але економію можна отримати і на них: скоротити або видалити коментарі в *HTML*-коді (ваші власні та ті, що вставив до документу *HTML*-редактор); видалити зайві символи в кінці кожного рядка

текстового файлу (в системах сімейства *UNIX* для вказування переводу рядка використовують один символ, а в системах *DOS* та *Windows* – два).

## Питання та завдання для самоперевірки

1. Чому потрібно строго слідкувати за компактністю ресурсів *Web*-сторінок?

2. Наведіть основні правила оптимізації розміру графіки, мультимедіа-матеріалів та тексту *Web*-сторінок.

3. Оформіть *Web*-сторінку, що містить той текстовий та графічний матеріал, які були отримані при виконанні завдань 2-3 для самоперевірки до параграфу 6.3, з відповідними коментарями до них та типовими графічними елементами оформлення, застосовуючи для цього технології мультимедіа, гіперпосилань та *Web*-дизайну.

## Навчальний практикум до гл.6-8

У процесі виконання цих завдань необхідно опанувати на практиці та описати засоби та технологію створення та публікування в мережі *Internet* інтерактивної, мультимедійної та гіпермедійної *Web*-сторінки з анімованим роликком та 2D/3D-графікою, застосовуючи візуальне програмування та вбудовану мову програмування на основі акцій скрипту *ActionScript* професійного програмного пакету *Flash* або аналогічні програмні

засоби. *Web*-сторінка обов'язково повинна містити анімовані символи та кнопки, шари, текст, редаговане поле тексту, звук, можливість відправки та зберігання *Email*-адрес.

**Завдання 1.** *Web*-сторінку потрібно створити для мережі фірм із продажу, встановлювання та ремонту електронно-обчислювальної техніки та мережного обладнання, присвятивши її одному з наступних конкретних питань діяльності цих фірм: а) рекомендуванню стратегії розвитку; б) продажу продукції; в) продажу послуг; г) навчанню персоналу; д) навчанню користувачів; е) звіту про досягнення; ж) повідомленню про недоліки в діяльності та шляхи їх усунення; з) рекламній кампанії; і) діяльності відділу по зв'язкам із громадськістю; к) висвітленню загальнодоступних офіційних віомостей (адреса, ліцензія, логотип, статут тощо).

**Завдання 2.** *Web*-сторінку потрібно створити для видавничої фірми та присвятити питанню рекламування тих її поліграфічних можливостей, що реалізуються на основі застосування наступних сучасних технологій: а) цифрових видавничих технологій; б) мережних технологій; в) технологій кольорового друку; г) технологій додрукарського підготування; д) технологій віддаленого виведення кольоропроб.

**Завдання 3.** *Web*-сторінку потрібно створити для виставки, організованої центром науково-технічної інформації, та присвятити питанням досягнень в галузі: а) апаратного забезпечення ЕОМ; б) програмного забезпечення ЕОМ; в) математичного забезпечення ЕОМ; г) комп'ютерних мережних технологій; д) комп'ютерних інформаційних технологій; е) комп'ютерних графічних технологій; ж) комп'ютерних об'єктно-орієнтованих технологій.

**Завдання 4.** *Web*-сторінку потрібно створити для науково-виробничої фірми та присвятити питанням рекламування та продажу запатентованих ідей (інтелектуальних розробок) цієї фірми.

**Завдання 5.** *Web*-сторінку потрібно створити для вищого навчального закладу та присвятити питанню ведення: а) електронної дошки оголошень; б) електронного розкладу занять; в) сторінки науково-технічних та науково-методичних досягнень ВНЗ; г) електронного рекламного стенду для абітурієнтів до Дня відчинених дверей; д) сторінки університетської бібліотеки.

**Завдання 6.** *Web*-сторінку потрібно створити для бібліотечної мережі та присвятити: а) діяльності міжбібліотечного абонементу; б) питанням нових надходжень літератури; в) комп'ютеризації бібліотечних фондів та бібліотечної діяльності.

Рекомендована література для виконання цих завдань – джерела [8, 18, 27, 29, 38, 65, 70, 74, 78, 82, 89, 101, 105, 107, 167–169, 180] (на ваш вибір) або інша аналогічна література.

## ГЛАВА 9. ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ ГРАФІКИ

### 9.1. Програмування та анімація графіки на базі мов програмування високого рівня

**Огляд методів анімації.** Анімаційними (від *animation* – оживлення) називають програми, що будують, пересувають та змінюють форму зображень на екрані ЕОМ за деяким сценарієм.

Анімацію широко використовують: у розробці навчального та ігрового ПЗ; для розв'язування задач ділової, наукової, інженерної та ілюстраційної комп'ютерної графіки, мультимедіа та *Web*-дизайну в різних галузях виробничої та невиробничої сфери; при створенні інтерфейсів різних програм.

Сучасні модифікації традиційних методів анімації дозволяють отримати гарну якість навіть на середньому ПК, а для потужних систем – задіяти всі їх можливості.

**Основна ідея анімації** є простою: якщо з певною частотою чергувати на екрані кадри, що дещо відрізняються один від одного, то створюється ілюзія руху та змінювання форми об'єктів.

Процес анімації містить дві основні складові частини, які необов'язково розподілено в часі: підготування кадру та виведення його до відеосистеми.

Як правило анімаційні програми вимагають використання складних алгоритмів та великого обсягу даних. Дані, на основі яких здійснюється рух логічно виділених фігур, звичайно



отримують такими способами: зчитують з диску порціями до *Heap*-області; оголошують у програмі як масиви або списки змінних або типизованих констант; отримують алгоритмічним шляхом у тілі програми.

**Анімаційні програми працюють з двома основними класами об'єктів: спрайтами та середовищем.** Спрайт – це логічно завершений елемент зображення на екрані ЕОМ, який рухається. Середовище – це фон, уздовж якого здійснюють рух спрайту. Способи організації руху спрайту підрозділяють на комбінаційні та морфологічні.

**Комбінаційні методи анімації** базуються на модифікаціях такого алгоритму: вивести варіант спрайту на екран; стерти спрайт; вивести з деяким зсувом інший варіант спрайту. Оскільки процес здійснюється швидко, для людського ока створюється ілюзія руху.

**Морфологічні методи анімації** маніпулюють спрайтами на основі виконання одного або декількох правил. Найефективніше вони реалізуються засобами ООП.

#### ***Розглянемо питання про прості анімаційні алгоритми.***

Найпростіший метод анімації підпорядкований такому алгоритму: виводять рисунок будь-якого потрібного кольору, що не співпадає з кольором фону; через певний період часу той самий рисунок формують кольором фону, що викликає зникнення зображення; рисунок виводять в іншому місці екрану тим самим кольором, що й перше зображення.

Ще один метод, який часто застосовують початківці, має повну схожість з мультиплікацією: виведення зображення; стирання (очищення) екрану, наприклад, за допомогою команд *ClearDevice* або *ClearViewPort*; виведення нового зображення;

очищення екрану. Навіть цей елементарний спосіб дозволяє будувати ефектні програми, особливо для комп'ютерів з високою швидкістю.

**Розглянемо питання про анімацію на основі використання сторінок та образів.** Можна позбутися миготіння екрану та примусити спрайт рухатися більш плавно, організувавши виведення на різні сторінки відеопам'яті. Для цього застосовують такий алгоритм: вивести зображення на сторінку 0, яка є видимою по умовчання; сформувати нове зображення на невидимій сторінці 1; зробити видимою сторінку 1; сформувати нове зображення на невидимій сторінці 0.

Маючи чотири сторінки відеопам'яті, можна побудувати будь-яку об'ємну фігуру та розглядати її в плавному наближенні, віддаленні та обертанні під будь-яким кутом без миготіння екрану (достатньо коректно будувати потрібну фазу руху фігури на строго визначеній сторінці). Бібліотеку образів можна: сформувати безпосередньо в програмі; створити в файлі на диску однією програмою, а викликати на екран іншою програмою.

Сторінковий метод анімації часто застосовують при розробці комп'ютерних ігор.

**Розглянемо питання про методи спрайтової анімації.** Найбільш розповсюджено такий спосіб формування анімаційних кадрів, який полягає в накладанні зображень різних фаз руху на деякий фон та змінювання форми об'єкту на цьому фоні. Кожну з фаз або певну їх послідовність називають спрайтом, а анімацію – спрайтовою. Ця анімація виникла до появи комп'ютерів: щоб уникнути багаторазового перерисовування кадрів із незмінним фоном, мультиплікатори студії

Діснея використовували прозорі плівки для різних фаз руху та змінювання форм персонажів. У комп'ютерному варіанті плівки замінюють растрові або векторні образи картинок-спрайтів.

У випадку спрайтової анімації використовують псевдо-тривимірну просторову модель. Усі об'єкти сцени, як статичні (декорації), так і динамічні (актори), розташовують на площинах (шарах), паралельних площині екрану та наділених функцією прорисовування. Кожен об'єкт має координату "вглиб" екрану, що визначає порядок прорисовування об'єктів у процесі накладання на кадр та дозволяє імітувати просторову перспективу.

У найпростішому випадку декорація складається з одного об'єкта-прямокутника, що покриває життєвий простір персонажу та наділений функцією заповнювання (растровим рисунком з бітового масиву, суцільним кольором або ще якимось). Спрайт немов би літає, пересуваючись над цією областю. Під "спрайтом" надалі будемо розуміти не просто бітовий масив, який відображає об'єкт анімації, а послідовний набір усіх фаз змінювання форми та положення об'єкту в просторі (це розуміння терміну ближче до його значення в мультиплікації).

*Існують такі базові методи розв'язування основної проблеми спрайтової анімації – відновлення фону, який було "попсовано" черговою фазою спрайту: XOR-анімація; анімація з запам'ятовуванням фону в тимчасовий буфер перед виведенням чергової фази спрайту; об'єктна анімація з відновленням фону шляхом виклику функцій прорисовування тих об'єктів, які було накрито фазою персонажу; розбиття (фрагментація).*

*Метод XOR-анімації* базується на реверсивності логічної операції *XOR* (*eXclusive OR*): результат її виконання дорівнює одиниці тільки тоді, коли обидва логічних операнди мають різні значення. Зображення, виведене з використанням операції *XOR* двічі до однієї області відеобуферу, не змінює його змісту. Символічно це можна висловити так:  $(a \text{ XOR } b) \text{ XOR } b = a$ .

*XOR*-метод має особливість, що одночасно є перевагою та недоліком: підсумковий рисунок буде повністю еквівалентним початковому зображенню тільки тоді, коли всі пікселі, що потрапили під виведений образ, мають нульове значення. Інакше кольори пікселів будуть формуватися за специфічними законами, визначеними умовами реалізації відеорежиму.

Безперечною перевагою методу є те, що він не вимагає багато системної пам'яті (тільки для растру зображення, що виводять) або зовсім обходиться без неї (якщо анімують каркасний графічний примітив).

Недоліки *XOR*-анімації такі: неможливо використовувати її в анімації з декількома спрайтами, що перекриваються; у тих графічних режимах, де використовують площинну організацію пам'яті, для виконання бітових операцій у парі "дані від процесору – зміст відеобуферу" необхідно звертатися до відповідних адрес у відеопам'яті для заповнювання "регістрів засувки" (на один анімований у кадрі піксель знадобиться дві операції читання з *VRAM* та дві операції запису, не враховуючи команд звертання до портів відеоконтролеру для вибору маски біту чи маски карти, залежно від режиму).

Типовими застосуваннями *XOR*-анімації є: невибагливі курсори та маркери; ті сюжети, складові частини яких не накладаються.

*Анімація з запам'ятовуванням фону в тимчасовий буфер перед виведенням чергової фази спрайту* базується на простій ідеї: та ділянка відеопам'яті, що може бути "попсована", переноситься до тимчасового буферу, що виділяють у системній пам'яті або в пам'яті відеоадаптеру.

Недоліком є додаткові витрати пам'яті (якщо програма вимоглива до цього). Якщо *VRAM* використовують як для зберігання зображення, так і для тимчасового буферу, то можна досягти непоганої швидкодії.

Помістити до відеопам'яті всі зображення неможливо, але для тих персонажів, які використовують найчастіше, це можна зробити. Для вирішення проблеми, пов'язаної з можливістю розташування зображень в горизонтальному напрямку тільки за тими координатами, що кратні восьми або чотирьом, створюють декілька копій зображення з різним зсувом відносно початку та маскують лівий та правий краї при пересиланні зображення.

*Об'єктна анімація з відновленням фону шляхом виклику функцій прорисовування об'єктів, накритих фазою персонажу*, передбачає таке.

У разі змін у просторовому положенні чи формі спрайту, необхідно викликати процедури прорисовування тих об'єктів, які опинилися в контакті зі спрайтом, у *Z*-порядку, тобто в порядку слідування площин (шарів) псевдотривимірного простору, починаючи від найдалшої (тільки цей спосіб може ефективно реалізовувати багат шарову просторову модель загального вигляду).

Те, що на один розглянутий піксель кадру припадає два звертання до відеопам'яті, на перший погляд здається теоретичною межею. Але це – оцінка знизу: в реальній ситуації

кількість звертань до відеоадаптеру буде більшою, ніж та, що була наведена для кожного з методів. У цілому описаний метод є універсальним та найбільш вдалим у всіх відношеннях.

**Метод розбиття (фрагментації)** використовують для запобігання виникненню такої проблеми: якщо в процесі спрайтової анімації наступна фаза об'єкту мало змінюється по відношенню до попередньої фази, то прорисовування тих областей декорації, що будуть накриті спрайтом у наступному кадрі, призведе до додаткових невиправданих втрат часу (в областях перетину пікселі перерисовують двічі).

Цю проблему можна усунути просто – так, як це показано на рис.9.1 та описано далі.

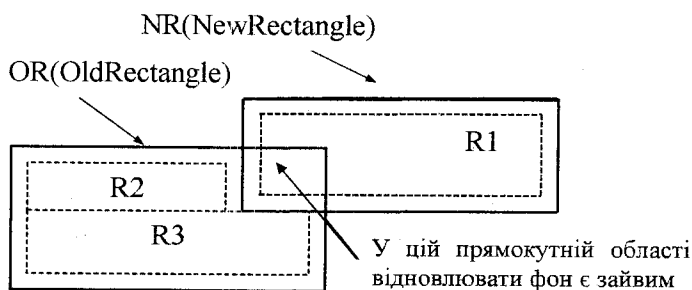


Рис.9.1. Сутність методу розбиття (фрагментації) з встановленням меж відсікання: початковими є прямокутники *OR* та *NR*, окреслені жирною лінією; на виході маємо прямокутники *R1*, *R2* та *R3*, позначені пунктирною лінією

Фазі об'єкту в будь-якому кадрі відповідає мінімальна прямокутна область, що повністю її охоплює. Проаналізуємо, як співвідносяться між собою ці області в двох суміжних кад-

рах – попередньому (область  $OR$ ) та поточному (область  $NR$ ). Якщо  $OR \& NR \neq 0$ , то: розбивають область, отриману в підсумку об'єднання прямокутників  $OR$  та  $NR$ , на декілька неперетинних прямокутників; обирають за перший прямокутник  $NR$ ; для кожного з обраних прямокутників встановлюють межі відсікання відповідної області сцени та виконують функції відрисовування об'єктів, які належать вказаній області відсікання.

У простих випадках фон+спрайт можна "рисувати" прямо до відеопам'яті. Якщо шарів багато, краще використовувати буфер у системній пам'яті (для запобігання мерехтінню в процесі відрисовування нижніх шарів та уповільненню роботи програми).

Головна перевага цього способу відновлення фону – відсутність зайвих звертань до відеопам'яті.

Якщо в анімації приймає участь декілька спрайтів, то метод розбиття ускладнюється, але основний принцип залишається незмінним: розбивають область покриття множини спрайтів у новому та старому положеннях на множину неперетинних прямокутників, для кожного з яких встановлюють межі відсікання на його сторонах у процесі прорисовування зображення до системного буферу та викликають процедури рисунку всіх об'єктів, які потрапили до цього прямокутнику.

Застосовуючи фрагментацію до маскованих зображень, необхідно виконувати основну роботу в системній пам'яті, оскільки область покриття має довільну форму. При цьому можливі такі стратегії витрачання пам'яті та порядку відрисовування.

Стратегія 1: виділення пам'яті здійснюють для повної копії робочого екрану, а процедуру перенесення зображення з

модифікованих областей до екрану викликають після того, як усе буде нарисоване. Це найпростіший шлях, але в разі високого дозволу графіки витрати пам'яті будуть надто великими.

Стратегія 2: у завчасно відведеному буфері один за одним розташовують бітові образи прямокутних областей екрану, отриманих після фрагментації; перенесення до екрану, як і в попередньому варіанті, здійснюють після заповнювання цих образів процедурами рисування об'єктів. У цьому випадку витрати пам'яті є меншими.

Стратегія 3: для кожної прямокутної області створюють на початку буферу блок необхідного розміру, здійснюють до нього прорисовування та одразу переносять зображення до відеопам'яті. Це ще економічніший спосіб, ніж попередній. Основним недоліком є помітна пауза між процесами перенесення до екрану різних неперетинних областей. Якщо додатково використовувати підхід із перемиканням відеосторінок, цей недолік стає несуттєвим.

Методи спрайтової анімації можна також класифікувати за особливостями використання багатосторінкової техніки, відеобуферу для зберігання зображень, буферу в системній пам'яті тощо. Все це допоміжні засоби, що підвищують якість відновлення фону. Важливим показником швидкодії кожного методу є кількість звертань до відеоадаптеру, його пам'яті та портів, що припадають на один піксель, який приймає участь в анімації.

**Огляд графічних можливостей мов програмування високого рівня.** До найбільш відомих сучасних мов програмування високого рівня слід віднести: *Visual Basic*, *Delphi*, *C++*



*Builder, Visual C++, Java, Visual J++, Visual Fortran* та інші. Вони мають гарні можливості створення та анімації графічних зображень (особливо ділової графіки) та надають інструментарій розробки мульти- та гіпермедійних документів, створення ілюстрованих *Web*-сторінок. Ці мови програмування містять стандартні бібліотеки візуальних графічних компонентів, за допомогою яких можна в лічені хвилини створити привабливий графічний інтерфейс до будь-якої типової прикладної програми, графічний редактор та багато іншого.

Розгляд всіх мов програмування високого рівня та всіх їх графічних можливостей міг би скласти окрему багатотомну енциклопедію. Тому обмежимося однією з найтиповіших систем – *Delphi* (з базовою мовою програмування *Object Pascal*) та оглянемо тільки ті її графічні можливості, що найчастіше застосовують на практиці [75, 86, 99, 129, 177]. Графічні можливості інших мов програмування ви можете опанувати, користуючись численною сучасною літературою, такою як джерела [14-16, 24, 30-32, 42, 45, 81, 83, 88, 96, 97, 102, 106, 112, 115, 135, 146, 151, 152, 160, 162, 163, 171].

Ті, хто має бажання самостійно опанувати графічні можливості мов програмування високого рівня, не будуть відчувати браку навчальних посібників із викладенням методів і засобів ефективного графічного програмування та прикладів графічних програм, орієнтованих на різні категорії користувачів із різною базовою підготовкою. В їх розпорядженні буде не тільки друкована література в книгарнях, а й численні електронні матеріали, що містяться в комплектах постачання та супроводження відповідної мови програмування (на *CD-ROM* та на *Web*-сайті в *Internet* фірми-розробника): добірки прикла-

дів типових графічних програм (від найпростіших до найскладніших, від учбових до високопрофесійних прикладних), демонстраційні покази, автоматизовані навчальні системи (інтерактивні, мульти- та гіпермедійні), електронні книжки (традиційні підручники, *Internet*-підручники, довідники, статті, реферати тощо).

*Розглянемо графічні компоненти системи Delphi.* Стандартна бібліотека візуальних компонентів системи *Delphi* містить ряд об'єктів, за допомогою яких можна швидко та привабливо оздобити графікою результати роботи програмного продукту. До таких об'єктів належать *TImage*, *TDBImage*, *TShape*, *TBevel* та інші.

*Візуальний компонент TImage* дозволяє розташовувати графічне зображення в будь-якому місці на формі. Для цього достатньо обрати об'єкт *TImage* з відповідної сторінки палітри компонентів та помістити в потрібну область. Зображення можна завантажити та помістити на форму під час дизайну, працюючи з властивістю *Picture* компоненту *TImage* в *Інспекторі об'єктів*. Зображення повинне зберігатися в файлі формату *Bitmap (BMP)*, *Windows Meta File (WMF)*, *Icon (ICO)*. Зображення, розташоване на формі під час дизайну, включається до файлу *\*.dpr* та підсумкового *\*.exe*-файлу, значно збільшуючи його обсяг. Для уникнення цього недоліку, зображення можна завантажувати під час виконання програми: використовують те, що властивість *Picture* є об'єктом із власним набором властивостей і методів та містить особливий метод *LoadFromFile*.

Властивості *Center* та *Stretch* з булевим типом значень допомагають сумістити центр зображення з центром об'єкту

*TImage* та розтягнути чи стиснути зображення до розмірів цього об'єкту.

Візуальний компонент *TDBImage* відображає графічне зображення, що зберігається в таблиці в спеціальному полі та початково подане виключно в форматі *BMP*.

Візуальний компонент *TShape* дозволяє розташувати на формі елементарні графічні об'єкти (квадрат, коло та інші). Конкретні типи цих об'єктів вказують у властивості *Shape*, колір та вигляд межі – у властивості *Pen*, а колір та вигляд заповнення внутрішньої частини – у властивості *Brush* (значення властивостей можна змінювати під час дизайну та виконання програми).

Візуальний компонент *TBevel* надає можливість прикрашати програму рамками та лініями, зовнішній вигляд яких задається за допомогою властивостей *Shape* та *Style*. Маючи менші можливості, ніж об'єкт *TPanel*, компонент *TBevel* не займає ресурсів.

Будь-який об'єкт бібліотеки візуальних компонентів *Delphi*, такий як форма або керувальний елемент (*TBitmap*, *TComboBox*, *TDBComboBox*, *TDBGrid*, *TDBListBox*, *TDirectoryListBox*, *TDrawGrid*, *TFileListBox*, *TForm*, *TImage*, *TListBox*, *TOutline*, *TPaintBox*, *TPrinter*, *TStringGrid* тощо) має властивість *Canvas* (канва, холст). Вона надає можливість рисування на цих компонентах, тобто відрисовування видимих частин компонентів.

Властивість *Canvas* організовано як об'єкт, який містить поле для рисування та має п'ять головних властивостей. Перші властивості є об'єктами, відомими як інструменти для рисування. Самі вони не рисують, але зберігають у властивостях

необхідні параметри. Це об'єкти *Pen* (олівець) для рисування ліній та меж геометричних фігур, *Brush* (пензель) для заповнювання фігур, *Font* (шрифт) для виведення тексту. Четверта властивість, *PenPos* є об'єктом для зберігання поточної позиції. П'ята властивість, *Pixels* є масивом для запису та зчитування пікселів холста.

Об'єкт *Canvas* містить методи для рисування графіки, виведення зображень та тексту на канву (*Arc*, *Chord*, *LineTo*, *Pie*, *Polygon*, *PolyLine*, *Rectangle*, *RoundRect*, *Draw*, *StretchDraw*, *TextOut*, *TextRect* тощо). Завдяки ним можна відтворювати на формі будь-які графічні об'єкти (цілісні зображення, окремі прямі, ламані чи криві лінії, стандартні та довільні геометричні фігури, заповнення внутрішнього простору фігур тощо), текст та багато іншого без використання компонентів *TImage*, *TShape*, *TBevel*. Тобто – без використання додаткових ресурсів (хоча за вами залишається оформлення події *OnPaint* того об'єкту, на канві якого ви рисуєте).

У цілому в *Delphi* існує два способи виведення графічної інформації: з завчасно підготованих зображень; з програми.

*Візуальний компонент TPaintBox* як правило використовується для побудови програмних додатків типу графічного редактора, а ще частіше – як область побудови графіків. Це обумовлено тим, що *TPaintBox* має єдину ключову властивість *Canvas* та являє собою канву для рисування. При цьому координати покажчика миші, передані до оброблювачів відповідних подій (*OnMouseMove* та інших), є відносними (зсув миші відносно лівого верхнього кута об'єкту *TPaintBox*).

Розглянемо основні можливості створення мультимедіа в *Delphi*. Компонент *TMediaPlayer* надає доступ до всіх основних

можливостей програмування мультимедіа, є простим у використанні: робота з ним є інтуїтивно зрозумілою та очевидною (навіть початківцям нескладно створити програму для програвання відео чи музики). За простоту роботи з мультимедіаплеєром доводиться розплачуватися тим, що в ньому реалізовано не всі потрібні можливості, наприклад, такі, як використання низькорівневих функцій.

Компонент *TMediaPlayer* надає набір підпрограм *Microsoft Media Control Interface (MCI)*, що пропонують програмістам простий доступ до широкого кола пристроїв мультимедіа.

Компонент *TMediaPlayer* оформлено у вигляді панелі керування пристроєм із кнопками. Подібно до магнітофону, є кнопки запису, відтворення, перемотування та інші.

Типова технологія роботи з компонентом *TMediaPlayer* є такою. Створюють новий проект та поміщають *TMediaPlayer* на форму. Після цього можна побачити, що *Інспектор об'єктів* містить властивість *FileName*. Двічі клацають на цій властивості та обирають ім'я файлу з розширенням *AVI*, *WAV* чи *MID* (наприклад, нехай буде обрано файл *DELPHI.AVI*). Встановлюють властивість *AutoOpen* у *True*. В підсумку цих дій програма стане готовою до запуску. Запустивши її, натискають кнопку "відтворення" та переглядають відеоролик. Якщо ж було вибрано не *AVI*, а *WAV* чи *MID*, то прослуховують звук. Таким чином можна програвати *AVI*-, *MIDI*- та *WAV*-файли, просто вказуючи їх імена.

За допомогою властивості *Display* компоненту *TMediaPlayer* можна відтворювати відео не тільки в окремому віконці, а й, наприклад, використовувати за екран для показу ролику панель. Для цього потрібно: помістити на форму компо-

нент *TPanel*, прибрати текст з властивості *Caption*, у властивості *Display* об'єкту *TMediaPlayer* вибрати зі списку *Panel1*; запустити програму та натиснути кнопку відтворення.

У *Delphi* існує два основних види мультимедійних програм, побудованих з використанням *TMediaPlayer*:

– цей компонент розташовують на формі, що дозволяє користувачу безпосередньо керувати відтворенням у тих випадках, коли йому необхідно мати простий шлях до програвання широкого кола файлів (користувач отримує доступ до жорсткого диску чи *CD-ROM* і можливість обирати та відтворювати будь-які файли);

– *TMediaPlayer* роблять невидимим установкою *Visible=False* та керують ним програмно в тих типових ситуаціях, коли необхідно відтворити звук чи відео без того, щоб користувач піклувався про їх джерело (наприклад, звук може бути частиною презентації, в рамках якої показ графіку на екрані повинен супроводжуватися поясненнями, записаними до *WAV*-файлу, а тому в процесі презентації користувач не повинен знати про існування *TMediaPlayer*, який буде працювати в фоновому режимі).

## Питання для самоперевірки

1. Сформулюйте поняття комп'ютерної анімації та наведіть перелік галузей її практичного застосування.
2. Який інструментарій візуалізації та анімації графічних зображень надають мови програмування високого рівня?

3. Опишіть основні можливості створення графіки та мультимедіа в системі *Delphi*.

## 9.2. Методи швидкого синтезу та анімації зображень із застосуванням мови програмування *Assembler*

Загальний огляд. Важливою складовою частиною розробки будь-якого професійного програмного додатку є програмування графіки (як мінімум – відповідного графічного інтерфейсу). Тому важливо знати принципи програмування графіки та мати теоретичну базу, відповідний досвід та професійні вміння створення динамічних графічних програм. Однією з найскладніших задач у процесі створення цих програм є програмування високошвидкісної та якісної анімації без мерехтінь та накладок, які здатні порушити ефект реалістичності руху.

Є декілька *базових методів створення високоякісної анімації*, кожен із яких розрахований на свою галузь застосування [7]:

– *анімація за допомогою бітових площин (bit-plane animation)* є високошвидкісною, але не дуже виразною, тому її застосовують до прикладних задач, які моделюють процеси;

– *анімація за допомогою розділених екранів* є швидкою та високоякісною, але вимагає застосування великого обсягу відеопам'яті або обмеженої частини екрану;

– *анімація за допомогою "брудних прямокутників" (dirty rectangles)* є найбільш вдалою та відповідає найвищим

вимогам, тому її використовують для створення високоякісних динамічних графічних програм, наприклад, відеоігор.

Використання "низькорівневої" мови програмування *Assembler* дозволяє підвищити продуктивність цих методів завдяки збільшенню швидкості роботи програми (за рахунок прямого доступу до відеопам'яті) [7, 62, 178, 179].

Написання функцій та процедур мовою *Assembler* – це складна та копітка праця, що вимагає гарної підготовки, але створення власного модуля мовою *Assembler* виправдовує ці витрати. Велика перевага використання мови *Assembler* у порівнянні з мовами високого рівня (особливо в програмах, які призначені для швидкого синтезу анімації) полягає в використанні ресурсів комп'ютера напряму (не застосовуючи стандартні процедури та функції мов програмування високого рівня, що не завжди є повністю коректними). Завдяки цьому програма з використанням розробок, написаних мовою *Assembler*, буде працювати швидше та ефективніше, ніж коли вона використовує процедури та функції, наприклад, мови програмування *Pascal* [7, 62, 178, 179].

**Анімація методом бітових площин.** Основна задача будь-якого методу анімації – зробити її швидкою та звільнити від мерехтінь та накладок. При цьому ПК дозволяє отримати гарну анімацію різними способами. Анімація методом бітових площин використовує наявність бітових площин та кольорової палітри для одночасної обробки декількох об'єктів високої якості, що перекривають один одного та пересуваються з високою швидкістю [7]. Анімація методом бітових площин не дає побічних ефектів, не викликає мерехтіння екрану та дозволяє рисувати зображення з максимальною швидкістю, ви-



користовуючи команду *REP MOVS*. Цей метод має обмеження: підтримує тільки чотири кольори зображення та один колір фону; вимагає, щоб зображення одного кольору не перетиналися.

Для простоти розгляду сутності анімації за допомогою бітових площин візьмемо *VGA*, де в режимах *0DH*, *0EH*, *10H* та *12H* у розпорядженні є чотири окремі бітові площини: нульова площина як правило містить синю складову частину кольору, площина 1 – зелену, площина 2 – червону, площина 3 – складову частину інтенсивності. До кожної з площин запис здійснюється індивідуально. Інформацію вилучають одночасно з чотирьох площин. Біт кожної площини контролює колір у відведеній йому точці. Перед відображенням чотирибітовий колір перетворюють у шестибітовий за допомогою ОЗП палітри, як це показано на рис.9.2. Проблеми полягають в тому, що: запис до кожної площини індивідуально вимагає багато часу; важко уникнути викривлення кольорів у разі накладання об'єктів, які займають однакові адреси відеопам'яті.

Оскільки записувати та читати можна індивідуально з кожної площини, доцільно працювати з чотирма бітовими образами окремо. Так можна уникнути наслідків перетину об'єктів: біти зображення з однієї площини не будуть суміщатися з бітами зображення з іншої.

Тобто добре мати можливість розташувати кожне зображення цілком в одній площині.

Проте і тут виникне проблема, коли різні зображення перетнуться: біти цих зображень, взаємодіючи між собою, створюють нові кольори та будуть виглядати як самостійне, не дуже привабливе зображення.

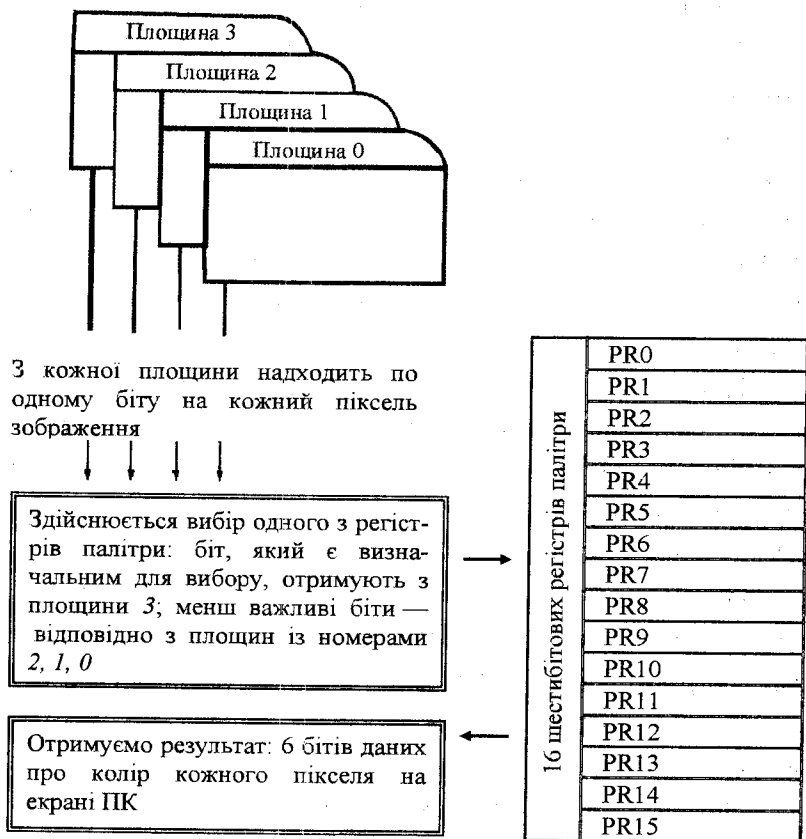


Рис.9.2. Концептуальна схема перетворення чотирьох бітів відеопам'яті в шестибітовий колір у процесі анімації методом бітових площин

Було б добре, якщо б: одне зображення виникало немов би зверху над іншим; можна було робити прозорими частини верхнього зображення.

Для вирішення проблеми розглянемо побудову регістрів палітри. Нехай чотири біти, що виходять з відеопам'яті, задають пікселю не один з 16, а один із чотирьох кольорів. Якщо встановлено біт з площини 0, то обирають один колір, якщо ж біт з площини 1 – інший. Для задавання кольору об'єкту використовується один встановлений біт із чотирьох. Якщо встановлено більше одного біту, колір визначає площина з меншим номером. Відсутність встановлених бітів відповідає кольору фону.

Цей підхід надає: чотири кольори основного зображення та один колір фону; можливість зображувати об'єкти з площин із меншими номерами над об'єктами з площин із більшими номерами та робити прозорі ділянки, виконуючи частину об'єкту в площинах із більшими номерами. Без проблем можна рисувати об'єкт цілком в одній площині, що дозволяє швидко працювати з ним. Для реалізації цієї схеми необхідно перепрограмувати ОЗП палітри так, щоб біт із площини з молодшим номером мав найбільший пріоритет – так, як показано в табл.9.1.

Чотири біти з відеопам'яті не визначають фактичний колір, а обирають відповідний регістр палітри: завдяки цьому крайній праворуч із встановлених бітів у лівому стовпці таблиці задає колір, який потрапить до екрану.

Наприклад: якщо встановлено біт із площини 0, то незалежно від стану інших бітів на екрані з'явиться червона точка, як зображено на рис.9.3; якщо біт з площини 0 скинуто, то домінуючим стає колір площини 1 – зелений колір.

Таблиця 9.1

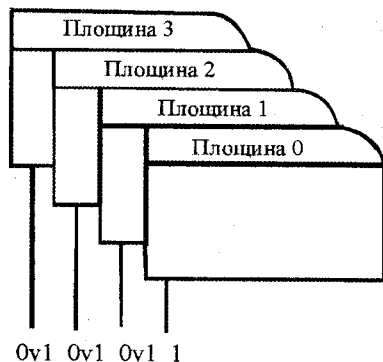
Установки ОЗП палітри для анімації за допомогою  
бітових площин

Значення бітів площин з номерами 3210 відповідно	Порядковий номер регістру палітри	Значення регістру палітри:	
		числове значення	колір
0000	0	00H	Чорний
0001	1	3CH	Червоний
0010	2	3AH	Зелений
0011	3	3CH	Червоний
0100	4	39H	Синій
0101	5	3CH	Червоний
0110	6	3AH	Зелений
0111	7	3CH	Червоний
1000	8	3FH	Білий
1001	9	3CH	Червоний
1010	10	3AH	Зелений
1011	11	3CH	Червоний
1100	12	39H	Синій
1101	13	3CH	Червоний
1110	14	3AH	Зелений
1111	15	3CH	Червоний

З цією палітрою можна досягти того, щоб об'єкти не створювали нових кольорів в областях їх перетину та накладалися правильно, дозволяючи створювати прозорість за бажанням.

Розглянемо обмеження методу "*Bit-plane animation*". Це не найкращий метод анімації. Початківці люблять використовувати його в площинних режимах 0DH, 0EH, 10H, 12H. Перепрограмування палітри зменшує кількість кольорів з 16 до 5. Кожен об'єкт може бути тільки одного кольору. Всі об'єкти

однакового пріоритету накладання повинні бути одного кольору.



Чотирибітовий номер регістру палітри

PR 0	PR 1	PR 2	PR 3	PR 4	PR 5	PR 6	PR 7	PR 8	PR 9	PR 10	PR 11	PR 12	PR 13	PR 14	PR 15
Чорний	Червоний	Зелений	Червоний	Синій	Червоний	Зелений	Червоний	Білий	Червоний	Зелений	Червоний	Синій	Червоний	Зелений	Червоний

↓

Червоний

Рис.9.3. Пріоритет бітових площин у процесі анімації методом бітових площин

Можна попрацювати над кольоровими обмеженнями, зменшивши кількість використаних площин до однієї або двох, а інші зарезервувати для багатоколірних зображень. Наприклад, можна використати площину 3 для анімації, а за допо-

могою площин 0-2 рисувати зображення в восьми кольорах. Тоді об'єкти, нарисовані в площині 3, будуть розташовані над 8-колірним зображенням.

Наприклад, якщо призначити площині 3 відображати жовтий колір, то реєстри палітри потрібно встановити так, як показано в табл.9.2. Жовтий колір відображається в будь-якій точці, де встановлено біт із площини 3. Це надає об'єктам площини 3 підвищений пріоритет і вони відображаються поверх рисунку, створеного іншими площинами (відповідну схему зображено на рис.9.4).

Таблиця 9.2

## Установки палітри для двоплощинної анімації

Регістр палітри	Вміст	Регістр палітри	Вміст
0	00Н (чорний)	8	3ЕН (жовтий)
1	01Н (синій)	9	3ЕН (жовтий)
2	02Н (зелений)	10	3ЕН (жовтий)
3	03Н (салатовий)	11	3ЕН (жовтий)
4	04Н (червоний)	12	3ЕН (жовтий)
5	05Н (фіолетовий)	13	3ЕН (жовтий)
6	14Н (коричневий)	14	3ЕН (жовтий)
7	07Н (світло-сірий)	15	3ЕН (жовтий)

Інше обмеження анімації за допомогою бітових площин таке: об'єкти, які нарисовано в одній площині, не повинні перетинатись. Інакше чисте місце, що доповнює об'єкт до квадрату, буде на мить виступати на фоні того, яке перекривають при прорисовуванні (якщо це несуттєво, то анімація годиться).

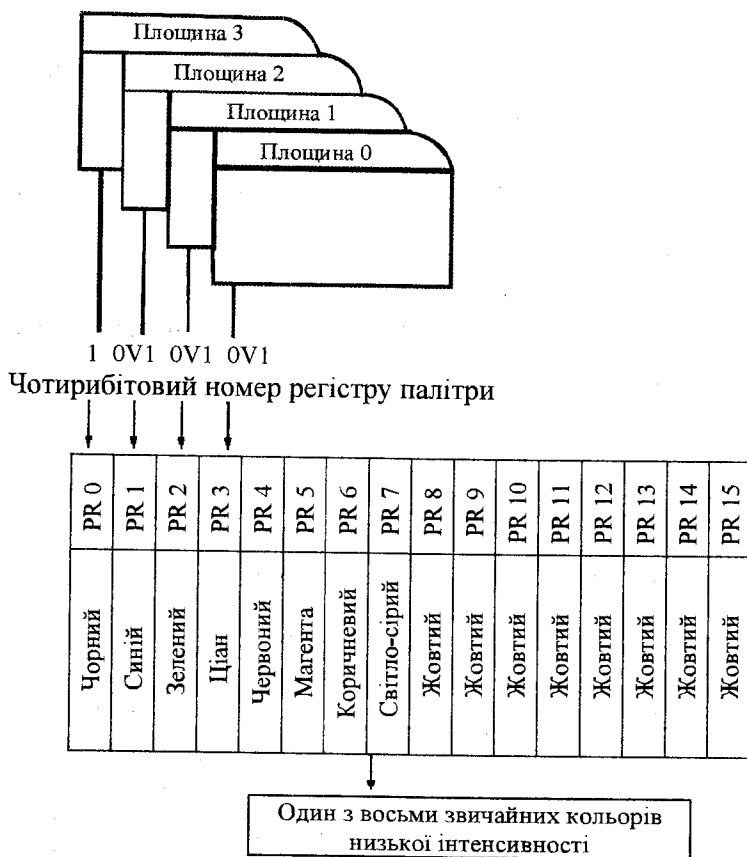


Рис.9.4. Змінений пріоритет бітових площин

Оскільки розглянутий метод анімації дає суттєві переваги (візуальні та в швидкості), доцільно долати обмеження цього методу, підвищуючи його ефективність. Наприклад, можна ретельніше аналізувати умови поставленої анімаційної задачі з тим, щоб прийти до "безболісного" рішення, коли об'єкти не будуть перетинатися та якість анімації не буде страждати.

Наведемо покроковий алгоритм типової програми анімації методом бітових площин. *Крок 1.* Встановлювання режиму 10h та палітри, відрисовування фону. *Крок 2.* Встановлювання вказівника на перший об'єкт. *Крок 3.* Перевірка умови "Чи потрібно пересувати об'єкт?". Якщо умова вірна, то перехід до кроку 4, інакше – до кроку 5. *Крок 4.* Пересування об'єкту. *Крок 5.* Перехід до наступного об'єкту. *Крок 6.* Перевірка умови "Чи закінчились об'єкти?". Якщо умова вірна, то перехід до кроку 7, інакше – до кроку 3. *Крок 7.* Затримка. *Крок 8.* Перевірка умови "Чи натиснута деяка клавіша?". Якщо умова вірна, то перехід до кроку 9, інакше – до кроку 2. *Крок 9.* Встановлювання режиму 03.

**Анімація методом розділених екранів.** Почнемо з розгляду особливостей анімації, що використовує перемикання сторінок. Застосування прийому перемикання сторінок є однією з вимог якісної анімації: він забезпечує, щоб перерисовування було невидимим для користувача. У методі перемикання сторінок застосовують два екрани: програма показує один із них, а на іншому, що є невидимим, рисується зображення [7]. Для розуміння того, як це здійснюється на практиці, необхідно знати, як ПК переводить байти з відеопам'яті в пікселі на екрані. Нехай, наприклад, бітовий образ зображення на моніторі розташовано в лінійному блоці пам'яті обсягом 64 Kb. Тоді: обирається байт з адресою 0 та виводяться ті вісім пікселів, які містяться в ньому, потім – байт з адресою 1, з розташуванням поряд із попередніми ще восьми пікселів, і так до кінця екрану. Для програмного встановлення початкової адреси, з якої потрібно починати сканування, є регістри старшої та



молодшої початкової адреси, що разом утворюють 16-бітову початкову адресу.

Для більшої наочності наведемо приклад. У режимі  $10H$  ( $640 \times 350$ , 16 кольорів *VGA*) відеопам'ять є достатньо великою, щоб умістити два повних екрани: перший екран розташуємо за адресами з 0 до 27999, другий – з 28000 до 55999, як це показано на рис.9.5. Якщо в реєстрі початкової адреси буде 0, то відобразатися буде нульова сторінка, а якщо 28000 – перша сторінка. Коли відображають нульову сторінку, то рисують, починаючи з адреси 28000, а коли сторінку 1 – з адреси 0, і так перемикають сторінки потрібну кількість разів.

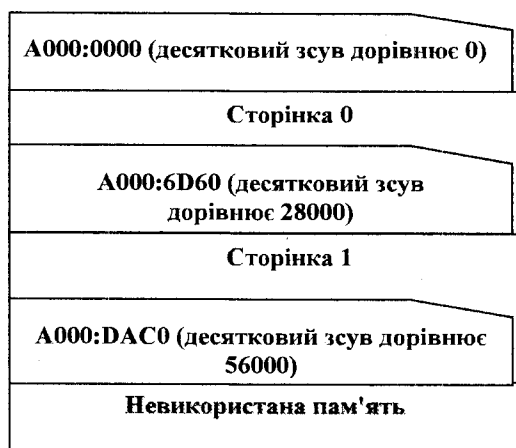


Рис.9.5. Розподіл пам'яті для перемикання сторінок у режимі  $10H$  у процесі анімації методом розділених екранів

Вирішимо, в який момент часу перемикати сторінки. Після перевстановлення початкової адреси сторінка перемикається не миттєво. Адаптер завантажує початкову адресу до внутріш-

нього регістру один раз на протязі кадру, не використовуючи вмісту цього регістру до початку нового кадру. Тобто завантажувати регістр початкової адреси можна будь-коли, але сторінки будуть перемикатись у визначений момент. Проблема полягає в необхідності зачекати, доки кадр закінчиться та буде впевненість у тому, що попередня сторінка не прорисовується на екрані. За результатами тестів, початкова адреса завантажувється по передньому фронту імпульсу вертикальної синхронізації, статус якого можна перевірити в біті 3 регістру вхідного статусу 1. Отож, щоб перемкнути сторінки, необхідно: закінчити рисунок на невидимій сторінці; дочекатися, доки буде дозволене відображення; встановити нову початкову адресу та дочекатись імпульсу вертикальної синхронізації. Далі, з повною впевненістю в тому, що сторінки перемкнулися, можна рисувати на поточній невидимій сторінці.

Позитивним додатковим ефектом цього перемикання сторінок є автоматична синхронізація програми частотою регенерації екрану (60/70 Гц).

Для прорисовування складних зображень може знадобитися більше одного кадру, що компенсується введенням лічильника кадрів та відстежуванням загальної швидкості їх чергування за допомогою таймеру через *BIOS*.

Обговоримо питання про застосування режиму запису 3. На *VGA* можливе ефективне оновлення бітового образу завдяки тому, що можна копіювати однією операцією 8 пікселів. У стандартному режимі 12H байт відеопам'яті містить 8 пікселів, кожен із яких "складається" з чотирьох бітів, по одному з кожної бітової площини. Часто необхідно змінювати лише частину пікселів, які знаходяться в одному байті. Але на практиці можна

перезаписати тільки весь байт цілком. Вирішити проблему за допомогою команд *AND* та *OR* неможливо: вони спрацюють для однієї бітової площини; якщо ж використовувати їх, записуючи до кожної площини індивідуально, то знизиться швидкодія. Тому доведеться використати апаратні можливості.

Цю проблему дозволяє вирішити режим запису 3. Регістр графічного контролеру встановлюють у цей режим. До регістру встановлювання/скидання заносять значення потрібного кольору (0-15). З відеопам'яті читають, а потім записують до неї новий байт, встановивши ті біти, що відповідають за точки потрібного кольору. Точки, що відповідають скинутим бітам записаного байту, залишаються такими ж, як і були.

Наприклад, якщо режим запису 3 встановлено та регістр встановлювання/скидання містить 1 (синій колір), то необхідно виконати таку послідовність команд (вона зробить перші чотири пікселі на екрані синіми, залишивши інші без змін).

*Mov Dx,0A000H* ;завантаження до *Dx* адреси сегменту  
;відеопам'яті в режимі *10H*

*Mov Es,Dx* ;перенесення до додаткового сегменту даних.

*Mov Al, Es:[0]* ;завантаження до *Al* першого байту з  
;відеопам'яті

*Mov Byte Ptr Es:[0],11110000B* ;завантаження до першого  
;байту відеопам'яті нового значення

Можна оптимізувати цей фрагмент програмного коду.

*Mov Dx,0A000H* ;завантаження до *Dx* адреси сегменту  
;відеопам'яті в режимі *10H*

*Mov Es,Dx* ;перенесення до додаткового сегменту даних

*Mov Al,11110000B* ;завантаження до *Al* нового значення  
;відеопам'яті

*Xchg Es:[0],Al* ;обмін вмісту першого байту відеопам'яті ;та регістру *Al*.

Якщо встановлювати одразу всі пікселі в байті, то читання з відеопам'яті не потрібне.

*Mov Dx,0A000H* ;завантаження до *Dx* адреси сегменту ;відеопам'яті в режимі *10H*

*Mov Es,Dx* ;перенесення до додаткового сегменту даних.

*Mov Byte Ptr Es:[0],11111111B* ;завантаження до першого ;байту відеопам'яті нового значення.

Розглянемо сутність методу розділеного екрану.

Якщо перейти до режиму  $640 \times 480$ , який забезпечує однакову відстань між сусідніми точками по горизонталі та по вертикалі та найкращий дозвіл серед шістнадцятиколірових режимів, то одна сторінка займе 38400 байтів відеопам'яті. Тобто на 64 *Kb* неможливо буде розмістити дві повні сторінки. Проблему вирішують завдяки підтримці адаптером технології "розділеного екрану", яка дозволяє, в разі досягнення заданої лінії екрану, обнуляти значення вказівника сканування відеопам'яті.

Розглянемо приклад. Нехай задано так, щоб адаптер сканував з відеопам'яті перші 338 ліній екрану традиційно (з перемиканням сторінок), а зображення з ліній із номерами 339 та далі відрисовував знову з нульової адреси відеопам'яті. Це дозволяє здійснити розподіл відеопам'яті для перемикання сторінок у режимі *12H* та розділити екран на області так, як зображено на рис.9.6.

Тобто ми отримали можливість здійснювати анімацію, перемикаючи сторінки в перших 339 лініях екрану (замість 339 можна обирати інше значення), а інша площа екрану (141 лінія) буде містити статичне зображення (різну корисну інформацію,

що присутня в будь-якій програмі, наприклад, логотип, рядок допомоги, рейтинг тощо). Межі між областями розділеного екрану видно не буде.

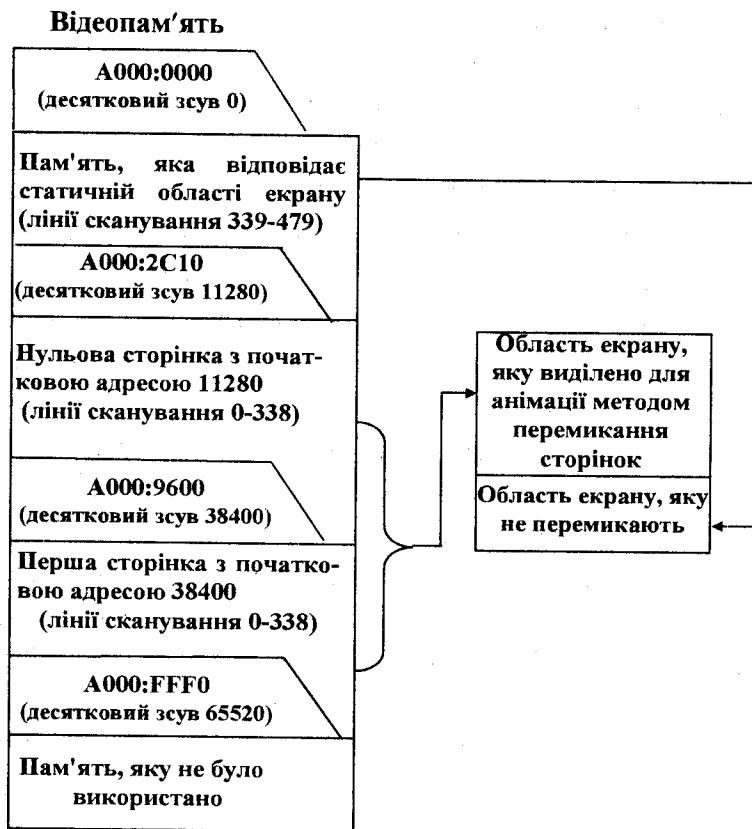


Рис.9.6. Концептуальна схема анімації методом розділених екранів

Покроковий алгоритм реалізації цього методу анімації в узагальненому вигляді подано нижче.

*Крок 1.* Встановлюємо режим 640×480×16 та вмикаємо розділений екран. *Крок 2.* Відображаємо сторінку 0 над лінією розділу екрану. *Крок 3.* Очищаємо обидві сторінки кольором фону та рисуємо межі на кожній сторінці. *Крок 4.* Рисуємо статичне зображення під лінією розділу екрану. *Крок 5.* Рисуємо об'єкт в його першій позиції. *Крок 6.* Пересуваємо об'єкт. *Крок 7.* Рисуємо об'єкт на невидимій сторінці. *Крок 8.* Перемикаємо сторінки. *Крок 9.* Перевіряємо умову "Чи натиснуто клавішу?". Якщо умова вірна, переходимо до кроку 10, інакше – до кроку 6. *Крок 10.* Відновлюємо текстовий режим.

На відміну від розглянутого прикладу, є велика кількість адаптерів з 256 Kb та більшим обсягом пам'яті, наприклад, *SuperVGA*. Але для швидкої анімації великі сегменти пам'яті не потрібні та не підходять, тому розглянутий приклад *VGA* є типовим.

**Особливості анімації методом прямокутників (*dirty rectangles*).** Поява цього методу була викликана відносною повільністю графічних адаптерів для ПК.

Основна його ідея полягає в зберіганні повної копії екрану не в відеопам'яті, а в звичайній пам'яті [7]. Рисунок виконують цілком у цьому системному буфері.

Далі працюють зі списком прямокутних областей екрану, що отримали назву *dirty rectangles* – "брудних прямокутників" (у них сталися зміни, а тому їх потрібно поновити на екрані). Як тільки кадр закінчиться, "брудні прямокутники" будуть скопійовані до відеопам'яті. Потім цикл почнеться спочатку.

Цей підхід забезпечує такі переваги [7]:

- візуальну якість, оскільки в процесі рисування ніколи не буде видно проміжкової прорисовки, а тільки – повністю

оновлене зображення (якщо б прорисовування екрану здійснювалося безпосередньо до відеопам'яті, то спостерігалось б миготіння зображення при стиранні та перерисовуванні об'єктів);

- полегшення обробки об'єктів, які накладаються один на одного (найдальші відрисовують першими).

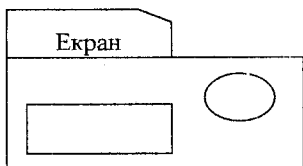
Рис.9.7 демонструє, як метод "брудних прямокутників" успішно справляється з проблемами анімації, що спирається на безпосереднє відображення на екран.

Таким чином, анімацію методом *dirty rectangles* підпорядковано такому покроковому алгоритму.

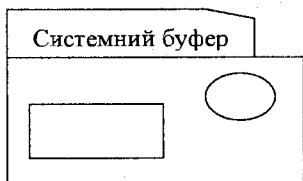
*Крок 1.* Виділяємо пам'ять для буферу рисування. *Крок 2.* Встановлюємо об'єкти, що потрібно пересувати, та потрібний графічний режим. *Крок 3.* Заповнюємо оперативну пам'ять та рисуємо "брудні прямокутники". *Крок 4.* Очищаємо список "брудних прямокутників". *Крок 5.* Стираємо об'єкти в ОЗП в їх старих позиціях, поновлюючи список "брудних прямокутників". *Крок 6.* Пересуваємо об'єкти, відштовхуючи їх від меж екрану. *Крок 7.* Перевіряємо умову "Чи натиснуто клавішу?". Якщо умова вірна, переходимо до кроку 8, інакше – до кроку 3. *Крок 8.* Відновлюємо текстовий режим.

Розглянемо, чому для отримання візуальної якості не завжди можна використовувати перемикання сторінок, а доцільніше застосовувати метод *dirty rectangles*.

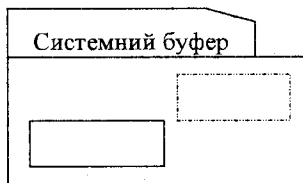
По-перше, анімацію методом *dirty rectangles* підтримує будь-який адаптер у будь-якому режимі, на відміну від перемикання сторінок.



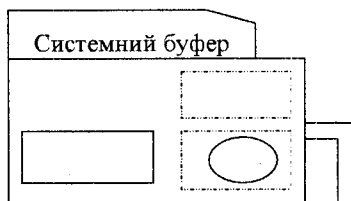
Зображення еліпсу знаходиться в позиції 1 екрану



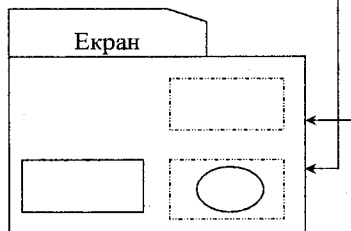
Зображення еліпсу знаходиться в позиції 1 системного буферу



Зображення еліпсу стерте в позиції 1 системного буферу



Зображення еліпсу нарисоване в позиції 2 системного буферу



Брудні прямокутники скопійовано з системного буферу на екран: в позицію 1 екрану скопійовано прямокутник, який має колір фону; в позицію 2 – прямокутник, який містить зображення еліпсу.

Рис.9.7. Анімація методом *dirty rectangles*

("брудні прямокутники" зображено за допомогою пунктиру)



По-друге, метод *dirty rectangles* працює швидше завдяки мінімізації кількості звертань до адаптеру: поновлення картинки здійснюється в системній пам'яті, а не в екранній; об'єкти рідко пересуваються за межі кадру по всій довжині екрану, що дозволяє перерисовувати тільки змінені області екрану, а не весь екран, як у разі перемикання сторінок; операції зчитування, редагування та запису здійснюються в системній пам'яті швидше, ніж у разі взаємодії з адаптером *VGA*; зчитування з відеопам'яті стають рідкими, а вони займають більше часу, ніж запис.

По-третє, перемикання сторінок вимагає значного часу на очікування міжкадрового проміжку, а в разі анімації методом *dirty rectangles* це очікування відсутнє, оскільки частково прорисовані зображення не з'являються в відеопам'яті: все рисування здійснюється в звичайній пам'яті.

Розглянемо недоліки методу *dirty rectangles*. По-перше, буває, що в процесі прорисовування прямокутника адаптер нарисує одну його частину в старому варіанті, а іншу частину – поновленою, на протязі одного кадру. Це – ефект *Shearing* (стрижки). При заповнюванні невеликих областей він не порушує відчуття реальності, але в разі роботи з великими об'єктами створює проблеми. На відміну від цього, перемикання сторінок відображає завершене зображення. По-друге, в разі необхідності відвести під заповнювання "брудних прямокутників" більше одного кадру, на екрані можуть з'явитися сторонні ефекти від поступового оновлення по прямокутниках.

Оптимізація методу "брудних прямокутників" на основі маскуванню образів полягає в доцільності виведення окремих пікселів та цілих об'єктів крізь маску, якщо: об'єкти

перетинаються, створюють неправильні форми, мають прозорі області; потрібно відобразити об'єкт над чи під іншим об'єктом однією операцією. Недолік маскованого виведення в порівнянні з прямокутним – повільність. Найчастіше використовують роздільне зберігання об'єкту та маски. Можна робити простіше – копіювати пікселі, що відрізняються від заданого прозорого кольору. Тоді не потрібно зберігати маску, але гнучкість методу знижується (випадає з використання один колір; неможливо використовувати об'єкт з різними масками, оскільки маска є вбудованою до об'єкту).

Можливість детально ознайомитися з методами швидкого та ефективного графічного програмування та анімації на базі мови *Assembler* надає серія книжок М.Абраша (книга [7] та її щорічні перевидання), в яких викладено: прийоми написання графічних програм із найоптимальнішим кодом, швидкою та якісною 2D/3D-графікою та анімацією, підвищенням продуктивності графіки; напрямки розвитку майстерності програмування графіки; ключові графічні елементи програмних додатків для ПК (ігри, мультиплікація, наочна візуалізація, САПР, імітація польотів тощо); численні приклади графічних програм, унаочнюючих теорію. Ця серія книжок, призначених як для новачків, так і для програмістів середнього класу, дозволяє не тільки успішно почати працювати з графікою на ПК, а й швидко та легко створити комерційні графічні програми (наприклад, ігрові).

## Питання та завдання для самоперевірки

1. У чому полягають переваги та недоліки технології синтезу та анімації графічних зображень засобами мови *Assembler*?

2. Виконайте порівняльний аналіз методів анімації за допомогою інструментальних засобів мови *Assembler*.

3. Створіть на базі інструментальних засобів системи *Delphi* або інших мов програмування високого рівня мульти- та гіпермедійну анімаційну програму, що демонструє ваші знання та відбиває ваші уподобання в області графічних можливостей мов програмування високого рівня, доповнивши її ефективними асемблерними вставками. Організуйте можливість її запуску з тієї ж *Web*-сторінки, на якій було розташовано всі попередні матеріали (див. завдання для самоперевірки до параграфу 8.4).

## ГЛАВА 10. ВИДАВНИЧІ КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ

### 10.1. Ключові тенденції комп'ютерної поліграфії

Останню чверть ХХ сторіччя характеризують прогресивні зрушення в поліграфії: від лінотипів – до офсетної літографії; від металевого набору – до фотографічного; від репрокамер – до електронного сканування; від традиційного друку – до цифрового.

*Сучасному етапу комп'ютерної поліграфії властива конкурентна боротьба таких видавничих технологій [28, 63, 137-139]:* сканерів – з цифровими фотоапаратами; настільних видавничих систем – зі спеціалізованими; ФНА (фотонабірних автоматів, що виводять на плівки окремі сторінки) – з імносетерами (автоматами, що виводять на плівки спущені смуги цілком); сухих матеріалів без вмісту срібла – з мокрими плівками на срібній основі; технології "комп'ютер – друкарська форма" (СТР), яка виключає з виробничого циклу фотоплівки – з ФНА; термальних друкарських форм – з формами на срібній основі та фотополімерними формами; традиційних друкарських машин, для яких форми готують окремо – з такими друкарськими машинами, для яких форми готують безпосередньо на місці; кольорових цифрових принтерів, які виводять на звичайний папір зображення з комерційною якістю та виключають з виробничого циклу плівки, друкарські форми та друкарські машини – з цими "виключеними" технологіями.

*Розглянемо основні тенденції в комп'ютерній поліграфії.*

Професійні робочі станції заміняють настільними комп'ютерами, під'єднуючи які до швидкісних мереж, отримують повністю комп'ютеризовані виробничі технологічні ланцюжки.

Професійні комп'ютерні художники та дизайнери самі виконують багато додрукарських функцій (виведення діапозитивів, сканування, попередня перевірка перед виведенням тощо), не перекладаючи їх на бюро додрукарського підготування.

Цифрові фотокамери та сканери дозволяють дешево та з високою якістю робити знімки та створювати бази цифрових зображень, які можна використовувати як друквані матеріали та в інших засобах розповсюдження візуальної інформації.

Що стосується фотонабору та виведення на плівки спускових смуг, то вдосконалюють технології на базі посторінкових діапозитивів, діапозитивів повних спускових смуг та діапозитивів на сухих плівках.

Налагоджування друкарських машин із безпосереднім виведенням вдосконалено так, щоб отримувати кольорову продукцію великими та малими тиражами; розроблено технологію підготування форм безпосередньо на друкарській машині.

Технологія "комп'ютер – друкарська форма" (СТР) стає домінуючою та містить безліч варіантів (від полієстрових друкарських форм, які можна експонувати безпосередньо на ФНА, до термальних форм, які експонують в автоматичних СТР-системах).

У сфері кольорового цифрового друку: цифрові кольоропробні пристрої (починаючи зі струминних та сублимаційних та закінчуючи апаратами типу *Kodak Approval*) дозволяють перейти на повністю цифровий стиль роботи; можливе віддалене виведення кольоропроб (ознайомитися з цими технологіями можна в рубриці видавничих технологій журналу "Компьютер пресс").

Переносні формати документів (*PDF*) дозволяють архівувати, передавати, читати з екрану та друкувати на папері будь-які електронні сторінки, повністю зберігаючи їх зовнішній вигляд, а технологія *Adobe Acrobat* вбирає в себе все більше нових можливостей друку, здатних покращити цифрові технологічні процеси.

### Питання для самоперевірки

1. Які прогресивні зрушення в поліграфії сталися за останню чверть ХХ сторіччя?
2. Сформулюйте ключові тенденції комп'ютерної поліграфії.
3. Що означає скорочення *СТР*?

## 10.2. Цифровий технологічний цикл поліграфічного виробництва

Для галузі поліграфічного виробництва та друку глобальними процесами є [28, 91, 180]: збільшення доступності та спрощення для замовників процесу друку в найширшому діапазоні якості та видів поліграфічної продукції; наближення всіх етапів підготовки та виробництва поліграфічної продукції до робочого місця редактора, художника, дизайнера, виробника; вдосконалення додрукарських та друкарських процесів та підходів до подання інформаційних матеріалів читачам та користувачам інформації, що здійснюється завдяки активному розвитку засобів комунікації та світової популярності глобальної мережі *Internet* та служби *WWW*; збільшення різноманітності та стрімкий розвиток систем та технологій випуску друкованої продукції, особливо цифрових комп'ютерних технологій, з усе ширшим використанням найсучаснішого апаратного та програмного забезпечення на всіх стадіях видавництва, починаючи з набору текстів і введення зображень та завершуючи друком тиражу.

Активне використання цифрових технологій приводить до появи досконаліших пристроїв та якісно нових принципів роботи, таких як: покращені цифрові камери для фотозйомки з професійною якістю, системи кольорового цифрового друку (*Direct-to-Print*) та пристрої виведення офсетних форм (*Direct-to-Plate*), покращені системи кольорокалібрації на всіх етапах видавничого процесу.

Ці процеси підсилюють роль системного підходу до формування та модернізації редакційно-видавничого виробництва.

**Планшетна та барабанна технології професійних сканерів.** *У поліграфії професійні сканери використовують для оцифрування зображень з кольорових слайдів або фотографій [176].*

Створення на комп'ютері будь-яких графічних робіт (колажів, фотолистівок, обкладинок журналів, рекламних аркушів, афіш та буклетів тощо) починається з підбирання та введення зображень. *Основним носієм початкової графічної інформації для поліграфічних та дизайнерських потреб є кольорові слайди на фотоплівці та традиційні (нецифрові) фотографії*, незважаючи на численність способів зберігання графічної інформації (бібліотеки CD-ROM, спеціалізовані БД цифрових графічних зображень тощо).

*Вимоги до сканеру будуть тим вищими, чим: більші вимоги до якості зображення, що вводять; складніший початковий слайд; більшу кількість слайдів потрібно відсканувати за виділений час.*

*У випадку планшетної технології сканування:* оригінали розташовують на пласкому піддоні; в якості оптичного сенсору використовують лінійну ПЗЗ-матрицю з великою кількістю чарунок, яка визначає оптичний дозвіл сканеру; сканування здійснюють в горизонтальному напрямку, лінія за лінією, при відносному пересуванні оригіналу та ПЗЗ-матриці на один крок обраного механічного дозволу сканеру. Горизонтальний та вертикальний дозвіл планшетних сканерів не співпадають: більше значення відповідає механічному дозволу (дискретизації пересування каретки); менше значення –



кількості чарунок ПЗЗ-матриці. Тому застосовують матриці, розраховані на різну кількість елементів, яка відповідає різним класам планшетних сканерів.

Параметр максимальної оптичної щільності (у планшетних сканерів вона невелика, до того ж у настільних – менша, ніж у професійних) характеризує можливість сканеру сприймати інформацію з оптично щільних слайдів, що забезпечує краще опрацювання їх темних областей. Більшої максимальної оптичної щільності вдається досягти завдяки штучному зміщенню оптичного діапазону сканеру до області тіней, але тільки тоді, коли сканер має значний запас розрядності (щоб не занадто звузити гістограму відсканованого зображення).

*У випадку барабанної технології сканування:* оригінали розташовують на поверхні прозорого барабану, що обертається; оптичним сенсором є три фоточутливих елементи (для червоної, синьої та зеленої складових частин); сенсор рухається вздовж осі барабану, що обертається, та коло за колом сканує зображення (за одиницю часу сканується точка оригіналу). Швидкодію цього сканеру визначає швидкість обертання барабану, а його оптичний дозвіл – крок пересування оптичного сенсору вздовж осі барабану. Залежно від кроку пересування оптичної системи, змінюють апертуру (діаметр отвору, крізь який світло проходить від оригіналу до фотоелементів), що усуває вплив сусідніх точок оригіналу на поточну скановану область.

Професійні та дешеві барабанні сканери мають розбіжності в конструкції (кожен із варіантів надає свої переваги та недоліки): за способом розташування та закріплення барабану

(оскільки в процесі обертання виникає вібрація однієї з його сторін, то гарним рішенням є вертикальне закріплення барабану); за типом фоточутливих елементів (у професійних моделях – це фотоелектронні примножувачі, що забезпечують високі чутливість та максимальну оптичну щільність; у дешевих моделях – це фотодіоди з не дуже високою чутливістю, що знижує максимальну оптичну щільність); за кількістю апертур, яка визначає кількість тих дозволів, які можна застосовувати в процесі сканування.

Оскільки досягти ідеалу – плавного змінювання апертури та, відповідно, дозволу – не вдається, бо виконати апаратно ці апертури складно, на практиці використовують набір із декількох апертур заданого діаметру. Чим більша кількість апертур у сканері, тим гнучкіше можна задавати дозвіл сканування. Для кожної апертури потрібен свій стан фокусування оптичної системи. Є сканери, що використовують окрему лінзу для кожного значення апертури, але вигідніше використовувати одну гнучку оптичну систему, оскільки задавання фокусу в ній можна здійснювати максимально точно для кожного значення апертури.

**Фотонабірні автомати (ФНА) та технології.** Порівняємо *ролеві (капстанові) та барабанні* фотонабірні технології.

Початково ФНА виготовляли виключно в вигляді складних пристроїв, які забезпечували найвищу якість результату, але коштували дорого та потребували складного кваліфікованого обслуговування. Тому невеликі видавництва газет, редакції журналів та рекламні агенства не могли дозволити собі власних фотовивідних пристроїв та виконання операцій кольороподілу.

Проблема високої складності та вартості пов'язана з традиційною технологією експонування лазерним променем у ФНА з внутрішнім барабаном (рис.10.1): фоточутливий матеріал розташовують навколо призми по колу, що забезпечує рівну відстань від призми до поверхні плівки; здійснюється експонування фотоформи, що непорушно закріплена на внутрішній поверхні порожнього замкненого циліндру; експонувальна система пересувається вздовж осі симетрії барабану, а лінза, що обертається, забезпечує сканування променя впоперек напрямку руху оптичної системи, вздовж радіусу барабану.

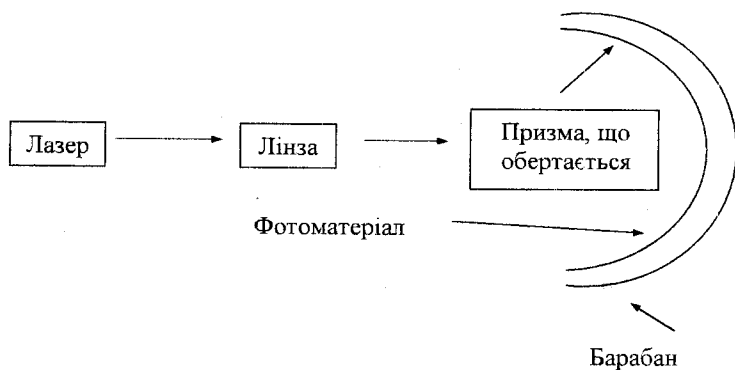


Рис.10.1. Конструкція ФНА з внутрішнім барабаном

Завдяки цій технології досягають високих значень точності позиціонування променя та повторюваності відбитків уздовж усього формату. Тому всі сучасні високоточні ФНА компанії *Heidelberg Prepress* виконані за принципом "внутрішній барабан". Але цей спосіб висуває високі вимоги до якості та точності виготовлення барабану та його напівокруглих боковин, уздовж яких розправлено плівку, що значно підвищує вартість пристрою.

Зменшення вартості та складності ФНА для практичних потреб невеликих видавництв стало реальним після розробки американською компанією *Ultr* ефективної та економічнішої технології фотоекспонування на базі напівпровідникового лазера, а також створення на її основі ФНА з інфрачервоним лазерним діодом.

Принцип, закладений до механізму ФНА від *Ultr*, називають ролевим або "капстановим" (від англ. *capstan* – рол, вал) через використання механізму подачі фотоматеріалу крізь систему валів. Як показано на рис.10.2, фоточутливий матеріал пересувають планарно відносно каретки з призмою, що обертається. Тобто фотоплівку протягують вздовж фокальної площини (лінії). В процесі обертання призми лазерний промінь пересувається від одного краю фотоматеріалу до іншого, при цьому експонують лише одну лінію точок уздовж усієї ширини матеріалу. Потім плівку зсувають для експонування наступної лінії, а описаний процес повторюють.

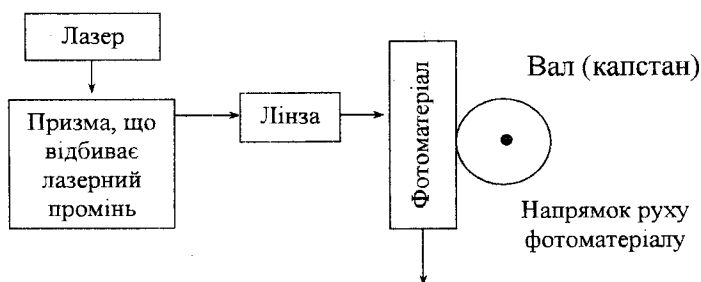


Рис.10.2. Конструкція ролевого (капстанового) ФНА

Цей принцип є набагато простішим та дешевшим за традиційний, а завдяки його використанню пристрій стає надійнішим та зручнішим в експлуатації та обслуговуванні.

Уздовж країв матеріалу спостерігається деяке збільшення діаметру світлової плями через невелике розфокусування лазерного променя та відхилення кута експонування від ідеального ( $90^\circ$ ). Для усунення цього недоліку за допомогою встановлюють спеціальну лінзу, що забезпечує необхідну корекцію фокусу.

Ця лінза працює ефективно тільки тоді, якщо кути відхилення лазерного променя не перевищують  $60^\circ$ . До її якості висувають високі вимоги. А для отримання ширини виведення 400 мм. призма повинна бути віддалена від фокальної площини на відстань не менше 400 мм. Тому капстанові пристрої як правило мають ширину вивідного формату до 400 мм.

Подальші спроби збільшити формат експонування завдяки збільшенню кута розгортки променя та розширенню механізму протягування плівки призводять до різкого падіння якості фотоформ. Для досягнення тієї якості, якої чекають користувачі від ФНА формату А2, необхідне використання додаткових конструктивних рішень для забезпечення точного позиціонування плівки, що пересувається, точного фокусування лазерного променя тощо. Використання високоточної механіки замість простого збільшення розмірів є причиною значної різниці цін на ролеві фотонабірні автомати формату А2 та ФНА менших форматів.

### *Розглянемо показники критеріїв якості ФНА.*

По-перше, це *базова технологія*.

По-друге, це *дозвіл, кількість градацій сірого та лініатура*, пов'язані певною пропорцією (лініатура дорівнює дозволу, поділеному на двійковий логарифм тієї кількості градацій сірого, що може бути передана однією растровою чарункою).

Так, пристрої *Heidelberg Prepress* забезпечують виведення традиційного растру з найвищою можливою лініатурою, не кажучи про спеціальні можливості алгоритмів растрування *Diamond Screening*, при повторюваності в 5 мкм.

По-третє, це **показники швидкодії ФНА**: час експонування фотоформи (залежить від швидкості пересування каретки з оптичною системою від одного краю плівки до іншого); швидкість отримання готової фотоформи після натискання на комп'ютері кнопки "PRINT"; сукупна швидкодія (визначається швидкістю растрового процесору, пропускною здатністю інтерфейсів, часом завантаження плівки з подавальної касети до барабану та її вивантаження до приймальної касети, часом обрізання, часом введення параметрів плівки, швидкістю заміни касет). За сукупністю цих показників, ФНА *Quasar* та *Hercules Pro* є одними з найпродуктивніших.

По-четверте, це **забезпечення лінійності ФНА на певному дозволі в процесі експонування**, тобто лінійності функції змінювання щільності растрової чарунки відносно проценту заповнення цієї чарунки елементарними точками (в процесі експонування кожному конкретному дозволу повинен відповідати певний розмір плями лазера, і чим вищим є дозвіл, тим меншим повинен бути цей розмір). Якщо ФНА на деякому дозволі не є лінійним, то потрібна додаткова корекція даних у растровому процесорі (це призводить до збільшення часу растрування та може погіршити якість зображення).

По-п'яте, це **можливість виключення хімічних реактивів із процесу підготування фотоформ для забезпечення екологічної чистоти цього процесу**. Так, ФНА *DrySetter* (*Heidelberg Prepress* + *Polaroid Graphics*) на базі *Hercules Pro*

вирішує цю проблему повністю завдяки особливому експону-вальному джерелу та використаним матеріалам: двошарова плівка *DryTech (Polaroid Graphics Imaging)* після експонування потрапляє до спеціалізованої системи, де шари розділяють та ламінують шар із зображенням спеціальним захисним покриттям.

По-шосте, це *можливість виключення з процесу виготовлення друкарської форми стадії копіювання для інтенсифікації видавничої праці*. Так, усі ФНА дозволяють експонувати поліестрові офсетні пластини, при цьому замість фотоформи виходить готова друкарська форма (ці матеріали підходять для друку чорно-білих або нескладних кольорових видань та мають порівняно невисоку тиражестійкість). Є спеціалізовані ФНА серії *Trend Setter (Heidelberg Prepress+CREO)*, призначені для прямого експонування металевих офсетних термопластин. Система *Gutenberg* дозволяє виготовляти пластини для високоякісного повнокольорового офсетного друку.

**Технології цифрового експонування офсетних форм.** Невпинне зростання вимог до процесу підготування поліграфічної продукції потребує постійного збільшення продуктивності поліграфічного обладнання, досягти якої можна завдяки цілеспрямованому перетворенню технологічного циклу в повністю цифровий комп'ютерний процес.

Виробники поліграфічного обладнання пропонують видавцям значну кількість концептуальних рішень, технологій та пристроїв, дозволяючи створити повністю цифровий переддрукарський цикл поліграфічного виробництва для видань будь-якого рівня складності. Але на етапі виведення фотоформ

та всіх подальших процедур (контактне копіювання офсетних пластин) виникають проблеми переведення процесу з розряду цифрових до розряду аналогових.

Повністю забезпечити цифровий тип переддрукарського циклу поліграфічного виробництва, зберігаючи аналогову форму тільки в типографських процесах, дозволяє система прямого виведення офсетних пластин із цифрової форми, обминаючи стадію виготовлення фотошаблонів та їх контактного копіювання. Але відповідні апарати серії *Gutenberg* компанії *Heidelberg Prepress*, що працюють зі світлочутливими пластинами – це громіздка та дорога техніка.

*Прогресивною течією в технології прямого експонування офсетних форм (Computer-to-Plate – СТР) є використання термочутливих пластин.* Різні матеріали, сприйнятливі до температурної експозиції, потребують різних технологічних прийомів отримання зображення на металевих офсетних пластинах, серед яких найпопулярнішими є термополімерний, термопереносний, термогібридний та термоабляційний (не потребує додаткових систем хімічної проявки) способи.

Переважна більшість виробників фотонабірного обладнання заявляють про випуск пристроїв типу СТР. Найбільш передовими є пристрої *TrendSetter (CREO + Heidelberg Prepress)*, виконані за технологією "зовнішній барабан".

Оскільки форма фіксована на барабані не вакуумною системою, а магнітною, можна експонувати пластини широкого діапазону форматів. Експонування здійснює багатопроменеве лазерне джерело з довжиною хвилі 830 нм, що може забезпечувати середньотемпературну обробку термополімерних пластин.



(400 °C) та високоенергетичний імпульсний режим для термоабляційних матеріалів.

Ця технологія дозволяє отримувати офсетні пластини незрівняно вищої якості, ніж у разі аналогового підходу, завдяки відсутності викривлення на стадії контактного копіювання форми та використанню висококонтрастних термопластин (принципово відсутній ефект крайової вуалі точки) у сполученні з прецизійним лазерним променем квадратного перерізу, виконаним за запатентованою технологією *CREO*.

*TrendSetter* із додатковими аксесурами (автозавантажувач пластин із пачки, система для додаткового експонування фотошаблонів), керований растровальною системою *Delta Technology*, є потужним інструментарієм, який можна легко інтегрувати до найскладнішого цифрового середовища, забезпечуючи найвищі показники якості кольоропередавання та растровання, гнучкості та оперативності роботи.

**Растрові процесори.** Для виведення растрових зображень до фотонабірних автоматів (ФНА) або систем цифрового друку потрібен растровий процесор – *Rasterising Image Processor (RIP)* – окремий електронний пристрій або програма та інтерфейсна карта для комп'ютеру. **Основне завдання *RIP* – перетворення файлу друку, створеного за допомогою прикладних програмних засобів, у файл растрового формату, та передача цього файлу до ФНА.**

У поліграфії традиційним стандартом на кодування даних у файлі друку стала графічна мова *Adobe PostScript* (з двома найбільш розповсюдженими алгоритмами інтерпретації – *Adobe CPSI* та *Hewlett Script*). Більшість *RIP* тривалий час працює з цим стандартом: наявність єдиного стандарту на графічні

дані полегшує роботу виробників прикладного ПЗ, забезпечуючи незалежність від вивідної апаратури при одночасному збереженні максимальної якості.

**Основні функції RIP: прийом даних від робочої станції; інтерпретація даних; кольороподіл; растрівання даних; виведення даних на фотонабірні автомати.**

**Прийом даних від робочої станції**, що здійснюється, як правило, крізь мережний інтерфейс, приводить до надходження даних до вхідної черги, асоційованої з певним набором характеристик виведення (дозвіл, лініатура, формат, кольорокорекція).

**Інтерпретація даних** складається з: обробки даних на наявність помилок, видалення прихованих елементів, підключення резидентних шрифтів тощо.

**Кольороподіл** виконується тоді, коли його не було виконано на робочій станції. Розглянемо сутність цієї технології.

Якщо вхідний файл є кольоровим композитним файлом, то кожен його елемент може відображати будь-який колір із величезної гами кольорів. На практиці для передавання цієї гами кольорів нема потреби використовувати таку ж кількість фарб. Можна взяти скінченну кількість базових кольорів, які достатньо рознесені в спектрі видимого світла, щоб з них сформувати всі інші: червоний (*Red*), зелений (*Green*) та синій (*Blue*) за умови використання адитивної моделі *RGB*; блакитний (*Cyan*), пурпурний (*Magenta*) та жовтий (*Yellow*) за умови використання субтрактивної моделі *CMYK*. У кольоровій поліграфії використовують субтрактивну модель та додатковий чорний колір для підсилення передавання темних відтінків.

*Кольороподіл – це розбиття одного композитного файлу на окремі кольороподілені файли, кількість яких відповідає кількості базових кольорів (кожен із цих файлів як правило відтворює 256 відтінків).*

*Растрівання даних, які були отримані на етапі інтерпретації та кольороподілу, є необхідним тому, що ФНА дозволяють передавати за допомогою елементарної точки тільки два відтінки: білий та чорний. Під растріванням розуміють спосіб відтворення гами відтінків шляхом об'єднання елементарних точок у групи (растрові чарунки): зі збільшенням групи зростає кількість відтворюваних відтінків, але логарифмічно падає дозвіл зображення.*

*Способи растрівання кольороподілених файлів належать до таких основних класів.*

*По-перше, це традиційний спосіб "напівтонового растру" – формування амплітудно-модульованих растрових чарунок.*

Растрові чарунки, починаючи з центру, заповнюють елементарними точками, формуючи геометричні фігури (кола, ромби, еліпси). Чим більша площа фігури, тим темніший відтінок передає растрова чарунка. У підсумку отримують растрову решітку (регулярну структуру, що повторюється), кількість вузлів якої на одиницю довжини (лініатура) характеризує ступінь деталізації надрукованого зображення. Растрові решітки різних кольорових шарів (базових кольорів) повертають на різні кути, щоб у процесі друку вони не накладалися одна на одну. Цей спосіб забезпечує правильне кольоропередавання, але має недоліки: утворення "муару" через інтер-

ференцію між решітками різних кольорових шарів; утворення "розеток" (кільцевих структур, які періодично повторюються).

*По-друге, це формування частотно-модульованих растрових чарунок, які є вільними від недоліків попереднього способу.* Простір растрової чарунки заповнюють елементарними точками за стохастичним законом. Головною є кількість точок (білих, чорних, сірих), які потрапили до чарунки. Цей спосіб забезпечує високу суб'єктивну деталізацію зображення, але теж має недоліки: для обчислення розташування елементарних точок потрібно багато часу та ресурсів комп'ютеру (цього можна уникнути, розраховуючи наново лише ті точки, відтінок яких ще не зустрічався); вимоги до калібрування друкарського пресу набагато вищі, ніж у разі використання стандартного напівтонового растру (цього можна уникнути, орієнтуючи друкарський цикл на роботу тільки зі стохастичними растрами, що виключає необхідність перекалібрування в разі змінювання типу растру).

*Ще один етап – передавання растрових полей до ФНА, де їх експонують та виводять.*

Оскільки задача управління растровим друком тісно пов'язана з управлінням потоками даних та розподілом інформації в локальних мережах, то до растрових процесорів включають відповідні опції (операції автоматичного трепінгу, організацію *OPI*-серверів на станції растрування, організацію кольоропроби з використанням тих даних, що й для ФНА, управління чергами задач друку, розподіл растрів на форматі виведення тощо). Всі ці задачі або більшість з них розв'язують растрові процесори *Heidelberg Prepress Delta Technology* та *Ultre RIP*.

**Проявлювальні процесори та технології.** Після експонування в фотонабірному автоматі, фототехнічну плівку, як і звичайну фотографічну, піддають чотириетапній обробці: проявлювання, закріплювання, промивання та сушіння.

Існує багато способів виконання цих етапів. Першим із них є ручна обробка фотоматеріалів у ванночках з розчинами. Це дешево, але для професійної роботи не підходить: продуктивність праці є надзвичайно низькою; параметри фотоформ змінюються від однієї фотоформи до іншої. **Для професійної роботи використовують проявлювальні машини – спеціальні установки, що автоматизують процес обробки фотоматеріалів.**

За технологією роботи проявлювальні машини розділяють так: автономні, що обробляють фотоматеріали, відекспоновані та вилучені з ФНА (процес типу *OFF-LINE*); такі, що з'єднані з ФНА за допомогою спеціального мосту та працюють у парі з ФНА (процес типу *ON-LINE*). Обробка матеріалу в процесах *OFF-LINE* та *ON-LINE* здійснюється в однакових проявлювальних машинах, а різниця полягає в наявності механічного інтерфейсу для підключення до ФНА. Вибір одного з процесів залежить від конкретних умов функціонування вивідної системи: якщо є декілька ФНА з різною продуктивністю, доцільно використовувати машини *OFF-LINE*; якщо потрібно отримати максимальну продуктивність та зручність роботи, обирають процес *ON-LINE*.

**Проявлювальні машини відрізняються ступенем автоматизації процесу роботи, якістю виконання системи протягування плівки, можливостями додаткового розширення, продуктивністю, що прямо залежить від обсягу смостей**

*для розчинів. Ці розбіжності впливають на вартість проявлювальних машин.*

Кольоропробне обладнання та технології. Для об'єктивної оцінки якості тиражного відбитку, до початку його виготовлення потрібно зробити пробні відбитки. У разі чорно-білого тиражу, пробний відбиток можна виконати на звичайному лазерному принтері, а потім урахувати різницю в дозволі пробного відбитку та тиражного. У разі кольорового тиражу, ідеальним варіантом є виконання пробного відбитку тими самими фарбами та обладнанням, що й тираж, але це дорого та довго. Для швидкого та недорогого виконання кольорових пробних відбитків є спеціальні кольоропробні пристрої (кольоропроби) двох класів: аналогові та цифрові.

*Виконання кольоропроби є важливою ділянкою в друккарському процесі, тому необхідно ретельно вибирати відповідне обладнання. Ідеальним варіантом є наявність як аналогової, так і цифрової кольоропроб, оскільки тоді можна досягти компромісу між якістю, оперативністю та сумісністю.*

Аналогові кольоропроби мають ряд переваг: завдяки особливостям технологічного процесу (за початкову інформацію використовують звичайні фотоформи, зображення з яких контактним способом переноситься на основу; якщо виконаний відбиток підходить за якістю, ці фотоформи використовують при виготовленні офсетних пластин для друку тиражу), пробний відбиток ідентичний тиражному за кольором та структурою растру; можуть бути використані не тільки базові СМУК-кольори, а й додаткові кольори, що дозволяє виконувати кольоропроби для нестандартних друкарських процесів. Проте

аналогова кольоропроба: є великим пристроєм; вимагає кваліфікованого оператора; не може бути використана там, де немає фотоформ, наприклад, у технології *СТР* (*Computer-to-Plate*).

В аналогових кольоропробах використовують дві основні технології роботи: "мокру" технологію (листові тонерні плівки експонують у копіювальній рамі крізь фотоформи та обробляють у рідких реактивах); "суху" технологію (листові тонерні плівки перед експонуванням накотують у ламінації на основу, засвічують та розділяють; неекспоновані місця видаляються, а експоновані – прилипають до основи; процес повторюють для кожного кольорового шару).

Цифрові кольоропробні пристрої є кольоровими принтерами, що використовують *СМУК*-тонери (у деяких випадках використовують додаткові кольори), мають велике кольорове охоплення (ширше, ніж у офсетних пресів) та працюють під управлінням такого ПЗ, що дозволяє емулювати кольорові стандарти офсетних пресів, а в ряді випадків дозволяють емулювати колір (відтінок) тиражного паперу. Цифрові кольоропроби не мають недоліків аналогових кольоропроб, але їм властиві власні недоліки: обмежений дозвіл кольорових принтерів не дозволяє емулювати форму растру (виключенням є принтери, що використовують струминну неперервну технологію друку), тому на цифрових кольоропробах в основному отримують лише емуляцію кольору; неможливість друку спеціальних простих кольорів (металізованих, високонасичених).

Для цифрових кольоропроб використовують декілька технологій кольорового друку, кожна з яких має свої переваги та недоліки: термо-сублімаційну, струменеву бульбашкову, лазерну, струменеву неперервну, твердо-чорнильну (перелік техно-

логії наведено в порядку зменшення їх розповсюженості). Термо-сублімаційна технологія забезпечує найширше кольорове охоплення та повну шкалу відтінків без растрівання (фотографічну якість) завдяки тому, що барвник у вигляді пари осаджують в одну й ту ж саму точку растру.

**Контрольно-вимірювальне та переглядове обладнання.** Для визначення якості поліграфічної продукції, на протязі всього процесу її виготовлення (починаючи від підготування інформації та закінчуючи офсетним друкарським пресом), використовують вимірювальні інструменти та пристосування, що здатні дати об'єктивно незалежну оцінку: спектро-фотометри, чорно-білі та кольорові денситометри, різноманітні лінзи та мікроскопи.

**Спектрофотометри** є інструментами для об'єктивного вимірювання кольору, які: дозволяють з високою точністю передавати спектральні характеристики кольору, на основі яких програмне середовище здатне, наприклад, підібрати аналог зі шкали *Pantone* чи розкласти простий колір на тріадні; можуть бути використані для калібрування та характеризації різних кольорових пристроїв (моніторів, принтерів, друкарських пресів тощо).

**Чорно-білі денситометри** застосовують для налаштування фотовивідної та проявлювальної техніки, в якій використовують чорно-білі матеріали (фототехнічну плівку), шляхом перевірки якості виведених плівок за такими їх параметрами, як: щільність плашок; щільність напівтонових зображень; якість растру; процент заповнення плашки.

Чорно-білі денситометри є необхідним інструментом при калібруванні фотовивідного апарату та растрового процесору,



підбиранні режиму та контролі роботи проявлювальної машини. В цьому випадку забезпечується контроль за ступенем обробки матеріалу: плівка не повинна бути перетримана або недотримана в проявлювачі при заданій температурі. Ці прилади дозволяють: вимірювати оптичну щільність надрукованої тестової таблиці (тестових плашок) у світлі, що проходить, та у відбитому світлі; вносити корегувальні зміни до ФНА, його растрового процесору та проявлювальної машини. Цим досягають точності передавання відтінків за всіма кольороподіленими формами та необхідної щільності фотограф, з метою якісного виготовлення офсетних пластин.

Денситометр працює за простим принципом. Він складається з фоточутливого елементу та логічної схеми. Джерело світла (зовнішнє або вбудоване) направляє світловий потік крізь дільницю плівки, яку піддають обробці, на фоточутливий елемент. Цей елемент приймає світловий потік та передає його характеристики на логічну схему для обробки, приведення до стандартних одиниць та відображення на вбудованому дисплеї.

На етапі контролю за процесом друку застосовують кольорові денситометри, різноманітні лінзи та мікроскопи.

*Кольорові денситометри* дозволяють контролювати якість кольору на протяжі друкування тиражу, що позбавляє від неприємних наслідків – розходження кольору на початку та під кінець тиражування. Вони вимірюють: оптичну щільність 100 %-них заливок базових кольорів, процент заповнення растру, рівень розчавлювання крапки (розтікання фарби після нанесення на папір), трепінг, баланс сірого при його друці тріадними фарбами та інші параметри.

У процесі вибору денситометру важлива його орієнтація на потрібний стандарт фарб для офсетного друку. В кольорових денситометрах повинен бути поляризаційний фільтр для уникнення різниці в показниках при вимірюванні "сухих" (готових) або "сирих" (з фарбою, що не висохла) відбитків.

*Лінзи та мікроскопи* дозволяють візуально оцінити якість формування растрової структури та помітити механічні дефекти в процесі друку (неправильний баланс барвника та розчинника, що викликає згортання фарби в краплі, призводячи до нещільної растрової точки; зсув та змазування растрових точок через нещільне або занадто щільне притискання паперу; "двоїння" точок через неповне перенесення фарби з валу на папір): це можна своєчасно помітити за допомогою мікроскопів із підсвіткою та виконати корекції на пресі.

Для досягнення максимальної якості в процесі візуального порівняння кольоропроби та тиражного відбитку, потрібно враховувати зовнішнє освітлення. Для цього застосовують *переглядові "ящики" (спеціальні ніші з каліброваним освітленням)*, а для перегляду слайдів та фотоформ використовують *переглядові столи з підсвіткою*.

Наведений огляд видавничих графічних комп'ютерних технологій складено на основі авторського аналізу, систематизації та узагальнення матеріалів журналів "Компьютер пресс" та "Chip".

## **Питання та завдання для самоперевірки**

1. Що являють собою фотонабірні автомати та технології?
2. Сформулюйте поняття кольороподілу.

3. Що таке колоропроба та як її здійснюють?

4. Розробіть ваші пропозиції по опублікуванню *Web*-матеріалів, створених у процесі виконання завдань для самоперевірки до попередніх глав розд.2, у вигляді друкованого рекламного буклету.

5. Ознайомтеся за матеріалами комп'ютерної періодики з провідним досвідом вітчизняних та зарубіжних виробництв та установ, останніми дослідженнями вчених, тенденціями розвитку в галузі сучасних комп'ютерних технологій. Проведіть аналіз тієї практики застосування комп'ютерних графічних технологій, що склалася на підприємствах, у фірмах та організаціях за місцем вашого проживання, навчання або роботи, та запропонуйте шляхи подальшого вдосконалювання цієї практики, ретельно обґрунтувавши свої пропозиції.

## Навчальний практикум до гл.9-10

**1-а група завдань.** У процесі виконання цих завдань необхідно створити на базі однієї з мов програмування високого рівня (*Pascal, Delphi, C++, C++ Builder, Visual C++, Java, Visual J++, Visual Basic, Visual Fortran* тощо) та опублікувати на *Web*-сторінці Internet навчальну (або просто демонстраційну) мульти- та гіпермедійну програму, що показує графічні можливості цієї мови програмування за одним із вказаних нижче типових напрямків їх практичної реалізації (попередньо ретельно обґрунтуйте вибір інструментарію та технологій розробки ПЗ, звернувши особливу увагу на застосування асемблерних

вставок для підвищення ефективності роботи програмного продукту).

**Завдання 1.** Побудова ділової графіки на прикладі комп'ютерної візуалізації відомостей про результати: а) залікової сесії; б) екзаменаційної сесії.

**Завдання 2.** Побудова графіків математичних функцій на прикладі комп'ютерної візуалізації: а) поверхні Келі, що описується функцією  $y = z^3 - xy$  ( $-2 < x, y < 2$ ); б) коноїда Плюкера, що описується рівнянням  $wx^2 - y^2z = 0$  ( $-2 < x, y < 2, w$  – параметр).

**Завдання 3.** Побудова схематичних зображень на прикладі комп'ютерної візуалізації: а) блок-схем алгоритмів; б) структурних схем (для різних класів архітектур комп'ютеру, топологій мереж, корпораційних ієрархічних структур); в) композицій декількох геометричних 2D/3D-фігур, довільно розташованих одна відносно одної.

**Завдання 4.** Створення високоякісної анімації на прикладі розв'язування задач комп'ютерної візуалізації: а) процесів виконання базової системи команд виробничими роботами-маніпуляторами горизонтального та вертикального руху; б) просторово-часової динаміки ансамблю декількох 3D-об'єктів, кожен із яких пересувається певною траєкторією; в) одного з фундаментальних фізичних законів, явищ або процесів (наприклад, броунівського руху, падіння тіла під дією сили тяжіння, руху снаряду, ковзання або котіння предмету тощо); г) ходу проведення та результатів хімічного експерименту; д) руху та різноманітних перетворень розміру та форми графічного тексту.

**Завдання 5.** Реалізація мультимедіа-, гіпермедіа- та Web-графіки на прикладі створення та публікації на Web-сторінці *Internet* невеликого електронного довідника чи енциклопедії

про вашу спеціальність (кафедру, факультету, вищий навчальний заклад), що буде містити текст, аудіо, графіку, фото, відео.

*Примітка.* Можна також обрати одне з тих завдань до гл.1-8, яке ви ще не виконували, та виконати його згідно з вимогами, поставленими до першої групи завдань до гл.9-10.

**2-а група завдань.** У процесі виконання цих завдань необхідно розробити на базі мови програмування *Assembler* або на базі однієї з мов програмування високого рівня, із застосуванням відповідних асемблерних вставок, анімаційну графічну програму, що виконує одну з запропонованих нижче функцій, ретельно обґрунтувавши вибір базових методів та технологій анімації.

**Завдання 6.** Пересуває вздовж ігрового поля, що плавно прокручується в горизонтальному та вертикальному напрямках: а) декілька стрибаючих м'ячів; б) рекламну повітряну кулю.

**Завдання 7.** Одночасно пересуває вздовж екрану  $K$  довільних графічних об'єктів-шаблонів розміром  $N \times N$  (наприклад,  $K = 12$ ,  $N = 32$ ), реалізуючи анімацію без мерехтіння, з ефектом прозорості та з чотирма рівнями глибини зображення.

**Завдання 8.** Здійснює в верхній частині екрану анімацію кульки, що відстрибує від меж ігрового поля та від інших статичних і динамічних об'єктів, які знаходяться на ньому, тоді як у нижній статичній області екрану відображуються результати підрахунку кількості зіткнень.

**Завдання 9.** Здійснює анімацію  $K$  зображень ігрових фішок-шаблонів розміром  $N \times N$  (наприклад,  $K = 10$ ,  $N = 11$ ), що рухаються вздовж фіксованих траєкторій ігрового поля, які перетинаються.

**Завдання 10.** Рисує статичний фон із зображенням мальовничого пейзажу (наприклад, гори, сонця, долини, річки та дому з димком із труби) та здійснює швидко анімацію  $N$  повітряних зміїв (наприклад,  $N = 20$ ), які, рухаючися за довільними траєкторіями, гармонійно вписуються у фон, нічого не затирають та безпроблемно перетинаються між собою.

**Завдання 11.** Виконує швидкодіючу анімацію в 3D-просторі в реальному часі заданих геометричних фігур –  $K$  багатокутників,  $M$  кубів та  $N$  довільних багатогранників (наприклад,  $K = 6$ ,  $M = 12$ ,  $N = 3$ , а за багатогранники взято сферичні об'єкти – 72-гранні м'ячі певних розмірів та з певною кількістю смуг), грані яких є: а) однокольоровими з суцільними кольорами (однотонними); б) однокольоровими з 3D-відтінченнями поверхні (імітацією її освітленості); в) різнокольоровими з накладанням текстури (візерунків, графічного тексту, будь-яких інших графічних зображень).

**Завдання 12.** Реалізує драйвер символного виведення в позначені позиції екрану без накладання на фон графічного тексту (текстових символів, які відповідають певним шаблонам  $N \times N$  у графічному режимі) та анімацію цього тексту (динамічне змінювання його розташування, орієнтації в просторі, розмірів та форми) для таких випадків: а) окремих літер; б) невеликих блоків тексту; в) великих блоків тексту. Зберігає в  $N$ -кольоровому режимі (наприклад,  $N = 16$ ) знімок (копію) екрану в файлі з заданим ім'ям, очищує екран, а потім відновлює графічний екран із цього файлу.

**Завдання 13.** Підтримує систему текстових меню в графічному режимі.

**Завдання 14.** Підтримує діалогове конструювання символів та графічних образів на екрані дисплею.

**Завдання 15.** Виконує кольорову анімацію набору різнокольорових концентричних кіл таким чином, щоб за рахунок перетікання кольорів та плавного змінювання розміру кіл створити ілюзію руху.

**Завдання 16.** Встановлює кольори, що плавно перетікають один в інший, та заповнює екран зірками так, що їх блищення плавно переходить від однієї зірки до іншої.

**Завдання 17.** Здійснює динамічне чергування кольорів набору різнокольорових смуг, імітуючи: а) плавний рух; б) захід сонця; в) течію річки.

**Завдання 18.** Здійснює динамічне змінювання кольорів всередині кадрів та від кадру до кадру (згідно з обстановкою), створюючи реалістичні ефекти для 3D-анімації.

**Завдання 19.** Здійснює наближення реальних прямих і кривих ліній та їх відрізків, контурів геометричних примітивів, зафарбованих геометричних фігур за допомогою наборів пікселів і відрізків постійної та змінної довжини зі згладжуванням та без згладжування драбинного ефекту.

**Завдання 20.** Здійснює визначення взаємного розташування статичних і рухомих об'єктів на екрані, визначає перешкоди на їх шляху та виконує зафарбовування статичних і рухомих фігур із заданими межами.

**Завдання 21.** Здійснює редагування відеопам'яті, що містить дані, виконуючи запис заданих користувачем фону та зображень (суцільних однокольорових, смугастих, довільних різнокольорових) у відеопам'ять з наявним у ній іншим кольоровим фоном та зображеннями (з кольоровим накладанням цих

зображень згідно з певною логічною умовою та без такого накладання).

**Завдання 22.** Рисує на фоні горизонтальних смуг із певним шаблоном заповнювання вертикальні смуги, заповнені набором із декількох візерунків, використовуючи логічні функції, що комбінують зображення з фоном (*AND*, *OR*, *XOR*, без змін).

**Завдання 23.** Рисує динамічні кольорові візерунки за допомогою радіальних багатокольорових ліній.

**Завдання 24.** Послідовно виводить на екран: бордюр певного кольору та радіальні кольорові смуги  $N$  кольорів (наприклад,  $N=16$ ), при цьому після кожного натискання на вказану клавішу здійснюється виведення нового набору кольорів та змінювання положення смуг; зафарбовані прямокутники, що динамічно змінюють всі свої параметри.

**Завдання 25.** Перевіряє ідентичність зображеного на екрані динамічного графічного об'єкту запропонованому шаблону.

*Примітка.* Можна також обрати одне з тих завдань до гл.1-8, яке ви ще не виконували, та виконати його згідно з вимогами, поставленими до другої групи завдань до гл.9-10.



## Висновки до розд.2

Апаратне та програмне забезпечення комп'ютерної графіки неперервно вдосконалюються, а тому неможливо бути фахівцем у цій галузі, не поновлюючи постійно своїх знань шляхом вивчення періодики – комп'ютерних газет, журналів, матеріалів *Web*-сторінок виробників графічного АЗ та ПЗ. Проте і в цій динамічній галузі існують такі непорушні засади та усталені концепції розвитку, що не похитнуться та залишаться прогресивними якнайменше на протязі найближчого п'ятиріччя. Висвітленню цих питань і було присвячено гл.4-5 розд.2 цього навчального посібника.

А саме, були розглянуті: головна мета та особливості міжнародної діяльності зі стандартизації в комп'ютерній графіці; коротка характеристика основоположних графічних стандартів та найпопулярніших графічних файлових форматів; функції та взаємодія базового та прикладного ПЗ комп'ютерної графіки, роль стандартних *API* для роботи з графікою; склад, функції, взаємодія, ієрархічна класифікація основних різновидів апаратного забезпечення графічної підсистеми ЕОМ.

Чільне місце в комп'ютерній графіці посідають такі прогресивні технології, як: технології реалістичної 3D-графіки на базі спеціалізованих 3D-карт (для користувачів ПК) та потужних професійних графічних робочих станцій; технології цифрового фото та відео, мультимедійної та гіпермедійної графіки, ефективного подання графіки на *Web*-сторінках *Internet*; технології візуалізації та анімації графічних зображень на основі використання стандартних бібліотек візуальних компо-

ментів мов програмування високого рівня; технології швидкого синтезу та анімації графічних зображень на базі інструментарію мови програмування *Assembler*; цифрові видавничі технології, що забезпечують ефективну роботу з документами, які містять графічні ілюстрації (див. гл.6-10 розд.2).

---

## РОЗДІЛ 3. ЗАСТОСУВАННЯ КОМП'ЮТЕРНОЇ ГРАФІКИ

---

### ГЛАВА 11. КОМП'ЮТЕРНА ГРАФІКА У ТВОРЧІЙ ТА ДІЛОВІЙ СФЕРАХ

#### 11.1. Основні поняття про комп'ютерні мистецтво, дизайн, рекламу та поліграфію

В основу реалізації за допомогою ЕОМ художніх та дизайнерських проєктів, створення реклами та випуску друкованої продукції для сфер творчої та ділової діяльності, покладено концепції, інструментарій та технології двох фундаментальних розділів сучасної комп'ютерної графіки та дизайну – ілюстраційної та ділової графіки [28, 30, 63, 91, 108, 171].

*Ілюстраційну комп'ютерну графіку призначено для створення комп'ютерних зображень, що відіграють роль статичного чи динамічного ілюстраційного матеріалу (художніх малюнків, фотографій, репродукцій витворів мистецтва, слайдів, відеороликів, ілюстрованих видань, технічних рисунків, умовних схем, ескізів, географічних карт тощо) [58, 130, 165].*

До засобів розробки ілюстраційної комп'ютерної графіки належать всі ті застосування, що дозволяють створювати графічні об'єкти, представляючи комп'ютерну графіку як галузь мистецтва.

Основними та тісно пов'язаними між собою відгалуженнями ілюстраційної комп'ютерної графіки є комп'ютерні мистецтво, дизайн, реклама та поліграфічна діяльність [136-139].

Основними класами користувачів сучасних систем ілюстраційної комп'ютерної графіки є художники, фотографи, дизайнери, журналісти, відеооператори, видавництва та друкарні, криміналісти.

Ілюстраційна комп'ютерна графіка має практичні застосування в усіх галузях діяльності: промисловості, сільському господарстві, бізнесі, менеджменті, підприємстві, торгівлі, маркетингу, правоохоронній діяльності, телебаченні, пресі, поліграфії, освіті, кінопромисловості, шоу-бізнесі тощо [44, 45, 58, 73, 76]

**3 точки зору внутрішнього подання, виділяють два класи об'єктів ілюстраційної комп'ютерної графіки:**

– **зображення на основі нерегулярних структур**, які створюють методом вільного рисування (ці зображення належать до **растрової графіки**, оскільки складаються з монохромних або кольорових точок растру);

– **зображення на основі регулярних структур**, які визначають на основі елементарних геометричних понять, таких як прямокутники, кола, трикутники та інші (ці зображення належать до **векторної (координатної) графіки**, оскільки елементарні об'єкти можна однозначно задати векторами координат опорних точок та деякими додатковими параметрами).

Розробники систем ілюстраційної комп'ютерної графіки направляють зусилля на те, щоб графічні об'єкти можна було легко формувати та перетворювати (подібно масивам чисел за допомогою електронних таблиць або текстам за допомогою

систем підготування текстів), подавати в найреалістичнішому вигляді, демонструвати в динаміці, найкомпактніше та найбезпечніше зберігати на носіях інформації, найшвидше надсилати каналами зв'язку, відображати з достатнім рівнем якості.

*Користувач систем ілюстраційної комп'ютерної графіки повинен мати можливість виконувати такі типові дії:*

– вводити зображення з будь-яких електронних та традиційних носіїв інформації;

– будувати власні зображення, зокрема шляхом їх компонування з інших зображень, поєднання з текстами, числами, звуком, фото та відео, змінювання кольору, спорядження гіперпосиланнями тощо;

– редагувати зображення, застосовуючи стандартні функції корегування, площинні та просторові трансформатори, модифікатори, деформатори, спецефекти тощо;

– моделювати розташування, форму та властивості зображень;

– анімувати та морфувати зображення (відтворювати їх у динаміці та природній пластиці руху);

– створювати ефекти об'ємності зображення, його реалістичного освітлення, природної структури та текстури поверхонь, атмосферні ефекти, ефекти перспективи тощо та формувати з їх допомогою реалістичні, динамічні 3D-сцени;

– стискувати та оптимізувати зображення (для економнішого їх зберігання та швидшого передавання мережами), за умови збереження необхідного рівня їх якості;

– зберігати зображення на носіях інформації;

– передавати зображення засобами зв'язку (внутрішнього комп'ютерного та міжкомп'ютерного);

– здійснювати адміністрування графічних зображень на всіх етапах їх життєвого циклу (особливо організацію їх захисту від несанкціонованого доступу).

***До систем ілюстраційної комп'ютерної графіки належать графічні пакети, призначені для:***

– оцифрування традиційного ілюстраційного матеріалу;

– фільтрації графічних зображень (для їх імпорту, експорту, оптимізації за обсягом та якістю);

– побудови та обробки растрових зображень, фотодизайну (ретушування фотографій, створення фотореалістичних колажів);

– роботи з векторними об'єктами та їх складними художніми композиціями;

– побудови тривимірних наочних, реалістичних та динамічних моделей об'єктів, сцен та віртуальних світів;

– анімації (мультиплікації) та морфіngu (змінювання форми) зображень;

– створення та обробки інтерактивного цифрового відео;

– розробки мультимедійних та гіпермедійних зображень;

– підготування зображень для *World Wide Web*;

– комп'ютерної поліграфії (верстання та видавництва);

– розпізнавання графічних образів (штрих-кодів у торгівлі, відбитків пальців та сітківки ока в криміналістиці тощо).

***У цілому системи ілюстраційної комп'ютерної графіки розділяють на:***

– потужні універсальні графічні пакети, що в певній мірі підтримують більшу частину з перелічених вище функцій, але

максимально повно та професійно реалізують тільки одну-дві з цих функцій – відповідно до того практичного призначення графічного пакету, що домінує;

– вузькоспеціалізовані графічні пакети, націлені на виконання тільки декількох із перелічених функцій.

*У кожному з цих класів виділяють професійні та любительські графічні пакети.*

*Якщо брати детальнішу класифікацію, то на практиці виділяють такі базові класи систем ілюстраційної комп'ютерної графіки:*

- *plug-ins*-програми (фільтри) для графічних пакетів;
- графічні редактори растрової та векторної ілюстраційної комп'ютерної графіки;
- системи для 3D-моделювання, анімації, морфінгу, наочної візуалізації, дизайну та відеодизайну;
- системи мультимедіа-дизайну;
- системи *Internet*-графіки (*Web*-дизайну);
- видавничі системи;
- системи розпізнавання графічних образів.

*Ділову комп'ютерну графіку призначено для наочного графічного відображення даних, які являють собою ділову інформацію та зберігаються в електронних таблицях та базах даних.*

*Типовими формами графічного подання даних у системах ділової комп'ютерної графіки є такі діаграми: стовпчикові (гістограми), лінійчасті, конічні, циліндричні, пірамідальні, колові, кільцеві, крапкові, бульбашкові, графіки, з областями, поверхні, пелюсткові, біржеві, часові (графіки Ганта), структурні схеми [4, 44, 58, 73, 76, 136, 138, 144].*

*Основними класами програмних систем, призначених для створення ділової комп'ютерної графіки на професійному рівні, є:*

– електронні таблиці та спеціалізовані додатки до СУБД, призначені для графічної візуалізації інформації з баз даних (у складі офісних пакетів чи окремо), типовими представниками яких є *Excel*, *QuattroPro*, *Lotus1-2-3*, *SuperCalc*, *BoeingGraph*, *FoxGraph*;

– програми створення інтерактивних мультимедійних презентаційних роликів, типовими представниками яких є *PowerPoint* та *HyperMethod*.

***Ще декілька класів програмних систем, не спеціалізуючися виключно на діловій комп'ютерній графіці, дозволяють створювати її в рамках основних функцій цих систем:***

– текстові процесори (*Word*, "Лексикон-*XL*" та інші);

– інструментальні середовища розробника сучасних компонентних об'єктно-орієнтованих візуальних мов програмування високого рівня (*Delphi*, *C++Builder*, *Visual C++*, *Java*, *Visual J++* та інших);

– наукові, ілюстраційні та інженерні графічні редактори.

***Професійні пакети ділової комп'ютерної графіки дозволяють [4, 58, 76, 138, 144]:***

– одразу після введення початкових даних та вибору способу їх подання, отримувати в заданих областях екрану ПК графічні двовимірні та тривимірні зображення, супроводжені текстовими коментарями та розміткою;

– варіювати розміри та відносне розташування окремих складових частин зображення, здійснювати геометричні транс-



формації елементів цього зображення (дзеркальне відбиття, обертання, витягування вздовж осі, накладання тощо);

– доповнювати зображення декоративними елементами зі спеціальної бібліотеки, підготованої професійними програмістами та дизайнерами;

– створювати динамічну, інтерактивну, мультимедійну та гіпермедійну ділову графіку з анімацією та спецефектами, можливістю розташування її на *Web*-сторінках глобальної мережі *Internet* та передавання цією мережею;

– здійснювати обмін (імпорт/експорт) інформацією (електронними таблицями, базами даних, текстами, графікою тощо) з іншими програмними системами.

Для реалізації ділової комп'ютерної графіки розроблено багато систем. Але більшість користувачів віддають перевагу роботі з програмами, переліченими вище: вони надають гарні можливості в процесі роботи над візуалізацією різних типів даних, є простими в користуванні. Свою багатофункціональність у роботі з діловою графікою демонструють офісні пакети. Лідером у цій галузі вважають компанію *Microsoft* та її пакет *Office*, що має велику кількість прихильників. Популярністю користуються складові частини цього пакету *Excel*, *PowerPoint* та *Word*.

Коротко розглянемо характеристику найбільш розповсюджених форм графічного подання даних у діловій комп'ютерній графіці.

**Гістограма.** Розмірність: 2D, 3D. Різновиди: об'ємна, з накопичуванням, нормована на 100 %. Опис: стовпчикова діаграма з вертикальною орієнтацією стовпчиків, тобто дані відображають у вигляді ряду вертикальних смуг або брусків (парале-

лепіпедів). *Застосування*: найбільш розповсюджений тип діаграм, який використовується по умовчанняю.

**Лінійчаста діаграма.** *Розмірність*: 2D, 3D. *Різновиди*: об'ємна, з накопичуванням, нормована на 100 %. *Опис*: діаграма з горизонтальною орієнтацією стовпчиків. *Застосування*: при порівнянні величин за один період часу; коли підписи категорій довгі.

**Конічна, циліндрична та пірамідальна діаграми.** *Розмірність*: 3D. *Різновиди*: об'ємна, з накопичуванням, нормована на 100 %. *Опис*: привабливі 3D-варіанти гістограми та лінійчастої діаграми. *Застосування*: аналогічно звичайним стовпчиковим діаграмам, у тих випадках, коли потрібен вишуканий дизайн.

**Колова діаграма.** *Розмірність*: 2D, 3D. *Різновиди*: пласка, об'ємна, розрізана, вторинна. *Опис*: дані подають у вигляді секторів кола чи диска, розмір яких є пропорційним внеску даних до загальної суми. *Застосування*: відображає один ряд чи категорію даних, наочно демонструючи те, скільки процентів складає кожна точка даних від загальної суми.

**Кільцева діаграма (колова гістограма).** *Розмірність*: 2D, 3D. *Різновиди*: пласка, об'ємна, розрізана, вторинна. *Опис*: схожа на колову діаграму, але дані відображають у вигляді секторів концентричних кілець. *Застосування*: відображає процент від загальної суми для декількох рядів.

**Крапкова діаграма.** *Розмірність*: 2D. *Різновиди*: поодинокі крапки; крапки, з'єднані відрізками чи згладжувальними лініями. *Опис*: для кожної точки даних відображають маркери у вигляді крапок певної форми та розміру. *Застосування*: для відображення розкиду даних (часто використовується в наукових працях).

**Бульбашкова діаграма.** Розмірність: 2D, 3D. Різновиди: звичайна, об'ємна. *Опис:* нагадує крапкову діаграму, але крапковими мітками є зображення бульбашок певного розміру (чим більше значення, тим більша бульбашка). *Застосування:* для відображення розкиду даних з вишуканим дизайном.

**Діаграма-графік.** Розмірність: 2D, 3D. Різновиди: об'ємна, з накопичуванням. *Опис:* дані відображають у вигляді точок, які з'єднано лініями чи смугами. *Застосування:* найчастіше – для відображення змін даних у часі.

**Діаграма з областями.** Розмірність: 2D, 3D. Різновиди: об'ємна, з накопичуванням, нормована. *Опис:* схожа на графік, але область під лінією зафарбовано. *Застосування:* для відображення великої кількості точок даних.

**Діаграма-поверхня.** Розмірність: 3D. Різновиди: звичайна, об'ємна, контур. *Опис:* подібно топографічним картам, відображає узвишшя та западини для сукупностей даних. *Застосування:* для відображення великої кількості даних.

**Пелюсткова діаграма.** Розмірність: 2D. Різновиди: звичайна, з полями. *Опис:* вісь значень подано за допомогою променів, які виходять із загального центру, а точки даних з'єднані відрізками, утворюючи структуру, що нагадує павутиння. *Застосування:* переважно – для подання інформації, що характеризує природні явища та процеси.

**Біржева діаграма.** Розмірність: 2D. Різновиди: для набору з трьох значень курсів (найвищий, найнижчий, закриття), для двох наборів з чотирьох значень курсів ("відкриття, найвищий, найнижчий, закриття" або "обсяг, найвищий, найнижчий, закриття"); для набору з п'яти значень курсів (обсяг, відкриття, найвищий, найнижчий, закриття). *Опис:* є одним із різновидів

гістограми, але початкові дані повинні бути розташовані в певному порядку. *Застосування*: для наочного відображення змін у курсі біржових цін.

**Часова діаграма (графік Ганта).** *Розмірність*: 2D, 3D. *Різновиди*: пласка, об'ємна. *Опис*: подається в вигляді ланцюжків горизонтальних ліній чи прямокутних паралелепіпедів у прямокутній системі координат, наприклад, на осі абсцис відкладають одиниці часу, а рівномірно вздовж осі ординат – назви операцій чи процесів (довжину та розташування ліній визначають тривалість та час початку операцій). *Застосування*: для зображення послідовності взаємопов'язаних операцій чи процесів певної тривалості (закінчення одних операцій викликає початок інших, а деякі можуть виконуватись одночасно); хід графіка дозволяє прослідкувати черговість виконання операцій та наближення до кінцевої мети (можна зображати течію в часі та взаємозв'язок декількох динамічних процесів).

**Структурна схема.** *Розмірність*: 2D, 3D. *Різновиди*: пласка, об'ємна. *Опис*: на вигляд нагадує дерево; блоки (листки дерева) відповідають об'єктам (найчастіше їх зображують прямокутниками або колами, всередині яких розташовують текст); блоки з'єднані лініями (гілками), що відповідають зв'язкам між об'єктами. *Застосування*: побудова структурних схем складних об'єктів та систем (архітектури комп'ютеру, структури підприємства тощо).

**Розповсюдженими різновидами описаних діаграм є: об'ємні діаграми** (дуже популярні, оскільки виглядають краще за пласкі, але потребують налаштування обертання, піднімання та перспективи); **діаграми з накопичуванням** (використовуються для відображення підсумкових даних); **діаграми, що**

*нормовано на 100 %* (використовуються для відображення відсоткового вмісту кожної точки даних у загальній сумі); *розрізані діаграми* (використовуються для відображення однієї групи точок даних окремо від інших для акцентування на ній уваги, є різновидом колових та кільцевих діаграм); *вторинні діаграми* (застосовуються для виділення частини елементів із загальної суми та подання їх в окремій коловій діаграмі чи гістограмі); *комбіновані діаграми* (наприклад, колова та вторинна); *спрощені діаграми* (містять тільки крапки чи контур, не мають полів тощо) [4, 76, 144].

У переважній більшості випадків, успішність практичного втілення ілюстраційної та ділової комп'ютерної графіки та дизайну тісно пов'язана з питаннями комп'ютерного верстання та видавництва на традиційних та електронних носіях інформації.

Розглянемо розподіл функцій між основними класами користувачів комп'ютерних поліграфічних технологій, де відбувається сильна конкуренція.

*Арт-бюро та дизайн-бюро* займаються додрукарським підготуванням та опанували більшу частину його процесів: художники давно використовують ПЕОМ для графічного дизайну друкованих та презентаційних матеріалів, тому природним продовженням є їх участь в підготуванні своїх матеріалів до друку.

*Підприємства додрукарського обслуговування* (бюро додрукарського підготування) займаються: художньою творчістю та дизайном (пропонують замовникам комплексне обслуговування, зокрема – розробку проектів та підтримку презентацій);

безпосередньо друком (встановлюють у себе кольорові цифрові принтери та офсетні машини).

**Типографії** виконують додрукарське підготування (вони займаються додрукарськими процесами, оскільки впроваджують у себе комп'ютерні технології – переходять на СТР та друкарські машини з можливостями безпосереднього виведення зображень з комп'ютеру), а також випускають електронні видання (за додаткову платню пропонують клієнтам поширений набір послуг).

**Фахівці з кольороподілу** розширюють асортимент послуг, виконуючи ще й функції сервісних бюро.

**Сервісні бюро** також розширюють асортимент послуг, виконуючи кольороподіл.

**На ринку ПЗ конкурують три програми верстання: Adobe PageMaker, Corel Ventura та QuarkXpress.**

Вони є найбільш популярними та доступними видавничими системами для ПК, дозволяючи без зусиль верстати книжки, брошури, газети, бланки та інші видання, включаючи професійне розбиття на колонки, обтікання графіки текстом, брошурування та інші ефекти [28,91,130, 138,171].

З відзивів користувачів, *Adobe PageMaker* є найбільш розповсюдженим, застосовується для обробки малих документів, надає ідеальний кольороподіл та ювелірну точність верстання.

*Ventura* та *QuarkXpress* застосовуються для більш громіздких документів та книг, у них спостерігаються труднощі з правильним кольороподілом. Але з виходом нових версій цих програм, спостерігається тенденція до вирівнювання їх можливостей.

Відзначимо також видавничі комп'ютерні системи *Adobe FrameMaker* (один із найпотужніших інструментів для високо-професійного верстання), *Adobe InDesign* (прогресивна система верстання, що акумулювала в собі переваги *Adobe PageMaker* та *QuarkXpress*), *Design Intelligence iPublish* (недорога альтернатива пакетам типу *Adobe PageMaker* та *QuarkXpress*) [25, 63, 77, 137-139, 147].

### Питання для самоперевірки

1. Дайте визначення ілюстраційній та діловій комп'ютерній графіці, охарактеризуйте напрямки їх практичного застосування.
2. Наведіть класифікацію та опишіть функціональні можливості систем ілюстраційної та ділової комп'ютерної графіки.
3. Назвіть найбільш розповсюджені форми графічного подання даних у системах ділової комп'ютерної графіки.
4. Опишіть особливості розподілу функцій між основними класами користувачів комп'ютерних поліграфічних технологій.
5. Охарактеризуйте системи комп'ютерної поліграфії.

## 11.2. Формування растрового та векторного ілюстраційного матеріалу

Технологія професійних растрових ілюстраційних графічних редакторів. Основними функціями растрових ілюстраційних графічних редакторів (програм обробки растрових зображень) є створення нових та редагування існуючих растрових зображень, фотодизайн.

Основними класами їх користувачів є художники, фотографи, дизайнери, розробники реклами, ілюстратори поліграфічної продукції, журналісти.

Оскільки в програмах обробки растрових зображень мають потребу не тільки професіонали, а й ті, хто хоче зробити любительський рисунок або фотографію, виділяють **професійні та любительські програми обробки растрових зображень**.

Растрові графічні редактори оцінюють з точки зору забезпеченості підтримки користувача, розвиненості функцій та вимогливості до апаратних засобів (оточення).

**Найбільш популярними системами растрової ілюстраційної комп'ютерної графіки професійного класу є графічні пакети Adobe Photoshop та MetaCreations Painter.**

**Розповсюдженими програмами цього класу є Corel PhotoPaint, Macromedia XRes, Micrografx Picture Publisher, Adobe Image Ready, Fractal Design Painter, Macromedia FreeHand, Fauve Matisse та інші.**

Програми професійного класу є складнішими за любительські. Вони пропонують більше тонких налаштувань для контрастності та кольорової палітри – функції кольорової моде-



лі *СМУК*, яких потребують професіонали на стадії підготування до друку. Розкладання кольорів *RGB*-зображень на *СМУК*-компоненти є необхідним, коли зображення для офсетного друку видають одразу до типографії.

Такі професійні графічні пакети, як *XRes*, *Photoshop*, *Painter*, *PhotoPaint*, *Picture Publisher* та інші, роблять можливим гнучкий монтаж. Це стосується і пакету *Photo Impact*, який не потрапив до класу професійних програм через відсутність повної підтримки кольорової моделі *СМУК*.

Велика кількість різних растрових графічних редакторів пояснюється тим, що оптимальної програми обробки зображень не існує. Навіть у класі "для професіоналів" *PhotoShop* здається універсальною програмою тільки доти, доки здійснюється корегування та комбінування наявних фото.

***Технологію роботи в середовищі будь-якого растрового графічного редактору можна умовно розділити на три етапи.***

*Етап 1.* Користувачу надається можливість попіксельно створити нове зображення або відкрити та відредагувати існуюче зображення чи шаблон (під шаблоном розуміють набір стилів, які застосовуються одночасно та визначають вигляд ілюстрації чи документу в цілому), використовуючи для цього такі віртуальні (традиційні та оригінальні) засоби:

– інструменти (олівець, пензель, перо, аерограф, штамп, відерце з заливкою, валик, лінійка, гумка, шпатель, губка, ласо, "чарівна паличка" тощо);

– матеріали (папір, полотно, вугілля, крейда, різноманітні фарби, вода, малі частинки для насипання мозаїки, рідкий метал та інші);

– техніки та прийоми рисування (писання олійною фарбою, пастеллю, тушшю, аквареллю, гуашшю чи в техніці "батік", створення вітражів, застосування розмиття, тобто зсуву сирої фарби, вичавлювання, випалювання, розжарювання або охолодження тощо).

*Звичайно графічні редактори дозволяють відкривати графічні файли великої кількості форматів (за це відповідають спеціальні фільтри).*

*Етап 2.* Редагування та створення спецефектів, використовуючи компоненти редактору (як правило, можна працювати з кожним об'єктом незалежно та відмінити будь-які помилкові дії).

*Етап 3.* Оптимізація та збереження зображення в файлі.

У процесі роботи з растровими зображеннями, якість результатів та розмір файлу залежать від параметрів дозволу (кількості елементів інформації, що міститься в файлі зображення, та рівня деталізації, забезпеченого пристроєм введення, виведення чи відображення), обраних на початковому етапі.

Робота з растровими зображеннями вимагає певного розрахунку, оскільки обраний дозвіл зображення звичайно зберігається разом із файлом: чи будете ви друкувати растровий файл на лазерному принтері з дозволом 600 точок/дюйм, чи на фотонабірному автоматі з дозволом 1270 точок/дюйм, його буде надруковано з тим дозволом, який було встановлено при створенні зображення (якщо він не вище дозволу друкувального пристрою).

Для того, щоб підсумкове зображення виглядало таким, як на екрані, потрібно перед початком роботи настроїти дозвіл зображення згідно з дозволом пристроїв виведення.

Збільшення розміру растрового зображення виконується шляхом збільшення кожного його елемента, що огрубляє лінії та форми.

Зменшення розміру растрового зображення теж викривляє його початковий вигляд, оскільки частина елементів видаляється.

Оскільки растрове зображення створене з впорядковано розташованих пікселів, не можна вільно маніпулювати його складовими частинами, наприклад, довільно пересувати їх.

Оскільки кожен елемент растрового зображення має власний колір, то можна створювати фотографічні ефекти, такі як затінення та підсилення кольору, змінюючи обрану область поелементно.

Програми редагування фотозображень працюють з растровими зображеннями через можливість корегувати малі деталі, здійснювати значні зміни та підсилювати різноманітні ефекти.

Програми рисунка, обробки зображень та сканування створюють растрові зображення там, де необхідно відобразити неперервне змінювання тону.

З растрового та векторного форматів, кращу якість відображення фактури та текстури забезпечують растрові зображення.

Оглянемо функціональні можливості, переваги та недоліки провідних растрових графічних редакторів.

*Фактичним стандартом серед програм створення та обробки растрової графіки є пакет Adobe Photoshop. Він є лідером на платформах Macintosh та Windows серед програм ретушування растрових зображень та фотодизайну.*

*Photoshop* підтримує функції імпорту/експорту, створення (рисування), редагування, вклеювання, маскування, пересування змісту, обробки за допомогою численних спеціальних ефектів, підготування для *Web* та друку фотореалістичних зображень, а також монтажу композицій цих зображень та створення фотореалістичних колажів за допомогою таких технологій [50, 71, 92, 110, 113, 131, 145]:

- імітації природних інструментів, матеріалів і технік рисування та корегування;
- виділення зображення та його окремих елементів;
- застосування звичайних та корегувальних шарів, багаторівневих масок, фільтрів, альфа-каналів та шляхів, макросів, керування напливом;
- багаторазової відміни виконаних дій;
- корекції кольорових тонів та контрастів, системи гнучкого та точного управління кольором і роботи з плашечними кольоровими каналами зі зниженою ймовірністю помилок кольоропередавання в сервісних бюро;
- кешування зображень для підвищення швидкості їх перерисовування;
- роботи з текстом та текстово-графічними композиціями.

Користувачами *Photoshop* найчастіше стають дизайнери, фотодизайнери, типографії, видавництва.

Потужні можливості *Photoshop* вимагають максимальної продуктивності комп'ютеру та визначають його високу вартість.

У процесі роботи з графічним редактором *Photoshop*, у центрі уваги знаходяться рядок меню та панелі інструментів,

які застосовуються для рисування на "холсті" (робочій області) та містять головні елементи управління.

Важливу роль відіграють інструментальні палітри – діалогові вікна для настроювання параметрів інструментів та здійснення операцій із зображенням.

*Photoshop* має великий набір інструментарію для роботи з бітовими образами: традиційні інструменти (пензлі, олівці та інші), оригінальні інструменти (наприклад, штамп для копіювання пікселів із однієї вказаної області до іншої), *Art*-пензлі для імітування традиційних технік рисування тощо (всі інструменти мають власні параметри, що дозволяють максимально точно настроювати їх для конкретної роботи).

Для виконання обтравки (точної обводки контурів об'єктів на зображенні) застосовують інструменти *Область*, *Ласо* та "*Чарівна паличка*": перші два інструменти виділяють ділянку зображення, обмежену геометричною фігурою; останній інструмент вибирає в рамках встановлених меж охоплення область, що відповідає умові співпадання кольорів.

Для пересування та копіювання виділених областей зображення застосовують інструмент *Пересування*.

Для рисування та ретуші як правило застосовують інструменти, що підтримують:

- імітацію олівця, пензля, ластика та аерографа;
- копіювання вибраних ділянок зображення шляхом штампування (набивки) ними як шаблоном інших ділянок зображення;
- прийом розмивання (зсуву сирої фарби), що часто застосовується художниками в процесі писання маслом, гуашшю, тушшю;

– змінювання на окремих ділянках зображення параметрів різкості/розмитості;

– локальну корекцію яскравості та кольорової насиченості за допомогою освітлювача, затемнювача та губки.

Для створення нових об'єктів, виконують такі дії: рисують чи виділяють відрізки прямих ліній та плавні криволінійні контури за допомогою інструментів *Лінія* та *Перо*, заповнюють виділені ділянки зображення суцільним кольором або з плавним переходом між кольорами за допомогою інструментів *Заливка* та *Градiєнт*, визначають колір будь-якої точки зображення та беруть його за зразок для інструментів панелі за допомогою інструменту *Піпетка*, виконують надписи за допомогою інструменту *Текст*.

Для визначення відстаней та кутів між елементами зображення застосовують вимірювальні інструменти (лінійки).

Для керування переглядом зображення найчастіше використовують інструменти, що дозволяють пересувати видиму область вздовж зображення та збільшувати/зменшувати зображення в цій області (інструменти *Рука* та *Масштаб*).

*Photoshop* має гарні інструменти для роботи з кольором, масками, формою відображення елементів управління програми.

Відзначимо такі можливості інструментальних палітр: настроювання параметрів інструментів редагування пензлів; редагування властивостей поточного інструменту; інформаційна підтримка засобів відображення; перегляд фрагментів зображення з різними масштабами; відображення та редагування поточних кольорових значень переднього плану та фону; управління відображенням шарів зображення; виконання мані-

пуляцій з каналами; управління контурами; створення макрокоманд (заданих послідовностей операцій з зображенням) та маніпулювання ними; робота з фільтрами (*Kai's Power Tools*, *Alien Skin*, *Andromeda* тощо).

*Photoshop* підтримує велику кількість графічних файлових форматів, включаючи формати *Web*-графіки. Він може імпортувати фотографії, отримані за допомогою цифрових камер. Також він дозволяє виконувати функцію оптимізації зображення з урахуванням специфіки роботи з ним в *Internet*.

*Photoshop* має можливість інтеграції (в процесі інсталяції) з іншим графічним пакетом фірми *Adobe* – *Image Ready*.

Хоча інтерфейси цих програм – однакові, а можливості редагування – одного рівня, з точки зору дизайнера, найоптимальніше використовувати *Photoshop* для виконання робіт з графічного редагування, а *Image Ready* – для виконання операцій, пов'язаних із використанням зображень в мережі, та для анімації, що базується на шарах.

**Лідером серед графічних пакетів для художників є *MetaCreation Painter*.** Він має ті ж базові можливості, що й *Adobe Photoshop*, але містить значно більше художніх пензлів, текстур, майстрів, спецефектів для імітування реальних інструментів, матеріалів та технік рисунка професійного художника. Допомагають у роботі з цією програмою графічні планшети. *MetaCreation Painter* підтримує кольорову модель *CMYK*, що зацікавлює користувачів, які працюють для поліграфії.

Недоліки *Painter* такі: складна робоча поверхня (через це його краще не застосовувати для "чистої" фотообробки); націленість тільки на тих, хто має художні навички; не найкра-

щі можливості для створення ілюстрацій, насичених текстовими ефектами; вимогливість до пам'яті комп'ютеру.

Програма обробки зображень *Corel Photo Paint* є традиційною складовою частиною графічного пакету *CorelDraw*.

Ця програма приваблює творчих графіків чудовими природними зразками заповнювань, що споряджені різноманітними переходами кольорів та художніми пензлями, точною роботою корекції контрастності. Професіоналам подобаються великі можливості програми, багатий набір інструментів, складна багаторівнева технологія, зручність перегляду та роботи.

Недоліки *Corel Photo Paint* такі: початково робоча поверхня розсіює увагу, а численні кнопки та палітри вимагають довгого звикання.

Графічний пакет *Picture Publisher*, який раніше розповсюджувався як складова частина *Micrografx Graphics Suite*, є недорогою, гарно відпрацьованою програмою з великими можливостями, яку легко освоювати. Безперечними її перевагами є рекордер команд, багато макросів та майстрів, недоліком – відсутність автоматичного попереднього перегляду фільтрів.

Графічний пакет *Macromedia XRes* спеціалізується на обробці великих файлів зображень: його перевагою є зберігання монтажів із економією ємності пам'яті. Але він не є для професіонала навіть частковою заміною *Photoshop* та вимагає для процесу монтажу великого обсягу пам'яті.

Для створення двовимірною комп'ютерного живопису часто застосовують графічні пакети *Fractal Design Painter*, *Macromedia FreeHand*, *Fauve Matisse*.



Ці графічні пакети надають широкий асортимент функціональних можливостей рисування, редагування зображень та тексту, роботи з кольором, наприклад, для моделювання: типових професійних інструментів художника (вуглинок, олівців, пензлів, пір'їн, аерографів та інших), матеріалів для рисування (акварель, олія, туш), багатокольорової градієнтної заливки, ефектів природного середовища.

На платформі Macintosh популярним є пакет **PixelPaint Pro** фірми *Pixel Resources*, призначений для редагування растрового живопису та зображень.

На платформі графічних робочих станцій *SGI* популярним є пакет **StudioPaint 3D** виробництва *Alias* та *Wavefront*, який дозволяє: рисувати різними інструментами в режимі реального часу на 3D-моделях; працювати з необмеженою кількістю шарів зображення; надавати декілька десятків рівнів відміни попередніх дій; виконувати операції корекції кольору; використовувати сплайнові пензлі (їх мазок можна редагувати попіксельно як сплайнову криву); підтримувати планшет із чутливим пером (це дозволяє художнику виконати традиційний ескіз від руки, перенести його в 3D-пакет для моделювання та анімації, та побудувати за цим ескізом 3D-модель).

**Особливості ілюстраційних програм обробки растрових зображень для початківців.** Ринок ПЗ пропонує більше десятка розповсюджених графічних пакетів різних цінових рівнів, які є недорогими растровими ілюстраційними редакторами без підтримки *СМУК* для початківців. Їх розділяють за продуктивністю на декілька класів.

По-перше, *це RGB-художники*, що своєю стандартною робочою *Windows*-поверхнею полегшують шлях до створення

користувачем перших витворів мистецтва: *PaintShop Pro*, *Softkey Photo Finish*, *Micrografx Windows Draw* та *Photo Magic*, *PhotoImpact*, *Lview Pro*, *Microsoft Windows Paint*.

По-друге, це програми з нетрадиційним інтерфейсом та великою кількістю шаблонів зображень, що полегшують входження в процес обробки зображень, типових для будь-якої сфери діяльності та побуту (в першу чергу – різних друкованих бланків, календарів, поздоровчих листівок, афіш, фотоальбомів, рамок для фотографій тощо). Їх типовими прикладами є програми: *Microsoft Picture It!*, *LivePix*, *Kai's Photo Soap*, *Ulead Photo Express*, *Adobe Photo Deluxe*.

У класі "для початківців" вартістю до 70 \$ багато вдалих інструментів пропонує графічний пакет *Windows Draw*. Проте той, хто робить ставку на фотомонтажі з заготовок, повинен звернутися до *Photo Express* чи *Picture It!*

Обробку зображень можна також здійснювати засобами офісних пакетів. Наприклад, у пакеті *Microsoft Office* зображення можна обробляти: з застосуванням наявної в комплекті програми *Photo Editor*; використовуючи можливості *Word*, *Excel* та *PowerPoint*.

Охарактеризуємо вказані пакети обробки растрових зображень для початківців.

Програма ***PaintShop Pro*** – це умовно безплатний класичний піксельний редактор, нескладний в опануванні, що швидко та гарно розв'язує типові повсякденні задачі, але має високу ціну повної версії. Програма *PaintShop Pro with Animation Shop* містить повноцінний засіб для створення анімованих GIF.

Пакет *Softkey Photo Finish* має різноманітні функції ретушування та є майстром у покращенні зображень. Але для монтажів – це не найкраща програма (не підтримує багаторівневі технології).

Пакет *Micrografx Windows Draw* характеризується найкращим співвідношенням ціни та якості функціонування, надає сотні зразків для друкованих бланків усіх видів, добре показує себе в створенні *HTML*-сторінок для *WWW*, містить потужну програму ретушування та ПЗ для рисування, підтримує багаторівневу технологію та автоматичні ефекти. Недоліком є використання власного формату.

У складі *Windows Draw* є програма обробки зображень *Photo Magic*, яка підтримує численні пензлі, ефекти, здійснення фотомонтажу, набір напівавтоматичних послідовностей операцій для генерації цікавих ефектів.

Пакет *PhotoImpact* підтримує відмінну техніку для творчості, забезпечуючи, завдяки своїм багатим функціональним можливостям, швидке наведення лоску на зображення. Його недоліком є відсутність *СМΥК*-функцій.

Пакет *LViewPro* – це гарна програма для швидкого перегляду та редагування графіки, що не відрізняється унікальністю, але повністю працездатна та може бути заміною для *PaintBrush*.

Стандартний у наборі *MS Windows* графічний пакет *Paint*, призначений для виконання найелементарніших графічних операцій, є найпростішим із перелічених растрових редакторів.

Програма *Microsoft Picture It!* – це вдале рішення для дизайнерів-початківців, яке надає можливість інтуїтивного використання, гарні ефекти та зразки, але має не дуже вдалу робочу

поверхню та неочевидне розташування потрібних пензлів та інструментів для вибору.

Програма *LivePix* – це ідеальний інструмент для виготовлення колажів, який містить зразки для багатьох чудових монтажів. Її безперечними перевагами є багато шаблонів та автоматична тінь, недоліком – слабкі опції настроювання контрастності.

Пакет *Kai's Photo Soap* – це продукт для дизайнерів, який має елегантну робочу поверхню з фотореалістичними символами інструментів. Його перевагою є гнучкий монтаж, недоліками – обмежені графічні можливості використання робочої поверхні та відсутність складних інструментів.

Програма *Ulead Photo Express* надає зразки для календарів та поздоровчих листівок, але може бути застосована для оформлення без використання цього набору зразків. Вона є корисною для початкового опанування технологій цифрової обробки зображень.

*Ulead Photo Express* пропонує опції для маніпуляцій з текстом та простий монтаж зображень для оформлення поздоровлень. Її перевагами є текстовий інструмент, який пропонує багато варіантів оформлення, та наявність багаторівневої технології для гнучких фотомонтажів, а недоліком – те, що деякі корисні функції доводиться шукати.

Пакет *Adobe Photo Deluxe*, призначений для початківців та тих, хто має деякий досвід роботи, пропонує два способи роботи: з попередньо штампованими проектами; шляхом самостійної обробки зображень. Ця програма підходить для швидкого виведення з застосуванням зразків.

Перевагами *Adobe Photo Deluxe* є багаторівнева технологія та потужна функція тіней, а недоліками – те, що в разі вільного оформлення зображень заважають не дуже вдала робоча поверхня та бідність деяких функцій, а також слабкий інструмент для плавних кольорових переходів.

Обговоримо особливості обробки зображень у *Photo Editor*.

Ця програма пропонує вибір прямокутників та не дуже зручну "чарівну паличку" Вона мітить (на основі подібних за кольором точок зображення) ті області об'єкту, що після вставки їх у документ *Word*, у сторінку *Web* або інші документи, повинні залишитися прозорими.

*Photo Editor* може обрізати зображення, обертаги його, робити його чіткішим, переобчислювати розмір файлу та змінювати глибину кольору. Є декілька простих можливостей корекції кольору та контрастності, чудовий фільтр ефектів (акварель, крейда, вугілля, папір для листів із ефектом ручної виділки).

Розглянемо особливості фотомонтажу в *Word*, *Excel* та *PowerPoint*.

У цих програмах можна керувати обробкою імпортованого зображення за допомогою інструментів рисування, що дозволяють регулювати яскравість і контрастність та робити прозорими подібні за кольором області тих об'єктів, які вставляють у зображення.

Можна відмінити будь-яку кількість виконаних команд. Усі зміни, пов'язані з яскравістю, контрастністю або виконанням вирізок, відмінюють за допомогою команди повертання графіки в початковий стан навіть після збереження зображення.

оригінал завжди залишиться збереженим у фоні, а при нашаруванні елементів зображення одне на одне точки того зображення, що лежить нижче, не будуть стертими.

Навіть на швидких ПК та в разі простих пересувань об'єктів ці монтажі швидко викликають втому. Для користувачів, які монтують багато, потрібен професійний пакет обробки зображень.

**Технологія графічних фільтрів.** *Plug-in* програми (фільтри) для графічних пакетів – це спеціальні програми, що спілкуються з графічними пакетами крізь *API* (виклик функцій фільтрів здійснюється за допомогою селектору крізь *API*).

Залежно від виду фільтрів, із посиланням на них можна передавати великі та складні структури даних: ці дані містять інформацію про зображення, на основі якої фільтр змінює початковий рисунок.

***Різні типи фільтрів розв'язують у графічних програмах різні задачі:***

- допомагають отримати дані зі сканеру або цифрової камери;
- здійснюють операції імпорту та експорту графічних файлів у різні формати;
- дозволяють ефектно маніпулювати зображенням (цей тип фільтрів називають графічними фільтрами).

Користувачі, що бажають художньо оформити зображення, звертаються до піксельних художників *Adobe PhotoShop*, *Micrografx Picture Publisher*, *Corel Photo Paint*, *Paintshop Pro* та інших, які пропонують великий набір функцій та інструментів для попіксельного редагування та дизайну графічних зображень. Але вони можуть далеко не все.

Для розширення можливостей ілюстраційних графічних редакторів існують додаткові інструментальні засоби. Наприклад, такі задачі, як створення природної текстури, моделювання структури поверхні, ефекти променистості та багато інших можна розв'язати тільки за допомогою спеціальних графічних фільтрів.

**Графічні фільтри – це невеликі програми (як правило, третіх фірм), які мають такі властивості:**

– залежно від потреб дизайнера, можуть бути легко вбудовані або видалені з графічного редактору;

– призначені для створення та ретушування зображення, реалізуючи різні ефекти на редагованому зображенні.

Наприклад, є такі графічні фільтри, що: стилізують зображення під акварельний рисунок, надають йому металевого блиску, викривляють згідно з задумом дизайнера, чітко прорисовують або м'яко окреслюють контури, накладають фарби а ля Моне, охоплюють полум'ям та димом, покривають снігом або краплями дощу, зображують відкинуті тіні тощо.

Фірма *Adobe* розробила інтерфейс *Photo-Shop-Plug-In* для програми *Photoshop*, який дозволяє легко додавати графічні фільтри. Його було вбудовано практично до всіх графічних програм.

**Графічні plug-in модулі, що звичайно становлять собою збірки окремих фільтрів, умовно класифікують за функціональним призначенням так:**

– фільтри для оптимізації зображення (*Intelligence*, *KPT Convolver* тощо);

– фільтри для моделювання природної техніки рисунку (*Chromatica, Eye Candy, Kai's Power Tools, Andromeda Series* тощо);

– фільтри для створення рамок, текстури та ефектів на задньому плані (*Photo Tools, Photo/Graphics Edges, Kai's Power Tools, Eye Candy, Andromeda Series* тощо).

Як правило, графічні фільтри тестують за такими показниками: ергономіка, функціональність, середовище, загальна оцінка.

**Технологія векторних ілюстраційних графічних редакторів.** Векторну комп'ютерну графіку використовують дуже широко – від зовнішньої реклами до підготування оригіналів для поліграфії.

Цьому сприяє те, що в порівнянні з растровим форматом, векторні зображення потребують меншого обсягу пам'яті, мають чіткіші лінії, а в процесі друку на них витрачається менше часу та ресурсів.

Векторні графічні редактори базуються на об'єктно-орієнтованому підході: зображення проєктують як об'єкт із певним набором властивостей та подій, що являє собою певну комбінацію базових (елементарних) об'єктів.

Головний елементарний об'єкт векторної комп'ютерної графіки – це лінія, що має властивості форми, накреслення, товщини, замкненості, кольору та інші. Найчастіше для формального опису ліній застосовують математичний апарат кривих третього порядку. В загальному вигляді рівняння кривої третього порядку, що може мати точку перегину, буде таким:

$$x^3 + \lambda_1 y^3 + \lambda_2 x^2 y + \lambda_3 x y^2 + \lambda_4 x^2 + \lambda_5 y^2 + \lambda_6 x y + \lambda_7 y + \lambda_8 = 0.$$



Інші базові математичні об'єкти-лінії є частинними або виродженими випадками кривої третього порядку:

– крива Безье є кривою третього порядку, форму якої визначають кути нахилу та довжини відрізків двох дотичних до сегменту кривої в її кінцевих точках;

– крива другого порядку без точок перегину (парабола, гіпербола, еліпс, коло тощо) в загальному вигляді описується рівнянням  $x^2 + \lambda_1 y^2 + \lambda_2 xy + \lambda_3 x + \lambda_4 y + \lambda_5 = 0$ ;

– пряма лінія описується рівнянням  $y = \lambda_1 x + \lambda_2$ ;

– точка описується вектором її координат.

У разі роботи з відрізками ліній, потрібне введення додаткових параметрів, які визначають кінцеві точки відрізків.

Завдяки поданню векторних зображень за допомогою комбінацій різних ліній та математичному опису ліній як єдиного об'єкту (на відміну від растрових зображень, що подаються як сукупності точок), векторний формат надає такі переваги:

– розмір та форму рисунку можна змінювати без втрати якості зображення;

– обсяг даних для відображення ліній суттєво менший, ніж у растровій графіці.

Узагальненням поняття лінії є елементарні графічні об'єкти відкритий та замкнений контури.

Контур може мати будь-яку форму (прямої, кривої або ламаної лінії, фігури). Основними параметрами обведення контуру, що визначають його вигляд при відображенні, є: колір лінії, товщина лінії, змінювання її товщини при різних кутах нахилу, тип лінії (суцільна, пунктир), ступінь згладжування, форма лінії на кутових та кінцевих точках, форма її

наконечників – маркерів на кінцевих точках (зі стрілкою, заокруглені тощо).

Усі замкнені контури мають властивість заливки (заповнення охоплюваної ними області), що буває декількох типів:

- заливка суцільним рівномірним кольором;
- градієнтна заливка з плавним переходом між кольорами (лінійна, квадратна, радіальна, кінцева);
- текстурна заливка (заповнення візерунками з регулярною структурою чи об'єктами *PostScript*);
- заливка зображенням, поданим растровою картою.

ООП-підхід дозволяє легко рисувати контури об'єктів, із їх подальшим заповнюванням. Оскільки контури об'єктів формуються за допомогою математичної моделі, складовими частинами якої є формальні описи відрізків прямих та кривих ліній, можна точно відтворювати ці контури, задаючи будь-який розмір.

Форму контуру змінюють шляхом пересування його вузлів (опорних точок), змінювання їх властивостей, додавання та видалення вузлів. Параметри вузлів впливають на форму кінців лінії контуру та характер її сполучення з іншими об'єктами.

Над декількома контурами можна виконувати такі операції:

- групування (початкові контури зберігають у сформованій групі об'єктів свої властивості та вузли);
- комбінування (складений контур набуває нових, загальних властивостей, а вузли залишаються тими ж);

– об'єднання (формується новий контур, із утворенням нових вузлів та змінюванням властивостей початкових контурів).

*Серед професійних редакторів векторної ілюстраційної комп'ютерної графіки лідирують та конкурують між собою три графічних пакети: CorelDraw, Adobe Illustrator, Macromedia Free-Hand.*

*Слід також відзначити розповсюджені векторні графічні ілюстраційні пакети CorelXara та Micrografx Designer.*

*До програмних засобів створення та обробки векторної графіки також належать векторизатори (трасувальники) типу Adobe StreamLine та CorelTrace – спеціалізовані пакети для перетворення растрових зображень у векторні.*

Проста монохромна графіка без напівтонів (креслення, чорно-білі рисунки) найкраще піддається векторизації. Напівтонові та кольорові зображення обробляються гірше, а результат векторизації потребує доопрацювання для наближення до оригіналу.

*Графічний редактор CorelDraw є фактичним стандартом серед програм створення та обробки векторної графіки, маючи велику популярність серед користувачів, а на платформі Windows його традиційно вважають основним векторним графічним пакетом.*

*CorelDraw є потужним та зручним інструментарієм створення традиційної ілюстраційної комп'ютерної графіки, здійснення художньої графічної творчості, дизайну фірмових знаків, оформлення реклами, створення ілюстрованої наукової та інженерної документації (наприклад, графіків даних науково-тех-*

нічних досліджень, точних креслень для галузей архітектури та машинобудування, навчальних посібників та керівництв).

Багато користувачів обирають *CorelDraw* завдяки звичному та зручному інтерфейсу, універсальності. В цілому, серед найважливіших переваг *CorelDraw* потрібно відзначити такі [72, 117, 149, 150]:

- прогресивна маркетингова політика фірми-розробника;
- великі, професійно виконані бібліотеки готових зображень;
- потужні графічні можливості, розвинена система управління та обширні засоби настроювання параметрів інструментів;
- найкращі можливості зі створення складних та якісних художніх композицій;
- потужні засоби вибору та редагування кольору (важливим є багатий вибір стандартних кольорових палітр фірми *Pantone*, що застосовуються в поліграфії);
- тісна інтеграція *CorelDraw* з пакетом обробки растрової графіки *Corel PhotoPaint* та програмою верстання *Corel Ventura Publisher*, які разом утворюють повноцінну систему підготовки друкованих та електронних публікацій;
- тісна інтеграція з *Internet*;
- гарний інтерфейс (потужні та широкі можливості інтерфейсу, потужна вбудована система навчання та підказок, зручні та інтуїтивно зрозумілі засоби створення та редагування графіки);
- наявність засобів, які не мають аналогів в інших векторних ілюстраційних редакторах.

Базові графічні інструменти *CorelDraw* дозволяють:

– рисувати різноманітні лінії – з властивостями кривих Безьє, в стилі "від руки", з імітацією рисування натуральним пером (плакатним пером, пензлем, аерографом, каліграфічним пером, чутливим пером для роботи з графічними планшетами), розмірні лінії, прямі з'єднувальні лінії (наприклад, для схем електричних з'єднань), інтерактивні з'єднувальні лінії (прив'язуються до об'єктів та пересуваються разом із ними);

– рисувати прямокутники, прямокутні контури з текстом, прості та зірчасті багатокутники, сітку для розлінування паперу (прямокутні об'єкти, розграфлені на однакові елементи), еліпси, кола, сектори, дуги, симетричні та логарифмічні спіралі;

– задавати параметри лінії абрису (контуру) та керувати способами заливки об'єктів, із можливістю вибору з бібліотеки готових варіантів заповнювання, керування параметрами та локального редагування заливки, створення власних моделей заповнення;

– обирати модель подання кольору, вказувати потрібний колір та присвоювати його контуру чи заливці об'єкту;

– динамічно керувати ступенем прозорості об'єктів;

– працювати з текстом, а саме створювати фігурний текст (вільно розташовані рядки) або абзацний текст (строго відформатований текстовий блок), редагувати та формувати текст (подібно до того, як це робиться стандартними засобами текстових процесорів та програм верстання), перетворювати текстові об'єкти в графічні, створювати суміщені текстово-графічні об'єкти (розташовувати текст уздовж прямої чи кривої лінії, всередині замкненого контуру);

– змінювати форму об'єктів шляхом модифікації їх вузлових точок, розбивати контури на складові частини, видаляти частини контурів та заливок об'єктів;

- вільно трансформувати об'єкти (обертати, масштабувати, нахилияти тощо);
- модифікувати форму об'єктів шляхом керування перетіканням (переходом) одного об'єкту в інший з автоматичним створенням заданої кількості проміжних трансформацій;
- створювати внутрішній та зовнішній ореоли графічного об'єкту (дублювати контурів об'єкту, що розміщаються з масштабуванням та зсувом відносно оригіналу);
- виконувати вільну деформацію об'єктів (застосовувати до них ефекти розшарування, зигзагу, смерчу);
- модифікувати об'єкти згідно з формою контуру, що їх огинає (пересуваючи опорні точки об'єктів або направляючи їх контури по вибраним користувачем лініям);
- здійснювати вичавлювання плоских об'єктів (відкритих або замкнених контурів) уздовж осі  $Z$  у растровому та векторному режимах для створення псевдотривимірного зображення;
- керувати параметрами контурів об'єктів;
- створювати ефекти перспективи та відкинутих тіней, застосування різних лінз (для перегляду об'єктів у режимі лупи), панорамного перегляду (пересування області видимості вздовж усього робочого поля);
- групувати та комбінувати об'єкти, здійснювати над ними логічні операції (*AND*, *OR*, логічне віднімання), поміщати їх в об'єкти-контейнери.

Можливості настроювання пакету *CorelDraw* (засоби управління параметрами інтерфейсу, властивостями проекту, параметрами взаємодії з апаратними пристроями та іншими складовими частинами зовнішнього оточення) унікальні та не мають аналогів в інших графічних програмах.

При настроюванні інтерфейсу *CorelDraw* можна визначати такі властивості:

- стиль інтерфейсу (стандартний, запозичений з графічних пакетів *Adobe Illustrator* або *Macromedia FreeHand*, власний, наприклад, індивідуальний інтерфейс із настроюванням на конкретні задачі та відповідною модифікацією системи меню);

- способи та параметри відображення робочої області, меню, інструментальних панелей, палітр, діалогових вікон та графічних зображень, параметри звукового оформлення, властивості графічних інструментів, точність позиціонування графіки, параметри вузлів, кількість рівнів відміни операцій;

- загальні властивості тексту (наприклад, одиниці вимірювання розміру шрифту) та атрибути текстових об'єктів, пов'язані з поняттями абзацу, шрифту, автозаміни, перевірки правопису тощо;

- параметри перевірки правильності адрес та стійкості зв'язків для документів, які містять посилання на ресурси *Internet*;

- параметри збереження файлів публікацій, робочих дисків, задіяних обсягів оперативної пам'яті.

У параметрах документу можна визначати такі властивості:

- розмір сторінки, режим відображення сторінки (наприклад, у вигляді аркушу поліграфічного формату з певним способом фальцовки), параметри наклейки до сторінки, режим показу елементів сторінки (наприклад, меж аркушу), параметри фонового зображення, властивості напрямних ліній (наприклад, меж області друку), параметри опорної сітки та прив'язки

вузлів об'єкту до вузлів цієї сітки, параметри вимірювальних інструментів;

– параметри рендерінгу векторних об'єктів, режим візуалізації, режими заливки незамкнених кривих та застосування ефектів до растрових зображень;

– атрибути текстів різних стилів оформлення (майже не відрізняються від тих, які застосовуються в текстових процесорах та програмах верстання);

– параметри підготування документів для публікації в Internet та зберігання їх у форматі *CorelDraw*.

*CorelDraw* дозволяє визначати глобальні параметри.

По-перше, це параметри управління відображенням та перетворенням кольору на різних апаратних пристроях (сканерах, моніторах, принтерах тощо), що є особливо важливими при підготованні кольорових публікацій.

Значення цих параметрів повинні відповідати конкретному набору апаратних засобів та особливостям публікацій. Для цього встановлюють відповідні кольорові профілі шляхом їх вибору зі стандартного переліку, завантаження з носія або через Internet. У разі відсутності спеціалізованого кольорового профілю, вказують згенерований профіль, що використовується системою управління кольором *Eastman Kodak Digital Science ICC*, застосованій в *CorelDraw*.

По-друге, це параметри перевірки перед друком (у разі необхідності *PostScript*-виведення документу) та параметри попереднього перегляду зображення на екрані.

Вкладка *Фільтри* містить великий перелік фільтрів імпорту/експорту растрової та векторної графіки, тексту різних файлових форматів (цей перелік можна редагувати), а вкладка



Асоціювати підтримує операції зв'язування розширень імен файлів із певними форматами документів.

Побудова та модифікація об'єктів засобами *CorelDraw* (як елементарних геометричних фігур, так і складних композицій) передбачає взаємодію різних елементів управління (інтерактивних меню, панелей інструментів та властивостей тощо), конфігурацію яких користувач може змінювати на власний розсуд.

Прийоми рисування типових об'єктів засобами *CorelDraw* в основному є традиційними для всіх векторних графічних редакторів: потрібно маніпулювати властивостями об'єктів та модифікувати їх.

Склад полів та елементів управління суттєво залежить від типу конкретного об'єкту. Але всі об'єкти мають ряд загальних атрибутів та можливостей модифікації.

Так, після вибору будь-якого об'єкту на панелі властивостей буде відображено:

- поля з параметрами координат базової точки, розмірів, відсотків масштабування та кута обертання об'єкту;

- кнопки дозволу/заборони пропорційного масштабування та інструментів дзеркального відбиття по горизонталі та вертикалі.

Якщо не вибирати жодного об'єкту, то панель властивостей містить елементи управління параметрами поточної сторінки.

Панель властивостей динамічно змінюється залежно від типу вибраного об'єкту та інструменту для роботи з цим об'єктом: вона відображає властивості цього інструменту та надає

набір елементів для керування параметрами інструменту та властивостями об'єкту.

Диспетчер об'єктів є потужним засобом управління їх положенням та контролю їх властивостей.

*CorelDraw* має спеціальні засоби для точного позиціонування, вирівнювання та розподілу об'єктів відносно робочого аркуша та інших об'єктів, управління проектом в цілому, його окремими елементами та конкретними об'єктами.

Цих засобів, сумісно з можливістю явно задавати розміри та положення об'єктів на панелі властивостей, достатньо для більшості інженерних задач, створення складних багаторівневих документів та організації сумісної роботи з їх розробки.

***Графічний редактор Adobe Illustrator є усталеним індустріальним стандартом та загально визнаним лідером серед програм створення та обробки векторної комп'ютерної графіки.***

Це потужна програма векторної графіки, що має повний набір інструментів для роботи з нею та стала моделлю-прототипом для багатьох інших векторних графічних редакторів [1, 148].

Суттєвою перевагою *Adobe Illustrator* є те, що він виконаний в єдиному стилі та може здійснювати відлагоджену взаємодію з іншими графічними пакетами компанії *Adobe*, такими як растровий графічний редактор *Adobe Photoshop* та система комп'ютерного верстання *Adobe PageMaker*, утворюючи разом із ними єдиний, функціонально завершений пакет графічного ПЗ.

Ті дизайнери, що поважають грамотний кольороподіл, безперечно обирають *Adobe Illustrator*.

*Adobe Illustrator* дозволяє відкривати та редагувати *EPS*-файли, створені в будь-якій програмі, що може генерувати формат *PostScript*. Він повністю підтримує файли програми *Acrobat* у форматі *PDF* (основному форматі розповсюдження сторінок у мережі *WWW*).

*Adobe Illustrator* є зручним не тільки для професіоналів, а й для початківців завдяки зрозумілому інтерфейсу та розвиненим функціональним можливостям (хоча він не має такого звичного та зручного інтерфейсу й універсальності, як *CorelDraw*).

Основні елементи управління *Adobe Illustrator* зосереджено в рядку меню, панелі інструментів та палітрах.

Панель інструментів нагадує *Adobe Photoshop*, поєднуючи інструменти виділення областей, рисування геометричних фігур та контурів, редагування, елементарних геометричних перетворень та трансформації графічних об'єктів, створення текстових об'єктів, вибору кольору за зразком, заливки об'єктів, управління переглядом об'єктів тощо.

Загальні властивості, методи управління відображенням та параметрами численних палітр інструментів *Adobe Illustrator* такі ж, як у *Photoshop*. Наприклад, палітри для роботи з графікою дозволяють задавати властивості контурів, присвоювати об'єктам додаткові атрибути (текстовий коментар, гіперпосилання на об'єкт *Internet* та інші), вирівнювати та рівномірно розподіляти виділені об'єкти відносно аркуша документу або інших об'єктів тощо. А палітра ММ-дизайн містить засоби управління шрифтами *Multiple Master*, які дозволяють міняти накреслення символів.

Векторний графічний редактор *Macromedia FreeHand* має потужні інструментальні засоби розробки складних ілюстрованих документів, поступаючись *Adobe Illustrator* та *CorelDraw* тільки по окремих позиціях. *FreeHand* спеціально адаптовано для сумісної роботи з програмою комп'ютерного верстання *QuarkXPress*. Він має простий та дружній інтерфейс, є зручним для початківців.

*FreeHand* властиві гарна швидкодія, невеликий розмір та невибагливість до апаратних ресурсів.

Графічний векторний редактор *CorelXara* призначений у першу чергу для створення графічного зображення та формування блоку тексту на сторінці за один сеанс роботи. Він дозволяє виконувати з рисунками, градієнтним заповненням та діапозитивами найдосконаліші дії. *CorelXara* може не все, але для деяких практичних задач не має собі рівних. Наприклад, при підготованні складних оригінал-макетів або в процесі навчання дизайнерів-початківців *CorelXara* стане гарним доповненням до комплекту інших інструментів.

Графічний пакет *Micrografx Designer* дозволяє за допомогою простого інтерфейсу легко впоратися з багатьма типовими для офісу графічними роботами. Але цей графічний пакет не дуже годиться для професійних художників-графіків (через принципові обмеження інструментальних засобів та можливостей кольорового друку).

Трасувальник *Adobe StreamLine* є лідером серед програм-векторизаторів. Він дозволяє покращувати точність векторизації завдяки тонкому настроюванню її параметрів. На відміну від потужних трасувальників креслень, *StreamLine* менш вимогливий до апаратних ресурсів та набагато дешевший.

## Питання для самоперевірки

1. Чим відрізняються растрова та векторна графіка?
2. Яке призначення мають графічні фільтри?
3. Які задачі виконують растрові та векторні ілюстраційні графічні редактори?
4. Чим відрізняються професійні та любительські ілюстраційні графічні редактори?
5. Назвіть галузі практичного застосування растрових та векторних ілюстраційних графічних редакторів.

### 11.3. 3D-графіка, мультимедіа-дизайн та Web-дизайн

**3D-графіка та дизайн.** *Створення реалістичних тривимірних зображень передбачає 3D-моделювання, реалістичну візуалізацію, анімацію, морфінг, дизайн та відеодизайн.*

Системи 3D-графіки та дизайну, що реалізують технології створення комп'ютерних 3D-моделей різноманітних об'єктів, суміщають векторний та растровий способи формування зображень.

Практичним застосуванням систем комп'ютерної 3D-графіки та дизайну є моделювання фізичних об'єктів – створення реалістичних статичних та рухомих 3D-зображень реальних фізичних тіл та оточуючого середовища для таких галузей, як інженерні САПР, наукові та науково-технічні дослідження та

розробки (з медицини, біології, хімії, астрономії, фізики тощо), дизайнерські розробки, рекламний бізнес, кіноіндустрія, навчання, індустрія ігор і розваг та інші.

Особливий інтерес викликає 3D-моделювання в режимі реального часу. Наочним прикладом є тренажери технічних засобів, які відтворюють технічні параметри об'єктів та властивості довкілля.

На ПК реалізують прості тренажери для навчання водінню наземних транспортних засобів, цивільної авіації та плавзасобів.

Тренажери для пілотування військової та космічної авіації реалізують на базі комплексів потужних та високошвидкісних графічних робочих станцій багатомільйонної вартості (вони швидко окупаються, оскільки вартість тренування на реальній техніці на порядок вища).

У загальному випадку 3D-моделювання, реалістичну візуалізацію та анімацію об'єкту виконують так [63, 130]:

- будують 3D-каркас, максимально наближений до реальної форми об'єкту;

- створюють текстури – віртуальні матеріали, що як найкраще імітують реальні за їх властивостями (фізичними, хімічними, біологічними та іншими) в процесі комп'ютерної візуалізації;

- накладають текстури на об'єкт (призначають матеріали різним частинам поверхні об'єкту);

- встановлюють фізичні параметри довкілля, тобто визначають стан (властивості) освітлення, атмосфери, гравітації, взаємодіючих об'єктів та поверхонь;

- задають траєкторії руху об'єктів;

- обчислюють послідовність кадрів анімаційного ролику;
- накладають на анімаційний ролик поверхневі ефекти для надання йому більшої реалістичності.

Після завершення процесу розрахунку реалістичних зображень (візуалізації, рендерінгу), комп'ютерна 3D-анімація використовується як автономна розробка або як складова частина іншого програмного додатку.

**Фундаментальними підходами до створення реалістичної 3D-моделі є каркасне та поверхневе 3D-моделювання об'єкту** – застосування до побудови його каркасу базових геометричних примітивів (трикутників, багатокутників, багатогранників) та гладких сплайнових поверхонь.

Одним із найпопулярніших є метод *NURBS*-поверхонь, визначених за допомогою математичного апарату бікубічних раціональних B-сплайнів на нерівномірній сітці.

Форма та ступінь гладкості *NURBS*-поверхонь залежать від: взаємного розташування опорних точок сітки; призначеного кожній опорній точці коефіцієнту, що визначає ступінь її впливу на найближчу ділянку поверхні.

Технологія 3D-моделювання дозволяє працювати з численними геометричними примітивами та сплайновими поверхнями об'єкту як із єдиним цілим, враховуючи їх взаємодію на основі попередньо заданої фізичної моделі.

Найчастіше деформацію об'єкту виконують або методом пересування контрольних точок, пов'язаних із опорними точками цього об'єкту (ступінь впливу на опорні точки визначається відстанню до них), або методом пересування вузлів тривимірної пружної сітки деформації, що охоплює потрібну частину об'єкту.

*Твердотільне 3D-моделювання є ще однією фундаментальною технологією побудови об'єктів на базі примітивів:* об'єкти подають як тверді тіла, що можна трансформувати шляхом їх взаємодії з іншими твердими тілами (наприклад, робити в них отвори, заглиблення або опуклості за допомогою стандартних логічних операцій).

Для того, щоб покрити каркас об'єкту матеріалами реалістичного вигляду, необхідно задати або обчислити коефіцієнти прозорості поверхонь та кути переламлювання променів світла на межі матеріалів та довкілля.

Залежно від типу поверхні реального матеріалу, для побудови його віртуального аналогу найчастіше використовують такі класи фізичних моделей: матові поверхні з дифузним відбиттям та без відблисків, металеві поверхні, глянцеві поверхні, візуалізація поверхонь з урахуванням ефекту поляризації світла, моделювання поверхонь з можливістю корегування напрямків відбиття та параметрів переламлювання світла [63].

Для зафарбовування поверхонь використовують методи Гуро (колір примітиву обчислюють у вершинах та лінійно інтерполюють по поверхні) та Фонга (обчислюють та інтерполюють по поверхням складових примітивів вектор нормалі до об'єкту, а далі обчислюють освітлення для кожної точки).

У 3D-графіці виділяють такі складові частини потоку світла, що йде від поверхні об'єкту: власне світіння поверхні; відблиски (відбите світло природних та штучних джерел); дзеркально відбите світло; дифузне світло, рівномірно розсіяне поверхнею; переламлене світло, що йде зі зворотної сторони поверхні.



Світло, що йде в бік спостерігача від конкретної точки поверхні, є сумою цих складових частин, яку помножено на коефіцієнт, залежний від матеріалу та кольору поверхні в цій точці [63].

Опис заданих або обчислених візуальних властивостей поверхні заноситься в масиви  $2D/3D$ -текстур, які містять дані про прозорість та коефіцієнт переламлювання матеріалу, коефіцієнти зсуву складових частин світлового потоку, колір у кожній точці, характеристики відблиску (колір, ширину, різкість), колір фонового розсіяного освітлення, характеристики шорсткості поверхні.

Як правило, моделюють віртуальні еквіваленти таких фізичних джерел світла, яким властиві колір та інтенсивність [63]:

- направлене джерело світла (прожектор, лазерний промінь), яке характеризується координатами просторового розташування та кутами растрову повного конуса світла та його найяскравішої плями;

- точкове джерело світла (електролампочка), що характеризується координатами просторового розташування та рівномірно випромінює світло в усіх напрямках;

- віддалене неточкове джерело світла (сонце, місяць), яке характеризується координатами просторового розташування та геометричними параметрами розміру та форми;

- розчинене (розсіяне) світло, тобто рівномірний світловий фон поза видимістю інших джерел світла, що не має геометричних параметрів.

Накладання текстур на певні ділянки каркасу об'єкту потребує творчих здібностей виконавця та враховування взаємного впливу текстур на межах примітивів.

У разі моделювання рухомих механічних пристроїв, найчастіше застосовують *метод інверсної кінематики руху*: анімація об'єкту (задавання параметрів його руху) базується на технології ключових кадрів, які відповідають ключовим положенням об'єкту (початковому, певній кількості проміжних та кінцевому); інші кадри обчислюються автоматично згідно зі спеціальним алгоритмом.

На відміну від простої лінійної апроксимації, цей алгоритм здійснює плавне змінювання положення опорних точок об'єкту згідно з заданими умовами, що визначені ієрархією підпорядкування, законами взаємодії та обмеженнями на значення параметрів руху об'єктів 3D-сцени (граничними значеннями швидкостей, прискорень та кутів обертання, дозволеними площинами пересування тощо) [63].

У разі анімації віртуальних аналогів живих об'єктів, використовують *методи скелетних моделей та фіксації реальних рухів фізичного об'єкту*.

Метод скелетних моделей забезпечує правдоподібну імітацію рухів живих істот у ході пересування об'єктів. Згідно з ним: створюють каркас, рухомий у характерних точках об'єкту моделювання; рух точок обчислюють за методом інверсної кінематики руху; моделюють та накладають на каркас оболонку з поверхонь, для яких каркас є набором контрольних точок; візуалізують цю каркасну модель шляхом накладання поверхневих текстур із урахуванням умов освітлення [63].

Метод фіксації реальних рухів фізичного об'єкту є найдосконалішим методом анімації: поведінка віртуальної моделі практично не відрізняється від живого прототипу. Згідно з цим методом [63]:

– у контрольних точках реального об'єкту закріплюють яскраві джерела світла та знімають певні види руху цього об'єкту на відеоплівку чи кіноплівку;

– координати контрольних точок реального об'єкту покадрово ідентифікують на плівці, заносять у комп'ютер та ставлять у відповідність контрольним точкам віртуальної каркасної моделі.

В основу більшості сучасних програм візуалізації (рендерінгу) покладено метод зворотного трасування променів.

Застосування складних математичних моделей дозволяє вдало імітувати природні та штучні фізичні ефекти (дощ, сніг, туман, дим, вогонь, вибухи тощо).

Технологія 3D-моделювання передбачає методи розрахунку процедурних ефектів та взаємодії систем часток, які вимагають настільки значних обчислювальних ресурсів, що на ПК реалізують їх спрощені варіанти.

*Серед засобів створення 3D-моделей та обробки 3D-графіки лідирують пакети Autodesk/Kinetix 3D Studio MAX, Microsoft Softimage 3D та Alias/Wavefront/TDI Maya, що найефективніше працюють на найпотужніших ПК.*

Дещо відстає від них розповсюджений додаток *CorelDream 3D*: він має менше можливостей та помітні недоліки при роботі з поверхнями об'єктів.

Також потрібно відзначити *Caligari TrueSpace* – систему розробки 2D/3D-моделей та сцен із наступною їх візуалізацією:

вона менш відома, але має широкі можливості, серед яких – створення *VRML*-файлів (побудови 3D-світів для перегляду в *Internet*).

Вказані системи забезпечують не тільки створення 3D-об'єктів та 3D-сцен, а й застосування до них таких перетворень:

- трансформації (зсув, обертання, масштабування, розчавлювання, морфування);
- модифікації (вигин, крутіння);
- деформації (вибух, хвиля, дія гравітації);
- ефекти, що додають зображенню реалістичності – текстурування та структурування поверхонь, реалістичне освітлення, атмосферні ефекти (наприклад, опади, туман, дим), ефекти перспективного зображення та об'ємності тощо.

Ці системи успішно застосовують для дизайну (особливо – внутрішніх та зовнішніх інтер'єрів), реклами, в кіно/відео-бізнесі, в індустрії комп'ютерних розваг.

**3D Studio MAX** є однією з найвідоміших та найрозповсюдженіших систем створення 3D-спецефектів, новим підходом до 3D-моделювання та візуалізації. Відносна простота та невибагливість до ресурсів зробили її розповсюдженою в усьому світі.

Програмний продукт тривимірної анімації, моделювання та візуалізації **3D Studio MAX** має широкі можливості: якщо чогось нема в самому пакеті – до послуг його численні доповнення [85, 93, 94, 114,116]. Він підтримує велику кількість апаратних прискорювачів 3D-графіки, потужні світлові ефекти, значну кількість доповнень від сторонніх фірм.

Цей графічний пакет є потужним та ефективним засобом для створення тривимірних роликів, заставок, спецефектів до фільмів та супроводжувальної анімації в комп'ютерних іграх.

*3D Studio MAX успішно конкурує з такими "китами" 3D-індустрії, як Power Animator, Softimage, Light Wave, а також специфічними програмами типу Bryce3D, Organica, Rhino для конструювання віртуальних світів (генерації 3D-ландшафтів, створення 3D-фігур та їх анімації) та іншими.*

Але початково *3D Studio MAX* було призначено для створення та обробки 3D-графіки на напівпрофесійному рівні з метою побудови якісних 3D-зображень неживої природи, тому його засоби моделювання та анімації поступаються розвинутішим програмним засобам.

Будучи першою серйозною системою для 3D-моделювання, *3D Studio* став прототипом для великої кількості інших систем, фактичним стандартом 3D-графіки на *PC* та має велику кількість прихильників в усьому світі серед усіх категорій користувачів. Це – найкращий вибір для тих, хто хоче ввійти до світу 3D-анімації.

Для професіоналів, особливо для тих, хто перейшов із платформи *SGL* на платформу *PC*, є цікава альтернатива пакету *3D Studio MAX* – *Softimage 3D* та *Maya*.

Графічний пакет *Softimage 3D* фірми *Microsoft* є фактичним стандартом на платформі спеціалізованих графічних робочих станцій *SGL*. Він має багаті можливості моделювання, велику кількість настроюваних фізичних та кінематографічних параметрів, якісний та швидкий модуль рендерінгу, численні доповнення від третіх фірм. На платформі *IBM PC* цей графічний пакет вимагає потужних апаратних ресурсів.

Графічний пакет *Maya* консорціуму провідних фірм *Alias*, *Wavefront* та *TDI*, є найпрогресивнішим інструментарієм створення та обробки 3D-графіки на платформах *SGI* та *PC*.

Система *Maya* має модульну архітектуру, найсучасніші та зручні в використанні функціональні можливості та інтерфейс, повний комплект базових класів інструментів (геометричне та фізичне моделювання, анімація, візуалізація) [59].

Прототипами *Maya* є: *Alias Power Animator/Designer*, програми комплексу *Wavefront* та *TDI Explorer* на платформі *SGI*.

До складу графічного пакету *Maya* входять опорні модулі (ядро *Base*, *F/X*, *Power Modeler*, *Artisan*, *Cloth*, *Fur*, *Live*), що підтримують: інструменти моделювання, засоби полігонального та сплайнового моделювання, метод твердого тіла, моделювання фізики взаємодії м'яких тіл, методи інверсної кінематики та захоплення руху, рендерінг, базові ефекти та додаткові ефекти обробки систем часток (снігу, дощу, граду, туману тощо), обробку віртуальних моделей за допомогою типових реалістичних методів роботи художників та скульпторів (рисуння по поверхням об'єктів за допомогою пензлів, згладжування поверхонь або надання їм шорсткості за допомогою скульптурних різців), моделювання повстяних та міхових поверхонь, моделювання одягу, сценарне сполучення реальних зйомок на натурі та комп'ютерної анімації, обробку звуку [59, 63].

**До функцій систем анімації та морфінгу графічних зображень входять:** реалістичне моделювання природної динаміки та пластики людського тіла та рухів тварин; моделювання іміджу людини (міміки обличчя, характерних рухів, зачіски, макіяжу, одягу тощо).

До найбільш відомих програм-іміджмейкерів належать: *FaceWorks Studio* від *Digital Equipment Corporation* (Америка); *Nautilus Designer Pro* (Санкт-Петербург) та її конкуренти "Ваш стиль" та *Virtual Makeover*; різноманітні фотороботи в криміналістиці. А до розповсюджених програм створення реальних моделей людей та тварин належить програма *Fractal Design Poser*.

Так, графічний пакет *Poser*, який може працювати сумісно з *Fractal Design Painter*, *Ray Dream Studio*, *Fractal Design Detailer* та має потужні можливості імпорту/експорту, в плані 3D-моделювання та візуалізації дозволяє: формувати віртуальні людські тіла, додавати опори, замінювати частини віртуальних тіл імпортованими 3D-моделями, споряджати їх одягом та іншими аксесуарами, анімувати, легко та природно вбудувати їх у 2D/3D-сцени, *Web*-графіку та мультиплікацію.

Окрім застосування в кінопромисловості, відеопромисловості, створенні мультиплікаційних фільмів, телебаченні та криміналістиці, візажистами, стилістами, перукарями та дизайнерами одягу, ці системи використовують для попереднього відпрацьовування рухів людини в небезпечних професіях (каскадери, циркові акробати, рятувальники, десантники тощо).

**Усі пакети 3D-графіки мають можливості відеодозайну. Але для цього існує спеціальне програмне забезпечення – *Adobe Premiere*, *Adobe Media*, *Adobe After Effects* та інші пакети** (дивіться про це детальніше в гл. 7) [21, 90].

Для побудови графічних зображень, що можуть бути подані як регулярні структури (від простих геометричних фігур до складних природних ландшафтів та 3D-об'єктів), застосовують *фрактальну графіку*.

Як і векторна графіка, фрактальна графіка базується на математичних обчисленнях, але не потребує зберігання об'єктів у пам'яті комп'ютеру, оскільки єдиними базовими елементами для побудови зображення є математичні формули (рівняння).

**Мультимедіа-дизайн.** *Основні компоненти мультимедіа* – це графіка, текст, фото, відео, звук, анімація (мультиплікація).

*Однією з переваг мультимедійних технологій є використання гіперзв'язків* – програмного методу, за допомогою якого різноманітні терміни, статті, зображення, звуки, фрагменти відео та інші компоненти внутрішньо ув'язують за логічними критеріями.

Використовуючи гіперзв'язки, матеріал неважко подати так, щоб користувачі змогли переглядати його максимально зручним способом – за асоціацією.

На основі мультимедійних та гіпермедійних технологій створюють значну частину таких широко розповсюджених на практиці програмних додатків [44, 63, 73, 76, 165]:

– інформаційно-довідкових, освітніх та розважальних програм (енциклопедій, альманахів, збірок довідників, інтерактивних ігор, кінофільмів, які супроводжені сценаріями, біографіями акторів, примітками режисера, аналітичних оглядів тощо);

– малотиражної продукції рекламно-інформаційного характеру (каталогів, довідників, презентацій тощо).

*Розглянемо класифікацію програмного інструментарію для створення мультимедійних додатків.*



Засоби, що дозволяють об'єднати окремі компоненти мультимедіа в єдине завершене ціле – в мультимедійний додаток – умовно поділяють на три групи.

Перша група – це універсальні авторські системи, тобто потужні інструментальні засоби розробки мультимедійних додатків будь-якого типу.

Типовими представниками першої групи засобів створення мультимедійних додатків є: *Grasp (Glpro)*, *Tempra Media Author*, *Ten Core Language*, *MediaView*, *Authorware*, *Icon Author*, *TIE*, *Quest*, *Apple Media Kit*, *Ten Core Producer*, *CBT Express*, *HyperCard*, *SuperCard*, *Multimedia Toolbook*, *Director*, *PowerMedia*, *MediaMogul*, *mTropolis*, *FireWalker*, *HyperMethod*, *Formula Graphic*, *HM-Card*, *Everest*, *HotDogPro*, *FrontPage*, *HotMetalPro*, *PageMill*, *Arachnophilia* та інші.

Друга група – це спеціалізовані програми для швидкого підготування окремих типів мультимедійних додатків (презентацій, публікацій в *Internet* тощо).

Типовими представниками другої групи засобів створення мультимедійних додатків є: *PowerPoint*, *Action!*, *Astound*, *Compel* та інші.

Третя група – мови програмування високого рівня.

Типовими представниками третьої групи засобів створення мультимедійних додатків є: *Delphi*, *C++ Builder*, *Visual C++*, *Java*, *Visual J++* та інші.

Розглянемо основні способи та оптимальний шлях створення мультимедійних додатків.

Виходячи з наведеної вище класифікації, існує три основних способи створення мультимедійних додатків:

- 1) використання інструментальних пакетів (авторських та спеціалізованих програм);
- 2) пряме програмування;
- 3) розширення функцій інструментальних пакетів за допомогою інструментарію мов програмування.

Перший спосіб надає економію засобів та часу, але програє за ефективністю роботи створеної програми, необхідністю оволодіння спеціальними прийомами роботи та наявністю ряду обмежень.

Другий спосіб – більш витратний, але дозволяє отримати ефективніші програми та надає гнучкий інструментарій.

Найоптимальнішим є третій спосіб, але його не завжди можна реалізувати на практиці.

Оскільки задача вибору засобу створення мультимедійного додатку є непростюю та не має універсального розв'язку, дуже важливим у процесі розробки є етап вибору: якщо ви помилитеся, час та гроші будуть втрачені.

Зробити правильний вибір дозволить володіння поточною інформацією про інструментальні засоби та методи створення мультимедійного ПЗ, яку систематично надає комп'ютерна періодика.

Щодо засобів створення мультимедійного ПЗ, які користуються великою популярністю у пересічного користувача, то це *PowerPoint* та *HyperMethod*.

**Зупинимся на ключових проблемах сфери мультимедіа.**

Перша проблема мультимедіа – ємність носіїв.

Мультимедіа-додатки, що містять текст, графіку, анімацію, стереозвук та 3D-зображення, вимагають значної ємності

сховищ даних, а тому надзвичайно швидко вичерпують можливості традиційних компакт-дисків. Певним вирішенням цієї проблеми стали двосторонні компакт-диски *MLR* та *DVD*.

Друга проблема мультимедіа, яку досі не вирішено – низька пропускна здатність *Web* у процесі передавання мультимедійної інформації. Ємність серверів може бути практично безмежною, але не буде мати сенсу без потужних каналів зв'язку.

На даний час настільні мультимедіа-додатки можуть бути доступні у кожній оселі, що не скажеш про доставку мультимедіа каналами Internet.

**Web-дизайн.** Програмування та дизайн для *Web* передбачають комплексне застосування численних та різноманітних пакетів, таких як: *Macromedia Fireworks*, *FreeHand*, *DreamWeaver*, *Flash*, *Generator*, *Director*, *Shockwave Player*, *Flash Player*, *HomeSite*; *JAVA Script It*; *Web Plus Shop E-commerce*; *Norton Internet Security*; *CoolIce*; *Liquid FX*; *MetaCreations Headline Studio*; *DigiCam Web Page Generator*, *Arles Image Web Page Creator*, *AU2HTML*; *WiseBot Pro*; *The Icon Extractor*; *Animagic GIF Animator*; *ADing SplitImg*; *Advanced GIF Optimizer*; *WebTrends Professional Suite*; *Button Studio*; *MailCollect*; *CGIMailer*; *Anfyjava*; *CGI StartPro*; *Ultimate Bulletin Board*; *Microsoft FrontPage*; *PERL* та інші.

Важливу роль для розв'язування задач *Web*-дизайну відіграють *Web*-броузери та *Web*-редактори.

Окрім того, сучасні версії графічних пакетів класу "для професіоналів", мови програмування високого рівня та офісні пакети теж містять засоби підготування матеріалів для *Web*.

*Розглянемо питання про те, як правильно орієнтуватися в цій інформації.*

У процесі підготовки графічних матеріалів для *World Wide Web* можна застосовувати будь-які програми. Але деякі графічні пакети є особливо зручними або мають унікальні властивості, що полегшують працю *Web*-ілюстраторів.

Рекомендовані графічні пакети можна класифікувати за програмною платформою їх функціонування [8, 20, 27, 35, 49, 69, 70, 82, 101, 107, 167, 172, 175, 80], виділивши серед них чотири базових групи. *Першу групу становлять графічні пакети для Microsoft Windows/NT/XP.* Коротко оглянемо їх.

*Adobe Photoshop:* багато *Web*-дизайнерів віддають йому перевагу завдяки легендарним фільтрам та підключуваним модулям, здатності видаляти зазубреності з усіх основних примітивів та підтримці великої кількості графічних форматів.

*Gif Construction Set for Windows (GIFCON)* компанії *Alchemy Mindwork:* дозволяє робити з *GIF*-файлами практично все.

*CorelXARA:* використовує, як і *CorelDRAW*, векторну графіку, але орієнтований на *Web*-дизайнерів, вмє видаляти зазубреності, має численні засоби для деформування, реалізує унікальну можливість компонування растрових та векторних даних, дозволяючи створювати неперевершені ефекти, вмє створювати анімовані *GIF*-файли.

*Пакети для роботи з 3D-графікою:* зручно використовувати як професійні (складні у вивченні) системи типу *Autodesk 3D Studio* та *3D Studio MAX*, так і простіші системи типу *Caligari TrueSpace* та комплекту засобів для створення 3D-сцен у форматі *VRML* від компанії *Caligari*.

### *Друга група – програмні пакети для ОС UNIX.*

Для деяких версій ОС UNIX (*SUN Solaris, SGI IRIX*) існують версії пакету *Adobe PhotoShop*. В *Internet* безкоштовно доступні декілька графічних пакетів: *GIMP (General Image Manipulating Package)*, *XPaint* тощо. За можливостями їх не порівняти з комерційними пакетами для ОС UNIX та MS Windows.

### *Третя група – програмні пакети для Apple Macintosh.*

Користувачам доступні пакети *Adobe Photoshop, Adobe Freehand* та багато інших пакетів на цій платформі, що здавна відома орієнтованістю на графіку та настільні видавництва.

### *Четверта група – програмні пакети для декількох платформ.*

Наприклад, програма *Gifblast* працює в середовищі MS DOS та різних версій UNIX, є корисною для роботи з зображеннями в форматі GIF (оптимізує GIF-файли, зменшуючи їх розмір, у тих випадках, коли виробники програм, які працюють з файлами в форматі GIF, погано оптимізують свої продукти).

### *Розглянемо деякі пакети, що отримали найбільше розповсюдження та стали найбільш відомими серед користувачів.*

*Adobe ImageReady* – пакет, призначений спеціально для Web-дизайнерів. Він реалізує такі корисні для підготовки зображень для Web-сторінок функції, як контроль ступеня стискання файлів у форматі JPEG та попередній перегляд зображень у вікні броузеру. Починаючи з 9-ої версії, цей пакет може бути в процесі інсталяції інтегрований з пакетом *Adobe Photoshop*.

Одними з найбільш популярних інструментів для *Web*-дизайну є *HTML-редактори HotDog та DreamWeaver* [78].

*Великої популярності набув професійний стандарт для створення привабливих Web-сайтів із 2D/3D-графікою та анімацією – пакет Flash від фірми Macromedia* [18, 29, 65, 74, 89].

*Flash* дозволяє, навіть після мінімального навчання роботи з ним, швидко створювати цікаві, барвисті та насичені *Web*-сайти, що одночасно є компактними, практичними, швидко завантажуються та добре запам'ятовуються користувачам.

Цей графічний пакет приваблює дизайнерів покращеними можливостями створення графіки, яку вони можуть сполучати з легким та зручним інтерфейсом користувача.

Професійних розробників *Flash* приваблює можливістю створювати різноманітні *Web*-аплети, вносячи нові ідеї в свої додатки завдяки використанню скриптів, форм та інтегрованості з серверами (наявності засобів підключення серверних можливостей).

*Flash* суміщає переваги (чіткість та гнучкість) ілюстраційної векторної графіки з растром (бітмапами), аудіо, анімацією та багатими інтерактивними ідеями, дозволяючи створювати *Web*-сайти на високому рівні, недоступному раніше при роботі з іншими пакетами. Це приваблює велику кількість відвідувачів цих сайтів.

Пакет *Flash* надає зручне середовище розробки, є зручним в управлінні, легко інтегрується в існуючий процес створення *Web*-вузла, будучи інтегрованим з іншим ПЗ фірми *Macromedia*.

Наприклад, *Flash* дозволяє імпортувати дані з *Macromedia FreeHand* та *Fireworks*, а його можливості збільшуються внаслідок стовідсоткової інтеграції з *Macromedia Generator*.

Визначні властивості *Flash* встановлюють новий *Web*-стандарт, яким користуються більше, ніж півмільйону дизайнерів у всьому світі.

**Підсумкові нотатки.** У більшості галузей сучасної ілюстраційної комп'ютерної графіки вироблено усталені стандарти – формальні або фактичні: *Adobe PhotoShop* – для обробки растрових зображень; *Adobe Illustrator* та *CorelDraw* – для роботи з векторною графікою; *AutoDesk/Kinetix 3D Studio MAX* та *Alias/Wavefront/TDI Maya* – в світі 3D-графіки; *Adobe Premiere* – для відеодизайну; *Macromedia Director*, *Microsoft PowerPoint*, *Hyper Method* – у сфері мультимедіа; *Adobe ImageReady*, *Flash* – для *Web*-дизайну; *Adobe PageMaker* та *Corel Ventura* – у верстанні та видавництві тощо.

Для роботи програм ілюстраційної комп'ютерної графіки дуже часто потрібне сканування. Відскановані зображення та образи текстових документів записують у вигляді файлів значного обсягу. Чим вищі дозвіл та розрядність подання кольору, тим файл більший.

Зокрема, оригінал 13×18 см при дозволі 300 точок/дюйм дасть нестиснені файли такого обсягу: при монохромному (чорно-білому) режимі та 1 біт/піксель – 399 *Kb*; при відтінках сірого кольору та 8 біт/піксель – 3,11 *Mb*; при *RGB* та 24 біт/піксель – 9,34 *Mb*.

Тому обов'язковою складовою частиною типового прикладного ПЗ сканеру (пакетів *Corel PhotoPaint*, *Adobe PhotoShop* тощо) та ПЗ корпоративних документних сканерів є алгоритми

стискання даних. З цієї ж причини сканеру необхідний комп'ютер із пам'яттю достатньої ємності, якісною графічною платою та містким жорстким диском, який вільно може працювати з пакетами типу *Adobe PhotoShop*, не перевантажуючись інформацією [176].

Імпорт/експорт та *OLE* (*Object Linking and Embedding*, зв'язування та впровадження об'єктів) являють собою два способи обміну інформацією між додатками. Різниця між ними полягає в методи, що використовують для обміну.

При експорті або імпорті файлу, його перетворюють до формату, зрозумілого для додатку, до котрого цей файл буде поміщено. Тобто в комп'ютерній системі для кожного формату файлу повинен бути встановлений свій фільтр.

При використанні *OLE* піклуватися про фільтри та формати непотрібно: якщо цей метод підтримують використані додатки, то між ними може вільно здійснюватись обмін інформацією.

### Питання для самоперевірки

1. Коротко охарактеризуйте можливості пакетів для створення 3D-графіки, анімації, морфінгу, дизайну та відеодизайну.
2. Окресліть загальну схему реалізації 3D-моделювання, реалістичної візуалізації та анімації об'єкту.
3. Сформулюйте сутність та виконайте порівняльний аналіз найвідоміших методів анімації 3D-об'єктів.
4. Які особливості має фрактальна графіка?



5. Проаналізуйте базові способи виконання мультимедіа-дизайну та назвіть основні класи широко розповсюджених на практиці мультимедійних додатків.

6. Наведіть класифікацію графічних пакетів для *Web*-дизайну.

7. Назвіть програми, що є фактичними стандартами для провідних напрямків практичного застосування ілюстраційної комп'ютерної графіки.

#### 11.4. Швидке формування ділової документації та рекламної продукції

Системи ділової комп'ютерної графіки дозволяють навіть недосвідченому користувачу швидко формувати ілюстровані ділові доповіді, прес-релізи, презентації, візитні картки, бланкову та рекламну продукцію тощо.

Розглянемо можливості цих систем на прикладі провідних програмних пакетів ділової комп'ютерної графіки *Excel*, *Word* та *PowerPoint* – складових частин популярного пакету *Office*.

***Microsoft Excel***. Є однією з найпотужніших програм для створення електронних таблиць та роботи з ними, сильна сторона якої полягає в таких можливостях [3, 22, 40, 52, 57, 67, 84, 95, 134, 170]:

– здатності виконувати різноманітні обчислення, глибокий аналіз даних та отримувати в підсумку нову корисну інформацію;

– наявності розвинених інструментів графічної візуалізації та редагування, анімування графіки, інтеграції з *Internet* (включаючи використання в процесі створення *Web*-публікацій);

– наявності дружнього інтерфейсу, розвиненого сервісу та надійних засобів захисту даних;

– постійному розвитку функціональних можливостей.

У процесі графічного подання даних в *Excel*, у центрі уваги знаходиться створення та редагування звичайних та складних діаграм. *Excel* надає досконалі засоби для роботи з ними.

**Ось базові методи, за допомогою яких *Excel* дозволяє графічно подавати числові дані та використовувати це подання:**

- використання набору вбудованих стандартних та нестандартних типів діаграм за основу для створення власних;

- побудова впроваджених діаграм (розміщуються в робочому аркуші разом із даними, використаними для побудови) або діаграмних аркушів (займають у робочій книзі окремий аркуш);

– створення діаграм, заданих по умовчання, натисканням швидкої клавіші або комбінації цих клавіш;

– використання *Майстра діаграм*, який веде крізь всі етапи побудови діаграми, виконуючи більшу частину роботи автоматично;

– друк побудованих діаграм на окремому аркуші або разом із даними робочого аркушу;

– змінювання діаграми (цілком або частково) згідно вашим вимогам, а саме редагування (додавання, вилучення та змінювання) об'єктів, із яких складається діаграма (легенди,

заголовків, ліній тренду тощо), та рисунків (об'єктів, які додаються до діаграми за допомогою панелі інструментів *Рисуння*, таких як стрілки, надписи в текстових полях, фігури, об'єкти *WordArt* тощо);

– використання для редагування діаграм панелей інструментів *Діаграми*, *Рисуння та Форматування*, що дозволяють швидко отримати доступ до параметрів діаграми та помістити в неї додаткові об'єкти;

– покращення зовнішнього вигляду діаграм та надання їм професійного вигляду за допомогою вбудованих засобів форматування (вставка додаткових об'єктів; відображення числових значень точок даних у підписах або за допомогою таблиці даних під діаграмою; додавання надписів з поясненнями найважливіших моментів; вибір кольору, текстури та рисунку заливки елементів діаграми тощо);

– застосування спеціальних засобів та методів для професіоналів (створення комбінованих діаграм, в яких порівнюються дані різних типів; використання рядів даних для побудови ефектних комбінованих діаграм; створення колових діаграм із вторинною гістограмою чи коловою діаграмою; використання рисунків як маркерів точок даних у графіках або як заповнювача в гістограмі та лінійчастій діаграмі; побудова діаграм на основі структурованих аркушів, управляючи відображенням даних за допомогою структури аркушу тощо);

– використання даних *Excel* в інших додатках *Windows* за допомогою технологій пересування та копіювання даних з файлу в файл із використанням буферу обміну та робочого столу, зв'язування та впровадження даних, включення даних *Excel* у повідомлення електронної пошти, створення підшивки

документів, отриманих засобами різних додатків *Microsoft Office* в рамках єдиного проекту;

– здійснення обміну даними *Excel* у мережі (сумісне використання файлів *Excel* різними користувачами, з наданням їм необхідного доступу до документів, доступ до даних *Web*-вузла *Internet* або мережі *intranet* вашої компанії на основі *Web*-запиту);

– здійснення обміну (імпортування/експортування) даних у форматі *Excel* з іншими програмами, наприклад, сумісне використання даних мережних серверів.

**Microsoft Word.** Оскільки ділова графіка передбачає створення графіків, діаграм, таблиць, звітів, рисунків та тексту з формулами, то гарні можливості розробки ілюстрованої ділової документації надає текстовий процесор *Microsoft Word*, який пропонує користувачам такі засоби ефективної роботи [5,26,87]:

– набір графічних засобів для легкого оздоблювання тексту та рисунків шляхом додавання до них об'єму, тіні, текстурних та прозорих заливок, автофігур;

– засоби спрощення роботи з таблицями, межами та заливою;

– засоби побудови типових класів діаграм;

– колекція професійно виконаних рисунків у складі *Microsoft ClipArt Gallery* (у комплекті постачання) для оформлення документів (зображення людей, будівель, елементів інтер'єру, географічних карт тощо);

– можливість вставки рисунків та відсканованих фотографій з інших програм та файлів;

- можливість розпізнавання розповсюджених графічних файлових форматів;
- доступність графічних фільтрів, встановлених для інших програм пакету *Office*;
- набір різноманітних засобів роботи з *Web* та *Internet* (наприклад, функція "Зв'язок із *Web*", що є загальною для додатків *Microsoft Office*, підтримує створення та перегляд гіпертекстових документів у мережі *Internet* та *Web*);
- засоби підвищення продуктивності та ефективності роботи членів робочих груп.

**Microsoft PowerPoint.** Цей програмний пакет надає сучасні можливості рисування та створення ділової графіки, дозволяючи виконувати такі дії [4,144,165]:

- швидко будувати діаграми широкого діапазону типів;
- використовувати автофігури (елементарні геометричні фігури, з'єднувальні лінії, фігурні стрілки, символи блок-схем, зірки, лозунги, виноски, керувальні кнопки та інші) для швидкої побудови схематичних рисунків та діаграм, створення блок-схем, анотування документів, додавання кнопок навігації та інших потреб;
- використовувати рисунки з колекції *ClipArt Gallery* (у комплекті постачання);
- здійснювати швидкий та якісний імпорт/експорт графіки без жодних викривлень;
- використовувати функцію автографіки, яка автоматично знаходить в презентації ті ключові слова, що важко відобразити за допомогою рисунку, та пропонує графічні елементи зі стандартної колекції зображень для надання слайдам професійного вигляду;

- креслити охайні криві лінії з точністю до точок за допомогою інструментарію кривих Безьє;
- здійснювати точний контроль товщини ліній та гнучке управління цією товщиною (можливі автоматичні настройки та настройки користувача);
- застосовувати в діаграмах та блок-схемах з'єднувальні лінії (прямі, ламані чи криві), що автоматично змінюють положення в разі пересування фігур, залишаючись прикріпленими до них;
- застосовувати такі типи стрілок, які дозволяють змінювати висоту та ширину їх гострого кінця;
- робити плаский текст та графічні 2D-зображення реалістичнішими, додаючи ефекти об'ємності, тіні (з перспективою, регулюванням глибини та кута падіння), заливку (текстуру, імпортований рисунок чи прозорий фон);
- рівномірно розставляти графічні об'єкти вздовж горизонтального та вертикального напрямків слайду;
- підвищувати якість зображення регулюванням його яскравості та контрастності;
- перетворювати кольори фону в прозорі області з тим, щоб вставлене растрове зображення виглядало природною складовою частиною дизайну слайду чи *Web*-сторінки.

Ці можливості доступні з панелі інструментів "*Рисуння*" та є тим набором інструментарію, що використовують всі додатки *Office*.

*PowerPoint*, як і всі інші додатки *Office*, використовує програму побудови діаграм *Graph*. З кожною версією цієї програми, її функціональні та графічні можливості вдосконалюються, стають зручнішими в експлуатації. Наприклад: розширюється набір типів діаграм; підтримані осі зі шкалою

часу (діаграми можна автоматично групувати та аналізувати в хронологічному порядку); можна застосовувати до елементів діаграм анімаційні ефекти та спецефекти, що підвищує наочність подання даних; можна змінювати масштаб шрифтів та розвертати текст уздовж осей діаграм для зручності перегляду даних; до звітів можна вносити пояснення до діаграм, є можливість безпосередньо приєднувати таблиці даних, зображених на діаграмі тощо.

*PowerPoint дозволяє створювати повноцінні інтерактивні анімовані презентації з елементами мультимедіа та гіпермедіа, спецефектами, якісним імпортом/експортом графіки, використанням можливостей служби Web та мережі Internet.*

У додатках *Office* кольори фігур та об'єктів обираються з однієї сорокакольорової палітри, що гарантує узгодженість кольорів при копіюванні та вставці фрагментів із одного додатку в інший. Це стосується і презентацій, виконаних у *PowerPoint*.

У *PowerPoint* підтримані найпопулярніші файлові графічні формати: *JPEG, WMF, EPS, PICT, GIF* та інші.

Розглянемо, що являють собою презентація в *PowerPoint* та процес її створення (далі деяку інформацію буде конкретизовано для певної версії *PowerPoint*, але можливі розбіжності з іншими версіями не носять принципового характеру для розуміння функціональних можливостей *PowerPoint*).

Підготування презентації (виступу презентаційного, рекламного характеру в вигляді тривалої ділової доповіді чи короткого ділового повідомлення) з використанням комп'ютерних технологій стало звичайною справою в бізнесі, підприємництві та менеджменті, в проведенні семінарів, конференцій та виставок, в навчальному процесі тощо.

Так, комерційний успіх підприємства суттєво залежить від того, як його представник презентує діяльність та продукцію потенційним замовникам. Для цього потрібно [78]:

- подати якнайбільше цікавої та привабливої інформації про продукцію (товари, послуги, ідеї тощо);
- запевнити замовників, що вони мають справу з солідним та надійним партнером, робота з яким не викличе проблем.

Цю проблему можна легко вирішити за допомогою *PowerPoint*, графічного пакету для підготування презентацій та слайд-фільмів, розробивши з його допомогою електронну презентацію з такими складовими частинами [3, 60-62, 65]:

- електронний інтерактивний презентаційний ролик (мультимедійний, гіпермедійний та анімаційний), що складається з набору слайдів та спецефектів, які супроводжують їх показ (слайд – кадр презентації, що містить заголовок, зміст, текст, графіку);
- електронна версія конспекту доповіді, при друкуванні якого на кожній сторінці буде виведене зменшене зображення слайду та текст, який пояснює його зміст;
- електронна версія плану доповіді (структури презентації – документу, що містить заголовки слайдів та основний текст без графічних зображень та спеціального оформлення);
- електронна версія роздаточного матеріалу (віддрукованих в компактному вигляді слайдів презентації по два, чотири чи шість на одній сторінці).

Слайди, створені засобами *PowerPoint*, можна демонструвати на екрані ПК або застосовувати для проведення традиційних презентацій (демонструвати на кольорових рідиннокристалічних панелях, друкувати на папері чи прозорих плівках



за допомогою кольорового чи монохромного принтеру, отримувати в спеціальних агенствах 35-міліметрові слайди на фотоплівці).

*PowerPoint* пропонує для різних класів користувачів різні технології створення презентації:

– за допомогою Майстра автозмісту, що дозволяє обрати за основу один із численних зразків готових презентацій з типовими темою, змістом та оформленням (для визначення стратегічного напрямку роботи фірми, для проведення наради на підприємстві чи в організації, для використання в мережі *Internet* тощо);

– шляхом вибору одного з численних шаблонів оформлення, що визначають композицію презентації, але не містять її наповнення (змісту), та заповнювання цього шаблону потрібною інформацією;

– на основі структури, імпортованої з іншого програмного додатку (наприклад, з *Word*);

– починаючи з порожньої презентації, в якій не задано ні оформлення, ні змісту.

Новачкам у підготованні рекламних кампаній та доповідей, що не до кінця визначилися з тим, що їм потрібно, найкраще користуватися *Майстром автозмісту*.

*Майстер автозмісту* допомагає вибрати один із численних вбудованих шаблонів змісту, що охоплюють великий спектр тем (наради робочих груп, інформаційні кіоски, дипломи, афіші, календарі подій, домашні *Web*-сторінки, шаблони від компанії *Dale Carnegie Training* із рекомендаціями щодо підвищення якості презентацій тощо), та пропонує корисні ідеї

щодо презентації, її початковий текст, форматування та принципи організації слайдів.

*PowerPoint* надає майстрів до кожної ключової компоненти презентації: слайд-майстер, майстер структури презентації, майстер роздаточного матеріалу, майстер приміток.

Рисунки та текст, які користувач розташує на цих майстрах, автоматично з'являться на кожному новому слайді та на сторінках приміток презентації.

Слайд-майстер (*Slide Master*) містить форматовані рамки фіксації (*Placeholders*), які з'являються на слайді до того, як ви почнете вводити в них заголовок, текст або фоновий рисунок.

Рамки фіксації оточені лініями, що складаються з точок. Використовуючи елементи управління рамок фіксації, можна: змінити їх розмір; пересунути їх вздовж слайду; змінити положення зображення або тексту всередині рамки, шрифт та колір тексту тощо. Якщо виконати зміни на слайд-майстрі, їх буде автоматично застосовано до всіх слайдів, настроєних по ньому.

*PowerPoint* дозволяє опрацьовувати та переглядати інформацію в різних видах її подання на екрані. Кожен із цих видів підвищує зручність роботи при виконанні певних дій: введенні тексту з розглядом його структури, створенні заміток, розташуванні в слайді графіки тощо. Конкретний вид можна встановити, натиснувши кнопку внизу головного вікна програми або за допомогою команд меню *Вид*:

– вид слайдів найзручніший для поступового формування кожного слайду, вибору для нього оформлення, вставки тексту та графіки;

– вид структури встановлюється для роботи над текстом презентації та дозволяє переглядати заголовки всіх слайдів, весь текст та структуру презентації;

– вид сортувальника слайдів найзручніший для додавання переходів, встановлювання тривалості перебування слайду на екрані, переставляння слайдів;

– вид заміток призначено для створення заміток до доповіді;

– вид демонстрації призначено для перегляду результатів роботи в режимі почергового виведення слайдів на екран.

Основні етапи створення нової презентації за допомогою *Майстра автосмісту* такі:

– запустіть програму *PowerPoint*, що викличе появу головного вікна *PowerPoint* та діалогового вікна *Корисна порада*;

– оберіть з меню *Файл* опцію *Новий*, що приведе до появи вікна *Нова презентація*;

– оберіть селекторну кнопку способу створення презентації за допомогою *Майстра автосмісту*, що приведе до послідовної появи діалогових вікон, в яких можна задати основні характеристики презентації;

– у вікні введення даних для оформлення слайду-заголовку (титального слайду презентації) введіть інформацію про розробника презентації, назву фірми, її девіз, тему презентації (доповіді, виступу) тощо;

– у вікні вибору потрібного типу (виду) презентації (рекомендування стратегії; продаж продукту, послуги чи ідеї; навчання; звіт про досягнення; повідомлення про погані новини тощо) виберіть відповідну позицію;

– у наступному вікні *Майстра автозмісту* виберіть потрібний стиль оформлення презентації та задайте тривалість виступу;

– у черговому вікні *Майстра автозмісту* виберіть спосіб видачі презентації та вкажіть, чи потрібен роздагочний матеріал;

– в останньому вікні *Майстра автозмісту* розробнику повідомляється про завершення попередньої роботи зі створення презентації, так що після натискання на кнопку *Готово* він зможе (після деякої затримки) отримати титульний слайд;

– збережіть презентацію у відповідній папці за допомогою опції *Зберегти* меню *Файл*.

Якщо у вікні вибору типу презентації нема потрібної теми, натисніть кнопку *Інші* для отримання переліку шаблонів презентацій. Після вибору певного шаблону, ви одразу потрапите в останнє вікно *Майстра автозмісту*.

Процес створення презентації на основі запропонованого змісту та оформлення може розвиватися інакше, наприклад: обрати в меню *Файл* команду *Створити*, вибрати вкладку *Презентації*, двічі клацнути кнопку *Майстра автозмісту* та слідувати подальшим його вказівкам. *PowerPoint* створить зразок презентації, в який потім можна додати власні текст та графічні зображення, та відобразить його в режимі структури. Потім потрібно: двічі клацнути *Слайд 1* для переходу в режим слайдів та ввести назву презентації; використовуючи полосу прокручування, пройтися по іншим слайдам та ввести замість зразків свої текст та графіку; вибрати в меню *Файл* команду *Зберегти*, дати презентації ім'я та зберегти її; для перегляду

створеного показу слайдів натиснути кнопку *Показ слайдів* у лівому нижньому куті вікна *PowerPoint*.

Розглянемо випадок створення презентації для показу в глобальній мережі *Internet*.

У комплекті *PowerPoint* є набір особливих розміток, призначених для створення інтерактивних презентацій, які можна зберегти в форматі *HTML* для подальшого перегляду в *Internet*. Початок створення презентації буде таким, як раніше. Потім потрібно: клацнути на *Виді презентації*, вибрати потрібний вид та натиснути кнопку *Далі*; в групі *Виберіть вид презентації* встановити перемикач *Internet, кіоск* та натиснути кнопку *Далі*; виконати інші вказівки *Майстра автотворення*. *PowerPoint* створить зразок презентації, в який можна включити потрібні текст, рисунки, анімацію, гіперпосилання. Щоб зберегти презентацію в форматі *HTML*, оберіть в меню *Файл* команду *Зберегти в форматі HTML*.

*Шаблон (Template)* – це презентація, формат та схему кольорів якої можна використати за основу для власної презентації.

Шаблон містить слайди, оформлені певним чином: у полі слайдів розташовано заглушки, в які можна вставити текст, графіку, таблиці, діаграми.

Можна змінювати дизайн шаблону за своїм смаком.

*PowerPoint* надає велику кількість професійно оформлених шаблонів презентацій на різні теми в комплекті постачання.

Є багато різноманітних шаблонів дизайну, що застосовуються до слайдів, наприклад, анімаційні шаблони з готовою анімацією.

Типова схема створення *PowerPoint*-презентації з використанням шаблону оформлення така:

- обрати в меню *Файл* команду *Створити* та перейти на вкладку *Оформлення презентацій*;

- переглянути за допомогою полоси прокручування шаблони оформлення та вибрати потрібний;

- переглянути за допомогою полоси прокручування варіанти розмітки та вибрати розмітку для титульного слайду; ввести заголовок презентації та інші відомості, що повинні бути на титульному слайді;

- для кожного з наступних слайдів виконати певні дії (натиснути на панелі інструментів *Стандартна* кнопку *Створити слайд*, обрати розмітку слайду та додати його вміст);

- обрати в меню *Файл* команду *Зберегти*, дати презентації ім'я та натиснути кнопку *Зберегти*;

- у лівому нижньому куті вікна *PowerPoint* натиснути кнопку *Показ слайдів* для перегляду презентації.

Розглянемо технологію створення презентації методом імпортування структури.

Зручно створювати слайд на основі документу, який було відформатовано за допомогою стилів заголовків, подібних до тих, які застосовуються в *Microsoft Word*. Послідовність дій буде такою: вибрати в меню *Файл* команду *Відкрити*, в списку *Тип файлів* клацнути *Всі структури*, в списку *Ім'я* двічі клацнути документ, який використовується.

Після цього імпортована структура відкриється в *PowerPoint* у режимі структури. Кожний основний заголовок документу перетвориться в заголовок окремого слайду. Кожний пункт структури буде відображено як текст слайду.

Ще зручніше імпортувати власне документ *Word*, у якому використано стилі заголовків: потрібно відкрити документ у середовищі *Word*, обрати в меню *Файл* команду *Відправити* та клацнути *Microsoft PowerPoint*. Абзаци, відформатовані в стилі "Заголовок 1", перетворюються в заголовки слайдів. Абзаци, відформатовані в стилі "Заголовок  $N$ " ( $N \geq 2$ ), перетворюються в тексти  $N$ -го рівня.

Далі наведемо перелік термінів та визначень.

Установочна презентація (*PowerPoint default presentation*) використовується за шаблон по умовчанняю. Для роботи з нею виберіть перемикач *Порожня презентація (Blank Presentation)* у діалоговому вікні *Нова презентація (New Presentation)*.

Об'єкт (*Object*) уособлює в собі: тексти, лінії та форми, створені за допомогою інструментів роботи з текстом та рисування геометричних фігур; зображення, імпортовані з інших застосувань.

Атрибут (*Attribute*) – властивість об'єкту (тип обмежувальної лінії, заливка, тінь, колір, форма тощо).

*Форма (Shape)* – атрибут об'єктів, створених із використанням інструментів роботи з текстом або *Автоформ (AutoShapes)*.

Об'єкти, створені за допомогою інструментів рисування ліній, дуг та довільних фігур, не мають атрибуту *Форма*.

*Контур (Border)* – видима лінія по краях об'єкту.

Кольорова схема (*Color Scheme*) – основа з восьми кольорів, яку можна застосовувати в слайдах, сторінках приміток та роздаточному матеріалі. Вона містить кольори фону, ліній, тексту та шість додаткових кольорів, баланс яких покращує сприйняття слайдів.

Інші кольори (*Other Colors*) – ті, що відсутні в основній схемі.

Абзац (*Paragraph*) – текст, введений між двома натисканнями клавіші *Enter*.

Коли здійснюють вирівнювання тексту та додавання символів бюлетеню, дії застосовуються до абзацу як до єдиного цілого.

*Інструмент Текст (Text Tool)* панелі інструментів *Рисуння (Drawing)* призначений для введення тексту в будь-якому місці слайду. Цей текст не буде виводитися на екран у режимі роботи зі структурою презентації (*Outline View*).

Якщо ви маєте здібності дизайнеру та достатньо часу, можна створювати презентацію "з нуля", без використання майстрів та шаблонів.

Для цього є потужні функції роботи з текстом, засоби для рисування та імпортування готових рисунків, набір стандартних ілюстрацій, можливості вставки таблиць, інструментарій побудови діаграм, засоби створення анімації та багато іншого, що перетворює слайдові демонстрації в цікавий та привабливий засіб активізації продажу споживачам продукції.

*PowerPoint* легко використовувати, а виконує він величезний обсяг рутинної роботи.

***Наведемо рекомендації з підвищення ефективності введення та оформлення тексту, створення текстових ефектів у PowerPoint.***

Користуючись Майстром автоформи, можна вводити текст у кожен слайд презентації. Але якщо презентацію подано в вигляді структури, працювати з текстом зручніше (його легше переглядати та виправляти в одному вікні).



Якщо презентацію подано в вигляді структури, зліва на екрані розташовано панель структури з кнопками для підвищення чи зниження рівня заголовків, пересування блоку тексту вгору чи вниз, розгортання чи згортання структури.

Загальні прийоми роботи зі структурою аналогічні роботі з *Word* у режимі структури документа.

Додатково відзначимо таке: для пересування слайду разом зі змістом у потрібну область документа, перетягніть значок слайду; для створення нового елемента списку встановіть курсор у кінець списку та натисніть кнопку вводу; для пересування елемента списку розташуйте курсор зліва від нього та перетягніть елемент у потрібну позицію; для створення слайду створіть елемент списку та натискайте кнопку *Підвищити рівень* доти, доки маркер не перетвориться в значок слайду; щоб повернутися з виду структури до виду слайдів, двічі клацніть на значку відповідного слайду.

Дизайн презентації стане кращим, а її сприйняття легшим, якщо вдало вибрати шрифт тексту та текстові ефекти. *PowerPoint* надає багато цих можливостей. Розглянемо ті, що використовуються найчастіше.

Щоб задати для вибраних текстових елементів усієї презентації однаковий шрифт, потрібно: вибрати в меню *Вид* команду *Зразок слайду*; клацнути в тій області, де потрібно здійснити зміни; вибрати в панелі інструментів потрібну кнопку для задавання типу, розміру та нахилу шрифту.

Щоб змінити реєстр виділеного тексту, потрібно задіяти команду *Реєстр* меню *Формат*.

Можна задати для символів тексту тінь (кнопка *Тінь тексту*) чи рельєфне накреслення (команда *Шрифт* із меню *Формат*).

Щоб створити фігурний текст, символи якого розташовано вздовж кривої лінії, кола чи всередині вибраної фігури, потрібно: вибрати з меню *Об'єкт* команду *Вставка*; вибрати зі списку *Тип об'єкту* елемент *Microsoft WordArt* (на слайді з'явиться область розташування фігурного тексту та відкриється вікно для введення тексту); ввести текст; клацнути на кнопці *Поновити екран*, після чого текст з'явиться в області слайду.

Панель інструментів *PowerPoint* надає списки для вибору форми фігурного тексту, типу та розміру шрифту, кнопки для задавання текстових ефектів.

Вдалий дизайн презентацій дозволяє подати дані в привабливому вигляді, виділити найважливіші моменти, полегшити розуміння важких положень, справити враження на слухачів. Розглянемо можливості дизайну презентацій у *PowerPoint* [3, 29, 60-62, 65].

По-перше, *PowerPoint* надає велику кількість професійних шаблонів дизайну презентацій.

Для задавання одного з цих шаблонів потрібно: активувати команду *Застосувати шаблон дизайну* меню *Формат*; клацаючи на іменах шаблонів дизайну, переглянути їх зображення у спеціальному вікні попереднього перегляду; вибрати потрібний шаблон, клацнувши на його імені двічі.

По-друге, можна нарисувати графічний об'єкт будь-якої складності за допомогою інструментарію панелі *Рисуння* (вона відображається при роботі з видом слайдів або заміток) та додаткової панелі *Рисуння+* (для її відображення дос-

татньо встановити вказівник миші на будь-яку панель інструментів, натиснути праву кнопку миші та вибрати в контекстному меню команду *Рисуння*).

Основні прийоми рисування за допомогою цих панелей такі: для рисування графічного об'єкту потрібно натиснути відповідну кнопку на панелі інструментів один раз або двічі (якщо потрібне багаторазове відрисовування об'єктів одного типу чи виконання над об'єктами однакових дій); для виконання над графічним об'єктом будь-яких дій треба виділити його, клацнувши мишею (якщо об'єкт приховано за іншими об'єктами, виділіть об'єкт на першому плані та натискайте клавішу *Tab*; для виділення декількох об'єктів, послідовно клацніть на них мишею, втримуючи клавішу *Shift*; для виділення всіх об'єктів призначено комбінацію клавіш *Ctrl+A*); для рисування інструментами *Лінія*, *Прямокутник*, *Еліпс*, *Дуга* горизонтальних та вертикальних ліній, квадратів, кіл, дуг треба під час рисування утримувати клавішу *Shift*.

По-третє, в презентацію можна вставити рисунок із колекції *Microsoft ClipArt*. Найшвидшим способом доступу до цієї бібліотеки є натискання на панелі інструментів кнопки *Вставити графіку*.

Можна виконати перегляд: всіх рисунків, розташованих за алфавітом (в списку *Розділи* виберіть елемент *Всі розділи*); рисунків, які відповідають конкретній темі (виберіть цю тему зі списку *Розділи*). Для вставки рисунку в слайд, двічі клацніть на ньому. Після появи рисунку на слайді, можна змінювати його розмір та розташування.

По-четверте, в презентацію можна вставляти рисунки, діаграми, відскановані фотографії, таблиці, формули та деякі

інші графічні зображення, підготовані засобами інших програмних додатків (*Word, Excel, Photoshop, CorelDraw* тощо) або розміщені в зовнішніх сховищах інформації.

Після вставки зображень у презентацію, їх можна перетворити в об'єкти *PowerPoint* та відредагувати: за допомогою інструментів рисування внести зміни в кольори ліній та заливки, розташування, об'єднати зображення тощо; за допомогою панелі Інструментів *Настроювання зображення* (з'явиться на екрані, якщо виділити рисунок) відрегулювати яскравість та контрастність рисунку, кадриувати його, перефарбувати, обвести рамкою тощо.

Можливість включати в презентації діаграми, скановані рисунки, таблиці та формули забезпечено наявністю у складі *Power Point* спеціальних програмних додатків: *Organization Chart, Graph, Photo Editor, Equation Editor*. Вони створюють об'єкти, впроваджені в презентацію.

Сумісне використання даних додатками забезпечується таким потужним інструментом оболонки *Windows*, як технологія *OLE (Object Linking and Embedding* – зв'язування та впровадження об'єктів): вона дозволяє взяти графічну інформацію (файл) із одного додатку (наприклад, *Excel* або *Word*) і без викривлень та накладок помістити в інший додаток (наприклад, *PowerPoint*).

Якщо графічну інформацію впроваджувати, вона передається разом із документом, у якому зберігається.

Якщо цю інформацію зв'язувати, то її можна зберігати деінде та працювати з нею в батьківській програмі: кожне відкриття документу, що містить зв'язок, буде приводити до його автоматичного оновлення.

Наприклад, якщо в презентації створювати або змінювати вказані зв'язані об'єкти (діаграму, сканований рисунок, формулу), то меню та панелі інструментів *PowerPoint* замінюються на відповідний інструментарій батьківського застосування.

Графічні об'єкти, впроваджені в презентацію, можна редагувати, двічі клацнувши на них.

Якщо в презентації потрібно використати таблицю, її можна створити в *PowerPoint*: звернутися до кнопки *Створити слайд* та вибрати слайд із таблицею.

Але часто застосовують інший підхід – таблицю створюють в іншому додатку та вставляють в презентацію *PowerPoint*:

– по-перше, є широко відомі програми типу *Excel* та *Word*, які спеціально призначені для створення таблиць;

– по-друге, часто потрібно використовувати таблиці, що вже існують та були створені засобами *Word* або *Excel*.

– для копіювання існуючої *Word*-таблиці в електронну презентацію *PowerPoint* потрібно:

– виділити таблицю в *Word* та обрати опції *Правка, Копіювати* (комбінація клавіш *Ctrl + C*);

– переключитися в *PowerPoint*, перейти до потрібного слайду та переключитися в режим перегляду слайдів;

– вибрати *Правка, Вставити (Ctrl + V)* для того, щоб вставити таблицю з буферу.

Якщо доводиться часто повертатися назад та редагувати дані у вставленій таблиці, краще зв'язати *Word*-таблицю зі слайдом, використовуючи можливості технології *OLE*: виділити таблицю в *Word*, установити курсор на ній та вибрати *Таблиця, Виділити таблицю*; вибрати *Правка, Копіювати (Ctrl + C)*, після чого виділену таблицю буде скопійовано в

*Буфер обміну* програми *Windows*, який використовують за тимчасову пам'ять всі програми; запустити програму *PowerPoint* (або переключитися в неї, якщо вона виконується); переключитися в режим перегляду слайдів та знайти слайд, в який потрібно вставити таблицю (або створити новий слайд із використанням кнопки *Створити слайд*); обрати *Правка, Спеціальна вставка*; клацнути на *Об'єкт, Документ Microsoft Word* та натиснути селекторну кнопку *Зв'язати* (якщо ж вибрати кнопку *Вставити*, таблицю буде впроваджено та обірано всі її зв'язки з початковим документом); натиснути на кнопку *OK*.

Для зв'язування електронної таблиці *Excel* зі слайдом потрібно: відкрити її; виділити чарунки, які потрібно скопіювати в *Буфер обміну*, та натиснути *Ctrl + C*; переключитися в *PowerPoint*; у режимі сортування слайдів перейти до слайду, в який потрібно помістити електронну таблицю (або створити новий слайд); обрати *Правка, Спеціальна вставка*; натиснути на *Об'єкт, Аркуш Microsoft Excel* та на селекторну кнопку *Зв'язати* (якщо зв'язок не потрібний, натискають кнопку *Вставити*; ще краще замінити частину попередніх кроків натисканням *Ctrl + V*, забезпечуючи швидку вставку даних із *Буферу обміну*); натиснути кнопку *OK*.

Розглянемо можливості створення засобами *PowerPoint* анімаційних ефектів, мультимедіа, гіпермедіа, взаємодії з *Web* та *Internet* [4, 144, 165]:

– є набір анімаційних шаблонів (професійно виконаних шаблонів оформлення презентацій з анімаційними ефектами), що зберігаються в зразку слайдів та автоматично додаються в нові слайди;

– є колекція кліпів, що містить графічні елементи, звуки, відеокліпи та графічні зображення, зручні для створення презентацій з елементами мультимедіа;

– вбудоване управління файлами *AVI* дозволяє запускати файли кіно безпосередньо з *PowerPoint* (це підвищує якість та розширює можливості управління кольорами);

– основою для створення власної анімації тексту та об'єктів є кнопки панелі інструментів *Ефекти анімації* (до стандартних кнопок можна додавати власні, призначені для запуску іншого додатку чи презентації, впровадженої в слайд, для реалізації переходу на певний слайд під час показу тощо);

– команда *Настроювання анімації* меню *Показ слайдів* дозволяє швидше визначати та переглядати анімаційні ефекти (в одному діалоговому вікні можна задати анімаційний ефект, змінити порядок показу анімації, здійснити хронометраж анімації, переглянути анімаційний ролик);

– анімаційні ефекти можна застосовувати до окремих елементів даних діаграми, акцентуючи на них додаткову увагу;

– можливість голосового супроводження дозволяє провести показ слайдів у автоматичному режимі та розіслати презентацію іншим користувачам мережею або на диск;

– надбудова *Music Tracks*, базовими даними для якої є стиль, тривалість та темп музики, допомагає виділити анімаційні ефекти, ввести фонову музику чи звук без суттєвого збільшення розміру файлу, забезпечити різне музичне супроводження для різних слайдів;

– програма *PowerPoint Viewer* дозволяє здійснювати показ слайдів на тих комп'ютерах, де не встановлено *PowerPoint*, а

також підтримує захист кіосків за допомогою паролів та гіперпосилань на інші слайди та на адреси *URL* у службі *Web*;

– засіб перегляду *Web*, який використовують всі додатки *Microsoft Office*, дозволяє переглядати презентації та документи на локальному ПК, сервері, мережі *Internet* або в службі *Web*;

– технологію пошуку *Web* (зі складу *PowerPoint*) можна використовувати для швидкого пошуку в локальних та корпоративних мережах документів формату *HTML* та документів, створених засобами *Microsoft Office*;

– у презентації можна визначати гіперпосилання на графічні об'єкти (фігури, рисунки, таблиці, кнопки дій тощо) та тексти, за якими можна переходити в інші слайди та презентації, *Word*-документи, адреси *Internet* (корисним доповненням до цього є вбудовані об'ємні керувальні кнопки "*Вперед*", "*Назад*", "*Основний слайд*", "*Довідка*", "*Відомості*", "*Звук*" та "*Відеокліп*", натискаючи на які під час показу, можна запустити інший додаток, відтворити звук та кіно, перейти до інших слайдів, файлів або *Web*-сторінок);

– можна зберегти презентацію в форматі *HTML*, перетворивши її в *Web*-сторінку, або подати її як файл анімації *PowerPoint*, впровадити його в документ *HTML* та автоматично запустити показ слайдів у захищеному вікні на сторінці *HTML* за допомогою *PowerPoint Animation Player* (розширення браузера *Internet*);

– команда *Настроювання дії* меню *Показ слайдів* дозволяє визначити для певного об'єкту інтерактивні дії, що будуть виконуватися після вказування на нього курсором або клацання мишею (в одному діалоговому вікні можна вказати місце



призначення гіперпосилання, вибрати завантажуваний додаток, задати звук, кіно чи перехід до іншого слайду тощо);

– можна закріпити дію (наприклад, перехід за адресою *URL* на *Web*-сервер) за певним місцем на слайді, що після збереження слайду як документу *HTML* перетворить цю "точку" на слайді в карту зображень, яку зв'язано з певними місцями в системі *Web*.

Розглянемо методи досягнення одноманітності в дизайні презентацій за допомогою управління зовнішнім виглядом слайдів:

- за допомогою зразків слайдів, будь-яка зміна в яких буде відображатися в кожному слайді, що будуть створювати (зразки слайдів містять елементи фону та визначають формат та розміщення заголовків і тексту, що вводять в слайди; зразки заголовків визначають формат та розміщення титульних слайдів);

- за допомогою кольорової схеми, яку можна обирати для презентації в цілому або для окремих слайдів (ця схема – набір із восьми гармонійно підібраних кольорів, які стають основними кольорами для тексту, фону, заливки, акцентів тощо; кожен із кольорів схеми автоматично відводиться під певний елемент слайду);

- за допомогою шаблонів оформлення, що містять гармонійні кольорові схеми (на вибір пропонується декілька схем), зразки слайдів та заголовків із нестандартним форматуванням, стилізовані шрифти для конкретних видів оформлення (в складі *PowerPoint* є професійно виконані шаблони, але можна створити власний шаблон, зберігши будь-яку презентацію як шаблон разом із зразками заміток та видачі).

Для перевірки узгодженості та стилю презентації: відкрийте її; виберіть у меню *Сервіс* команду *Перевірка стилю*; виберіть потрібні параметри перевірки (за необхідності натисніть кнопку *Параметри* для точнішого їх задавання); натисніть кнопку *Почати*.

Командою *Презентація PowerPoint Central* із меню *Сервіс* можна запустити електронний журнал для користувачів, який містить: статті та рекомендації з підвищення якості презентацій; гіперпосилання на сервери *Internet* та в папку *Office Value-Pack* (диск *CD-ROM*), де є нові колекції графічних зображень, відеокліпів та аудіокліпів, графічних ефектів, шаблонів та порад.

## Питання та завдання для самоперевірки

1. Охарактеризуйте переваги, що надають для реалізації ділової графіки складові частини пакету *Microsoft Office* – електронна таблиця *Excel*, текстовий процесор *Word*, пакет для створення мультимедійних електронних презентацій *PowerPoint*.

2. Які засоби імпорту/експорту графіки та інтеграції з *Internet* надають *Microsoft Excel*, *Word* та *PowerPoint*?

3. Проаналізуйте, які системи ілюстраційної та ділової комп'ютерної графіки та для розв'язування яких задач використовуються на практиці на підприємствах, в установах та організаціях за вашим місцем проживання, навчання або роботи. Висуньте ваші пропозиції щодо вдосконалювання тієї практики, що склалася.

## Навчальний практикум до гл.11

### Частина 1

Використовуючи інструментальні засоби систем ілюстраційної комп'ютерної графіки та дизайну, необхідно розв'язати такі типові прикладні задачі та тренувальні вправи.

**1-а група завдань.** Необхідно здійснити такі типові обчислення [63]: а) виконати розрахунок частоти сітки (лінійатури) растру, потрібної для друку зображення на заданому пристрої виведення, з заданим дозволом та певними вимогами до якості; б) обчислити дозвіл, потрібний для оцифрування слайду на 35-міліметровій плівці, що необхідно надрукувати на аркуші заданого формату, з заданою частотою сітки растру та вимогами до якості; в) обчислити кути обертання растру, потрібні для підготування на ПК кольороподілених зображень для подальшого їх виведення на ФНА, з урахуванням друку фарбами, кольори яких є додатковими стосовно кольорової моделі *CMYK* (якщо в процесі виконання цих обчислень у вас виникне потреба в додатковій теоретичній інформації, зверніться до навчального посібника [63], а початкові дані для обчислень вам надасть викладач).

**2-а група завдань.** Користуючись інструментарієм растрових графічних редакторів (наприклад, редактору *Adobe Photoshop*), потрібно виконати такі тренувальні вправи [63].

2.1. Відкрити файл із растровим зображенням (цифровою фотографією), запропонованим вам викладачем, та покращити якість цього зображення, змінюючи динамічний діапазон

(розкид між мінімальною та максимальною яскравістю) всього зображення та окремих його частин, та добиваючися при цьому найбільшої сбалансованості яскравості та контрастності зображення.

2.2. Відретушувати зображення, а саме виконати по відношенню до цього зображення локальне згладжування (розтушовування), освітлення, видалення (наприклад, "забивання" за допомогою штампю) непотрібних деталей та тіней, корекцію (покращення) різкості в потрібних частинах зображення.

Застосування інструментів місцевої корекції та ретуші повинне привести до: покращення сприйняття та підкреслювання важливих елементів зображення (наприклад, прорисовування тих деталей, що раніше були невидимі через зайву затемненість деяких ділянок фотознімку; підсилення рельєфності деталей; відновлення пошкоджених, запилених та постарілих частин фотознімку); видалення непотрібних деталей.

2.3. Застосувати фільтри для покращення якості зображення та підсилення того враження, яке повинне викликати це зображення, шляхом його ретуші, акценту на важливих деталях, імітації розташування зображення на різних матеріалах, змінювання умов освітленості та впливу інших художніх ефектів.

2.4. Здійснити обтравку (точне виділення) вказаного елемента зображення та зберегти обтравочний контур у тому ж файлі, окремо від основного зображення, за допомогою технології каналів.

Зауважимо, що застосування настройок динамічного діапазону, інструментів корекції та ретуші, фільтрів, каналів та

обтравочних контурів є потужною технологією редагування та композиції зображень, дозволяючи достатньо легко створювати складну та якісну рекламну та художню продукцію на професійному рівні.

2.5. Створити з застосуванням спеціальних ефектів складний колаж (композицію) з декількох заданих зображень для переднього плану та фону (ним можуть бути водойма, рослинність, зображення неба тощо), використовуючи для цього всі попередньо набуті вами знання, вміння та навички.

**3-я група завдань.** Використовуючи інструментарій векторних графічних редакторів (наприклад, редакторів *CorelDraw* або *Adobe Illustrator*), потрібно виконати такі тренувальні вправи [63].

3.1. На прикладі розв'язування задачі створення електронного оригінал-макету векторного рисунку для нанесення на тканину (згідно зі зразком, виданим викладачем, або за вашим власним дизайном), потрібно відпрацювати такі вміння та навички:

- налагоджування інтерфейсу програми та встановлювання потрібних параметрів документу;

- створення елементарних векторних об'єктів (прямих ліній, замкнених багатокутних контурів, еліпсів, кіл, спіралей, стандартних багатокутників), замкнених та незамкнених криволінійних контурів (ліній довільної форми; кривих, які базуються на математичних формулах, наприклад, кривих першого, другого та третього порядків);

- застосування інструментів редагування контурів (шляхом пересування, змінювання властивостей, створення нових та видалення зайвих опорних точок) та інструментів обробки

замкнених контурів (шляхом виконання їх суцільної, градієнтної та текстурної заливки, розмикання та розбиття);

– створення складних композицій векторних об'єктів шляхом застосування інструментів для виділення, групування та розгруповування об'єктів, операцій з контурами (об'єднання, перетину, виключення, застосування команд типу "Мінус верхній" та "Мінус нижній", комбінування контурів).

3.2. На прикладі розв'язування задач із наведеного нижче переліку (задачі 3.2.1-3.2.3), потрібно закріпити набуті вами (в процесі виконання вправ 3.1) знання, вміння та навички, важливі для формування таких прийомів: настроювання інтерфейсу векторного графічного редактору та параметрів документу; створення векторних об'єктів різних типів та змінювання їх властивостей; взаємного позиціонування, групування та виконання інших типових операцій з графікою та текстом; створення складних текстово-графічних документів; виконання засобами векторної графіки креслярсько-графічних робіт.

3.2.1. Рисування структурної схеми (наприклад, ієрархічної схеми підпорядковування відділів великого підприємства або навчальної установи).

3.2.2. Створення текстово-графічного документу для ілюстрації фізичного процесу (наприклад, побудови та текстового супроводження графіків змінювання метеорологічних параметрів).

3.2.3. Побудова точного креслення машинобудівної деталі (наприклад, кулькового підшипника).

**4-а група завдань.** Потрібно створити електронний оригінал-макет та друкований зразок художньої, дизайнерської, рекламної або видавничої продукції, що належить до одного з

наведених далі типових класів. При цьому необхідно: дотриматися вимог, поставлених викладачем; комплексно використовувати інструментальні засоби систем ілюстраційної комп'ютерної графіки; виявити свої індивідуальні художні та творчі здібності.

4.1. Внутрішній та зовнішній інтер'єри (включаючи обладнання, спорядження та засоби пересування), ландшафт довкілля для: а) житлового будинку; б) офісу; в) крамниці (супермаркету, кіоску, торгівельної точки тощо); г) установи суспільного харчування (кафе, ресторану, бару, їдальні, буфету тощо); д) вокзалу; е) гаражу; ж) складу; з) іншої суспільної, промислової або сільськогосподарської будівлі чи споруди.

4.2. Вбрання та аксесуари до нього, зачіска, макіяж та інші складові частини моделювання іміджу людини для: а) щоденного застосування; б) ділового виступу або презентації; в) свята; г) прийому; д) відвідування театру; е) показу мод; ж) балу-маскараду; з) спортивних занять.

4.3. Театральні декорації, афіші та реклама.

4.4. Дизайн тканини.

4.5. Дизайн друкованої продукції (книжки, журналу, газети, буклету, рекламного аркушу, візитної картки, щотижневика, календаря, альбому, листівки, конвертів та паперу для листів тощо).

4.6. Художня картина (у будь-якій техніці рисунка), репродукція або інший витвір мистецтва.

4.7. Продукція фотодизайну для прикрашання приміщень: фотокартини (портрети, пейзажі, натюрморти тощо), фотопалери та інша продукція.

4.8. Відеодизайн рекламного ролику.

## Частина 2

**5-а група завдань.** Використовуючи інструментальні засоби систем ділової комп'ютерної графіки, необхідно розв'язати такі типові прикладні задачі.

5.1. Створити ілюстровану електронну та традиційну друковану доповідь-звіт про підсумки поточної діяльності з початку календарного року фірми з продажу електронно-обчислювальної техніки за такими показниками діяльності цієї фірми: а) збільшення обсягу продажу; б) розширення асортименту послуг; в) збільшення кількості філій.

5.2. Розробити інтерактивну електронну та традиційну презентації для рекламування продукції та послуг фірми-ресторану швидкого приготування їжі за такими аспектами: а) асортимент продукції та послуг, тенденції його розширення; б) зростання обсягу продажу продукції та реалізації послуг; в) збільшення пропускної здатності виробничих площ (кількості відвідувачів за добу).

5.3. Отримати електронний та друкований прес-релізи фірми з виробництва дитячих іграшок за такими показниками діяльності цієї фірми: а) зростання кількості охоплених вікових дитячих груп; б) розширення тематики та спрямованості за функціональним призначенням іграшок та ігор; в) збільшення обсягу продажу.

5.4. Підготувати оголошення в електронній та друкованій газетах для заохочення до участі в аукціоні наступної спрямованості: а) благодійний аукціон; б) аукціон із продажу



підприємств та фірм, які збанкрутували; в) аукціон антикваріату.

5.5. Створити оригінал-макет рекламного буклету для художньої виставки, зупинившись на такому етапі його формування: а) макетування та брошурування оригінал-макету; б) публікування на *Web*-сторінці *Internet*; в) отримання віддрукованого оригінал-макету.

5.6. Створити електронний та друкований оригінал-макети рекламного оздоблення вітрин для крамниці такої спрямованості: а) канцелярські товари; б) аптечний кіоск; в) продовольчі товари; г) промислові товари; д) супермаркет.

5.7. Створити електронний та друкований інформаційні бюлетені медичної установи, присвятивши їх такій тематиці: а) профілактика хвороб (наприклад, грипу); б) пропагування нових ефективних ліків; в) інформування про спектр послуг зубних поліклінік міста.

5.8. Створити електронний та друкований оригінал-макети рекламного оздоблення холу офісу поліграфічної фірми за таким напрямком рекламної діяльності: а) рекламування асортименту послуг; б) рекламування зразків продукції; в) рекламування можливостей поліграфічного обладнання.

5.9. Підготувати електронну та друковану ділову доповідь для візуального супроводження виступу на телебаченні: а) про сучасні суспільні (економічні, соціальні, екологічні або інші актуальні) проблеми; б) про роботу комунального підприємства; в) про забезпеченість навчальних закладів міста обчислювальною технікою.

5.10. Створити електронний та друкований каталог та прайс-аркуш для виставки-розпродажу наступної спрямо-

ваності: а) виставка витворів мистецтва; б) виставка передових науково-технічних досягнень; в) виставка кулінарних виробів.

У межах цієї задачі необхідно:

– обґрунтувати вибір інструментарію розробки (апаратного та програмного забезпечення; сучасних комп'ютерних графічних, інтерактивних, об'єктно-орієнтованих, інформаційних, мультимедійних, гіпермедійних та мережних технологій);

– розробити загальну концепцію та комп'ютерну технологію розв'язування поставленої задачі, зробивши акцент на питаннях організації інтерфейсу з користувачем, візуалізації даних, публікування створеного проекту на традиційних носіях, на *CD-ROM*, на локалізованих комп'ютерах, в *intranet* та в *Internet*;

– створити необхідне програмне забезпечення та продукцію комп'ютерних інформаційних графічних технологій; опублікувати створений проект.

## ГЛАВА 12. КОМП'ЮТЕРНА ГРАФІКА У НАУКОВІЙ ТА ВИРОБНИЧІЙ СФЕРАХ

### 12.1. Функції та застосування систем наукової комп'ютерної графіки

**Визначення наукової комп'ютерної графіки.** *Під науковою комп'ютерною графікою розуміють технологію використання методів та засобів комп'ютерної графіки для наукових досліджень та експериментів у найрізноманітніших галузях науки та техніки [28, 63, 91].*

Системи наукової комп'ютерної графіки, залежно від ступеня їх універсальності, підтримують процеси розв'язування всіх або декількох задач із наведеного нижче переліку [33, 58, 63, 66, 171]:

- введення початкових даних (з клавіатури, за допомогою графічного планшету, шляхом сканування, з датчиків тощо) в форматі, потрібному для конкретного застосування чи користування;

- введення (з клавіатури, зі спеціалізованої бібліотеки тощо) та редагування закономірностей та співвідношень із провідних галузей науки та техніки в спеціальній нотації (математичних та фізичних формул, хімічних рівнянь тощо) для подальшого їх спрощення, виконання обчислень або інших типових маніпуляцій з ними;

- виконання математичних (аналітичних та чисельних) обчислень, статистичних розрахунків та обробки результатів

експериментів, інших перетворень згідно з тими закономірностями та співвідношеннями, що є стандартними для даної системи або введені з клавіатури в спеціальній нотації;

– корегування даних, перетворення форматів та типів даних, приведення даних до потрібних одиниць вимірювання;

– оптимізація, захист та збереження даних на електронних носіях інформації, ефективний розподілений доступ до даних;

– ведення підсистем з елементами штучного інтелекту та прийняття рішень (баз знань, експертних систем, систем розпізнавання образів тощо);

– графічна візуалізація даних (вхідних, результатів обчислень та інших) у вигляді 2D/3D-діаграм, графіків, схем, карт, рисунків тощо;

– формування інтерактивних науково-технічних текстів та документів, які містять формули, графічні зображення, пояснювальні тексти та інші типові елементи (створення їх з нуля або на основі зразків із колекції типових науково-технічних документів із провідних галузей науки та техніки, підготованої професіоналами);

– виведення сформованої науково-технічної документації на традиційні носії інформації, публікація отриманих результатів на *Web*-сайті та передавання їх мережею.

**Основні класи систем, спеціалізованих на науковій комп'ютерній графіці. По-перше, це універсальні системи автоматизованого проектування науково-технічних задач (*MathCAD, Mathematica, MathLAB, Maple, Statistica, SPSS, Axum, NCSS, S\_Plus, StatGraphics Plus та інші*), що поєднують обчислення та перетворення, візуалізацію та документування в єдиному середовищі та призначені для засто-**

сування до широкого кола прикладних галузей [48, 53-56, 118, 132].

*По-друге, це програми-побудовники графіків функцій (GraphIt, SuperGraph).* Сама назва говорить про основне призначення цих систем, але вони мають і додаткові можливості.

*По-третє, це системи, що спеціалізуються на створенні науково-технічних текстів та документів (вузькоспеціалізовані текстові процесори ChiWriter, epiGraph та інші; універсальні текстові процесори, зі складу офісних пакетів або автономні, такі як Microsoft Word або Лексикон-XL).*

Це так звані "редактори формул", які містять засоби формування наукової документації з використанням спеціальної нотації та повинні забезпечувати підтримку: спеціальної командної мови та стандартизованого внутрішнього подання для введення формул з клавіатури, їх редагування та зберігання; генерації зовнішнього (екранного та друкованого) подання формул зі стандартизованого внутрішнього подання; по можливості – механізм перетворення введених формул до такого запису, що дозволяє підключити систему аналітичних перетворень.

*По-четверте, це системи наукової комп'ютерної графіки, спеціалізовані на конкретних прикладних галузях, таких як:*

- дослідження та моделювання географічних та природних явищ (об'єктів, процесів, ресурсів, систем, комплексів тощо) для застосування в геології, гідрогеології, нафтогазових розвідці та видобуванні, рибному промислі, сейсмології, метеорології, екології (ці графічні системи повинні забезпечувати

створення та обробку географічних та рельєфних карт, дослідницьких карт для буріння та гірничих робіт, океанографічних карт, карт погоди та ізоліній тощо);

- медичне діагностування з динамічною графічною візуалізацією результатів на екрані комп'ютеру (томографія, ультразвукове дослідження внутрішніх органів та систем людини тощо);

- астрономічні та космічні дослідження.

**Основні класи систем, які не спеціалізуються на науковій комп'ютерній графіці, але мають певний набір функцій її підтримки.** Це офісні пакети (*Microsoft Office* та інші), мови програмування (наприклад, *Delphi* має потужну бібліотеку графічних візуальних компонентів та математичні бібліотеки *Advanced Calculus*), науково-інженерні калькулятори (програми для обчислень за формулами для наукових та технічних розрахунків) [4, 28, 91].

**Огляд класичних графічних пакетів наукової комп'ютерної графіки.** *MathCAD* – це найбільш яскравий представник САПР науково-технічних задач. Він може виконувати складні розрахунки, наочно подавати їх результати, пояснювати математичні співвідношення за допомогою тексту на базі відшліфованих функцій обробки графіки та текстів [53, 54].

*MathCAD* підтримує базові математичні операції з дійсними та комплексними числами з плаваючою комою, нечіткими множинами. Числові дані можна як завгодно форматувати, округлювати, подавати в показниковій формі. Підтримана обробка іменованих чисел разом із одиницями вимірювання, конвертація результатів обчислень у потрібну систему мір.

*MathCAD* надає всі можливості класичного редактору формул. Підвантажувані панелі інструментів допомагають у виборі модулів формул, а піктограми резервування звертають увагу на недостатні дані. Надається автоматична обробка вводу (опускання непотрібних дужок тощо).

У складі *MathCAD* є експертна система *Smartmath*, що містить велику кількість математичних правил та застосовує їх для символьного спрощення формул, значно прискорюючи обчислення.

Стандартний набір функцій *MathCAD* містить численні інструкції для числових та символьних розрахунків із усіх галузей математики та статистики, наприклад: різноманітні перетворення; розв'язування невизначених інтегралів та складних диференціальних рівнянь; прогнозування часових послідовностей; нелінійне сполучення кривих; розширені функції кореляційного та регресійного аналізу; операції для обробки матриць (великі матриці розміром до  $8 \times 10^6$  елементів розміщують у вікнах із смугами прокручування так, що можна швидко керувати їх окремими елементами) тощо. Можна складати власні функції: *MathCAD* оброблює *DLL (Data Link Libraries)* користувача.

*MathCAD* має набір функцій обробки текстів на рівні текстових процесорів (перевірка правопису, розвинені команди пошуку та переходу для обробки великих документів, необхідний набір шрифтів та можливостей виділення тощо).

Набір функцій обробки графіки дозволяє будувати графіки функцій (стандартних або введених користувачем), одним натиском клавіші перетворювати функції та таблиці значень в *2D*- та *3D*-діаграми (можна змінювати їх розташування та роз-

міри, робити виділення кольором та штрихуванням, використовувати лінійні та точкові символи, вставляти легенди, обирати ділення чи розриви осей, висвітлювати допоміжні лінії тощо).

Система *MathCAD* найбільш підійде таким класам користувачів: тим, хто вимушений спиратися на інтуїтивне використання математичного забезпечення (цьому сприяє велика кількість готових *MathCAD*-документів, розроблених професіоналами для типових задач із різних спеціальних галузей); тим, кому потрібно часто комбінувати формули, діаграми, розрахунки та тексти в єдиному документі; тим, хто хотів би складати презентації, пов'язані з науково-технічною діяльністю.

Подібно іншим класичним програмам із галузей математики та статистики, *MathCAD* постійно розвиває можливості та поновлює версії для платформ *DOS*, *Windows*, *NT*, *XP*.

**Система *Mathematica*** поєднує можливості аналітичних та чисельних обчислень, візуалізації та документування в єдиному середовищі, дозволяючи працювати ефективно з моменту виникнення ідеї до отримання кінцевого результату [55].

Візуалізацію система *Mathematica* виконує приблизно на одному рівні з системою *MathCAD*, дозволяючи будувати та трансформувати *2D*- та *3D*-графіки, діаграми.

Окрім графічних робіт, ядро системи виконує всі цифрові та символічні розрахункові задачі. Можна виконувати арифметичні операції з довільною точністю, дії над комплексними числами, матричні перетворення, перетворення Фур'є, символічне інтегрування, спрощення систем рівнянь, розв'язування їх та диференціювання в символічному вигляді тощо. Можна



здійснювати вибір серед великої кількості функцій із комплекту стандартного постачання системи: пакети функцій з галузей алгебри, комбінаторики, математичної статистики, геометрії, лінійної алгебри та теорії чисел вводять до пам'яті під керуванням ядра системи. Можна розширювати бібліотеку функцій за допомогою спеціальної інтегрованої мови програмування.

Основною властивістю системи *Mathematica* є прогресивна технологія *Wolfram Research "packed array technology"*, що комбінує символну алгебру з швидкими та ефективними чисельними розрахунками: можна інтерполювати масиви з більшою у 2,0-2,5 рази кількістю точок, використовуючи той самий обсяг пам'яті, та працювати з великим обсягом даних значно швидше (більше, ніж у десять разів), у порівнянні з роботою без застосування цієї технології.

Для роботи з документами та їх публікації в різноманітних форматах: користувачу надається зручний та елегантний інтерфейс для запису результатів обчислень, підготування професійних звітів та статей, презентацій та технічного листування; постійно розширюються можливості виведення в форматах *HTML* та *TeX* (можна створювати *Web*-сторінки та графіки за допомогою однієї команди, зберігати формули та цілі документи в форматі *TeX* тощо); система має перевірку правопису, до її комплекту постачання включено спеціалізовані математичні та наукові словники, вона здійснює автоматичне перенесення в тексті та математичних виразах.

*Mathematica* є потужним та зручним засобом експериментального аналізу даних, обробки зображень та математичного моделювання завдяки широким можливостям імпорту та

експорту числових даних, графіки та звуку (більше двох десятків найвідоміших форматів). Математичні функції та опції для аналізу даних дозволяють працювати з великими та малими обсягами даних гнучко та швидко.

*Mathematica* має мережний менеджер ліцензій для всіх платформ. Міжплатформова мережна архітектура цього пакету робить можливими різні схеми ліцензування: розташування ядра на сервері або на локальному комп'ютері, локальне або централізоване розташування виконуваних програм; можливість роботи з пакетом в одній мережі машин, керованих різними операційними системами.

Виходячи в *Internet*, можна звертатися до готових рішень, що підготовані іншими користувачами системи *Mathematica*.

Пакет *Mathematica* є найкращим вибором: для тих, кому потрібні потужні математичні можливості (цей пакет надає до розпорядження максимум функцій, яких можна очікувати від математичної програми); для тих, хто не кожного дня користується математичними пакетами або бажає вивчити та випробувати нові функції (завдяки розвиненим засобам довідкової системи з незліченими прикладами застосування, зручним панелям інструментів та довідкової системи); для тих, кому доводиться виконувати роботи з дужками різних рівнів та форм (завдяки тому, що ці роботи суттєво полегшуються доданими палітрами).

Головною відмінністю *системи SPSS* є поділ на базовий пакет та додаткові модулі, що можна придбати окремо.

Модуль *Professional Statistics* надає процедури кластерного та факторного аналізу, багатовимірного масштабування. Модуль *Advanced Statistics* здійснює регресію, аналіз, дисперсійний аналіз та аналізи, пов'язані з імовірністю безвідмовної

роботи на протязі заданого часу. Модуль *Trends* містить способи сполучення, екстраполяції та згладжування кривих, регресійну модель. Модуль *Tables* надає різноманітні можливості форматування для перетворення числового матеріалу в таблиці, підготовані до друку. Модуль *Missing Value* керує підбиранням значень, враховуючи побажання користувача. Модуль *Exact Tests* відповідає за аналіз малих блоків даних. Модулі *Categories*, *Chaid*, *Lisrel* містять додаткові функції.

Базова програма містить зручний редактор даних. Набір функцій перетворення даних дозволяє привести невпорядковані початкові дані (за допомогою сортування, групування, об'єднання, перерахунку) до форми, що є придатною для подальшого аналізу даних. Нові блоки даних можна створювати автоматично з наявних блоків. Бажану формульну залежність можна створювати дуже просто – за допомогою натисків кнопки "миші", зі списків початкових змінних, команд переходу та численних математичних функцій.

Базовий пакет *SPSS* забезпечує такі графічні можливості: побудову описувальних графіків, деяких діаграм для отримання якісної оцінки процесів, різноманітних часових діаграм; спеціальний редактор для індивідуальних оформлювальних потреб (графіки можна споряджати рамками, символами, штрихуванням, текстовими полями з різними кольорами та шрифтами; 2D-діаграми можна обертати; 3D-графіки можна переводити в бажане просторове положення).

Потенційні класи користувачів пакету *SPSS* обумовлені такими його особливостями: базовий пакет *SPSS* надає достатньо повний асортимент статистичних та графічних функцій, що є легкими та зручними в використанні навіть для новачків; завдяки можливості помодульного придбання *SPSS*, потенціал

цієї системи можна гнучко нарощувати та адаптувати до потреб конкретного користувача, але з іншого боку, в порівнянні з повними комплектами, можуть виникнути значні додаткові витрати; регресійний аналіз даних може бути автоматизовано досвідченим користувачем із застосуванням синтаксису команд; постійно зростає кількість додаткових модулів, які можна вставляти до програми [33, 66].

*Paquem Statistica* має модульну структуру, завдяки цьому оптимально використовуючи системні ресурси: оскільки в кожному окремому випадку використовують обмежене коло функцій, то є можливість вводити до оперативної пам'яті тільки потрібні на поточний час модулі [28, 33, 66, 91].

Базовий модуль відповідає за описувальну статистику, підготування кореляційних матриць, проведення простих тестів та розрахунок таблиць відносних імовірностей частот подій та статистичних розподілень.

Інші модулі підтримують: широко розповсюджені на практиці непараметричні методи випробувань (якщо дані не підпадають під класичний нормальний розподіл); однофакторний та багатфакторний дисперсійний аналіз, лінійну регресію (до 500 змінних у моделі), нелінійну регресію, факторний аналіз, багатовимірне масштабування, аналіз та ймовірносне моделювання; у межах аналізу часових послідовностей – інструменти для трансформації та згладжування, автокореляції та перехресної кореляції, розділення систематичних та випадкових флуктуацій; тематичні галузі "планування експериментальних досліджень", "аналіз процесів", "забезпечення якості", що обслуговуються підпрограмами; командну мову *SCL*, що обслуговується підпрограмами; управління даними (введення

та виведення зовнішніх форматів даних, змішування та розділення блоків даних); для прискорення стандартизованих аналізів у режимі підпрограм – вільно програмовані макроси (можуть з часом вдосконалюватися та тим самим оптимізуватися), нарівні зі звичайними функціями програмування – спеціальні статистичні та математичні функції, інтеграцію зовнішніх файлів *DLL*; візуалізацію даних.

Графічний модуль має потужні можливості для презентації та аналізу даних: надає більше 100 типів діаграм, які однаково широко використовуються в науці та в промисловості; завдяки тісному зв'язку між статистичною та графічною частинами, будь-який результат аналізу буде підтримано відповідними діаграмами одним натиском кнопки "миші"; індивідуальне оформлення діаграм повністю може виконати користувач, хоча для швидкого огляду вистачає стандартного оформлення; тривимірні зображення є повноцінними, їх можна плавно обертати навколо просторових осей, регулюється ступінь їх перспективного викривлення; завдяки спеціальній функції (*Brushing*), можлива одночасна вибірка точок даних із декількох графіків або розгорнутих таблиць, а за допомогою режиму "анімованого *Brushing*" можна наочно подавати взаємозалежності в межах блоків даних.

Усі варіанти постачання пакету *Statistica* надають ідентичні функції управління даними, загальної статистичної обробки та графічної візуалізації. На відміну від цього, спеціальний інструментарій має такі розбіжності: основна версія відмовляється від методів вищої статистики на користь промислових застосувань та моделей на основі структурних рівнянь.

Основні класи користувачів системи *Statistica* визначені такими факторами: *Statistica* є давно відомою на ринку статистичного ПЗ, комплектною статистичною системою, що використовує уні-, бі- та мульти-методи; інтегрований графічний модуль надає великий вибір 2D- та 3D-діаграм; макроси та інтегрована командна мова автоматизують процес аналізу.

**Застосування наукової комп'ютерної графіки в геології, гідрогеології, нафтогазових розвідці та видобуванні, сейсмології, екології, метеорології.** Дослідникам, які працюють у цих галузях науки та техніки, добре відомі пакети наукової та інженерної графіки фірми *Golden Software*. Серед її продукції особливо цікавою є система оцифрування даних *Didger* – програмний продукт, призначений для перетворення віддрукованих на папері карт, графічних зображень, аерофотознімків та іншої графічної інформації в цифровий формат [28, 33, 171].

*Didger* – не єдина система оцифрування зображень, але є особливо цікавою завдяки інтегрованості з іншими пакетами фірми *Golden Software* та націленості на розв'язування задач, що належать до повністю визначеної та широкої науково-технічної галузі.

Особливо зручно те, що *Didger* дозволяє отримувати безпосередньо з програми проектні дані для побудови карти основи (*Base Map*) та карти початкових даних (*Post Map*) у пакеті *Surfer*, а також графіків у програмі *Grapher*.

У першу чергу *Didger* призначений для роботи з паперовим носієм та планшетом для оцифрування (*digitizing tablet*), який називають також дигитайзером або координатографом. Як правило він надає внутрішню координатну систему з високим дозволом, поверх якої поміщають карту або графічне зобра-

ження. *Didger* може працювати з усіма моделями планшетів, що підключаються до ПК та споряджені драйверами для *Windows/NT/XP*.

*Didger* можна використовувати на ПК зі стандартним набором периферійних пристроїв, виконуючи оцифрування карт на екрані монітору: координати знімають з поточної позиції курсору, що пересувають за допомогою "миші" або клавіатури. В такому випадку необхідно мати готове електронне зображення карти, а точність введення буде обмежена дозволом екрану.

*Didger* працює під управлінням *Windows/NT/XP*, висуваючи обмежені вимоги до ресурсів ПК (мінімально – 8 Mb оперативної пам'яті та 5 Mb на жорсткому диску).

## Питання для самоперевірки

1. Сформулюйте визначення наукової комп'ютерної графіки.
2. Які основні задачі розв'язують системи наукової комп'ютерної графіки?
3. Наведіть класифікацію систем, які спеціалізовані на науковій комп'ютерній графіці або мають певний набір функцій для її підтримки, та охарактеризуйте їх практичне призначення.
4. Сформулюйте особливості практичного використання класичних пакетів наукової комп'ютерної графіки *MathCAD*, *Mathematica*, *SPSS*, *Statistica*.
5. Яку роль відіграють системи оцифрування даних?

## 12.2. Основні поняття про інженерну комп'ютерну графіку

**Системи автоматизованого проектування (САПР) як головне практичне втілення інженерної комп'ютерної графіки.** Інженерна комп'ютерна графіка призначена для автоматизації креслярсько-графічних, конструкторських та дизайнерських робіт у процесі проектування та наочної (реалістичної) динамічної графічної візуалізації компонентів та систем механічних, електричних, електромеханічних, електронних, радіотехнічних, радіоелектронних, оптичних конструкцій, схем, мереж, пристроїв та приладів.

Основними об'єктами інженерної комп'ютерної графіки складні структури: інтегральні схеми; промислове та побутове електричне та електронне обладнання; хімічні, енергетичні та космічні установки; кузови автомобілів, фюзеляжі літаків, корпуси суден; будівлі та споруди (промислові, сільськогосподарські, житлові, суспільні, військові); комунікації та мережі (газові, водопровідні, каналізаційні, електричні, оптичні, телефонні, телевізійні, комп'ютерні); пожежно-охоронні системи; обладнання інтелектуальних споруд; системи впорядкування адміністративно-територіальних одиниць (сіл, міст, регіонів) тощо.

На стику інженерної та наукової комп'ютерної графіки знаходяться питання дослідження та моделювання природних явищ, об'єктів, процесів та систем, задачі картографії.

Системи інженерної комп'ютерної графіки дозволяють виконувати в реальному часі 2D-моделювання, каркасне, по-



верхнєве та твердотїльне 3D-моделювання (включаючи отримання проєкцій та перерїзів), морфїнг, аїмацію, реалїстичну вїзуалїзацію (ефект перспективи, накладання текстур, їмїтацію природних та штучних джерел свїтла, атмосфернї ефекти, моделювання рїзноманїтних властивостей проєктованих об'єктів, явищ, процесів, систем та комплексів тощо).

Системи їнженерної комп'ютерної графїки та дизайну передають комп'ютеру бїльшу частину рутинної роботи з проєктування та вивільняють час їнженера-конструктора для творчої дїяльності, одночасно суттєво пїдвищуючи якїсть результатів та скорочуючи строки проєктування.

Першочерговими користувачами систем їнженерної комп'ютерної графїки є машино-, приладо-, судно- та авїабудївнї пїдприємства, архїтектурно-будївельнї фїрми.

В їнженернїй комп'ютернїй графїцї загальноприйнятими є скороченї термїни *CAD*, *CADD*, *CAM*, *CAE*, *PDM*, *AEC*, *GIS*.

*CAD* – система конструювання (*Computer Aided Design*), *CADD* – система технїчного креслення (*Computer Aided Drafting and Drawing*), але найчастїше цї поняття поєднують пїд однїєю абрєвіатурою *CAD*. *CAM* (*Computer Aided Manufacturing*) – виробництво за допомогою комп'ютерів. *CAE* (*Computer Aided Engineering*) – аналітично-розрахункова пїдсистема. *PDM* (*Project Document Manufacturing*) – система управлїння документо-обїгом та даними проєкту. *AEC* (*Architectural Engineering Computing*) – це скорочене позначення для архїтектурно-будївельних комп'ютерних систем. *GIS* (*Geo Information System*) – геоїнформацїйна система.

Прикладами найвїдомїших систем їнженерної графїки є:

– *AutoCAD* – потужний пакет для автоматизації розробки та виконання проектно-конструкторських та графічних документів, можливості якого постійно покращуються згідно положень Всесвітньої групи користувачів *AutoCAD*;

– *Solid Pipe Designer* – проектувальник вузлів та механізмів, а також інструментальних засобів для проекту конвеєрного пересилання процесу, що забезпечує користувача графічно масштабованим поданням виробу;

– *bCAD* – повністю об'єднане функціональне середовище для 2D/3D-моделювання та реалістичного подання фотографій;

– *ArchiCAD* – лідер серед архітектурних CAD-пакетів та професійний інструмент для архітектурних фірм, який є корисним для вдосконалення процесу проектування, розробки фірмового дизайну та реклами, створення презентацій, виконання конкурсного проектування (поєднує можливості 3D-моделювання та об'єктно-орієнтованого підходу, маючи такі переваги, як інтеграція комп'ютерного 3D-будівництва з потужними засобами візуалізації, одночасний доступ до всіх даних, здатність використання інтелектуальних будівельних матеріалів тощо);

– *ArcView* – система, що дозволяє переглядати та друкувати бази даних, виконані в *ArchiCAD*;

– *3D Home Architect Deluxe* – незамінний помічник архітектора;

– *AutoDesk Mapguide Author* – система, що дозволяє професійно створювати діалогові карти;

– *Engineering Geometry Assistant* – це система, що вирішує сьогоденні геометричні проблеми, обробляє координатну геометрію, здійснює заключні технічні рисунки, виконує для

основного рисунку розробки та розміщення згідно з тим, що електронна таблиця виконувала для обчислень проекту;

– *Femap* – система, за допомогою якої користувачі можуть визначати моделі аналізу, об'єднувати відповідні технології вирішуючого пристрою та огляд, інтерпретувати та документувати їх результати швидко та ефективно;

– *VariCAD* – потужний інструмент для "розуміння" ідей користувача, до якого можна швидко пристосуватися завдяки високій діалоговості та дружності, наявності різних методів та підходів для задоволення індивідуального стилю та творчого потенціалу.

Типовими прикладами систем інженерної комп'ютерної графіки також є: *ProEngineer*, *KOMIAC*, *ArCon*, *P-CAD*, *OrCAD*, *MicroCap*, *Electronics Workbench* та багато інших.

Найважливішим та найголовнішим втіленням уявлень про інженерну комп'ютерну графіку та дизайн є системи автоматизованого проектування (САПР), розвиток яких здійснювався в три етапи: дослідження можливостей автоматизації проектування (1970-і роки); поява масових САПР та базових програмних продуктів для них такого рівня, що використання їх у промисловості стало економічно виправданим (1980-і роки); напрацьовування практичних застосувань, виправлення помилок, досягнення високої ефективності *CAD*-систем у високотехнологічних виробництвах (1990-і роки). Перспективним напрямком подальшого підвищення продуктивності систем інженерної комп'ютерної графіки є забезпечення їх функціонування в складі інтелектуальних САПР у мережі *Internet*.

***Розглянемо переваги застосування інженерних САПР та їх роль у галузі матеріального виробництва. До появи***

ЕОМ випуск конкурентноздатної продукції здійснювався завдяки науковій організації процесу проектування та першокласним фахівцям. Застосування САПР дало безперечні переваги: дозволило впровадити стратегію накопичування та передавання знань та досвіду конструкторів, що забезпечує високу живучість; сприяє розв'язуванню тих практичних задач виробника, що дозволяють проводити прогресивну політику на ринку (знизити виробничі витрати та собівартість, скоротити час розробки та підвищити споживчі якості продукції).

Головною перевагою використання САПР є висока швидкість виходу продукції на ринок. Тут спрацьовує відома в економіці залежність між обсягом продажу товару та часом: виробник, який отримав через пізній вихід продукції на ринок менший прибуток, направить на розробку нового покоління продукції менше засобів, і так – до повної зупинки виробництва. Це підтверджує дослідження компанією *Hewlett-Packard* можливих втрат прибутку виробниками в конкурентній боротьбі: 33 % втрат – через піврічне спізнення виходу продукції на ринок; 22 % втрат – через підвищення матеріалоємності продукції на 9 %; лише 3,5 % втрат – через підвищення витрат на проектування на 50 %.

САПР є індустріальними технологіями, що націлені на найважливіші галузі виробництва, визначаючи рівень розвитку та стратегічний потенціал нації. Без САПР неможливо виробляти сучасну техніку, що є дуже складною та вимагає надзвичайної точності виготовлення. Рівень розвитку САПР, кількість робочих місць САПР та кількість інженерів, які професійно володіють *CAD*-технологіями, впливають на статок кожного члена суспільства.

Розглянемо відмінності *CAD*-програм від інших класів графічних програм.

Професійні графічні програми надають можливість працювати з графічними об'єктами: ПЗ зберігає лише вид об'єкту та значення його властивостей. *CAD*-програми працюють за тією ж схемою: технічне креслення складається з невеликої кількості основних геометричних об'єктів, розміри яких зберігаються в файлі креслення. Основні розбіжності між *CAD*-програмами та іншими графічними програмами полягають в їх практичному застосуванні.

У звичайних графічних програмах: головну роль відіграють художнє зображення, графік або ескіз, які потрібно гарно зафарбувати; зображення складається з зафарбованих поверхонь та нарисованих від руки ліній; основною метою є гарне графічне оформлення документу; всі об'єкти розташовують у системі координат користувача.

У *CAD*-програмах: на першому місці знаходиться виробнича продукція (конструкція виробу, проект будівлі або агрегату); зображення складається з прямих ліній, кіл, дуг та ламаних ліній, які можна конструювати та змінювати багатьма способами; методи конструювання для базових геометричних об'єктів, стискання та розтягування довільних об'єктів, заокруглювання кутів належать до стандартного набору функцій *CAD*-програм; у цих програмах завжди працюють у масштабі та з точними координатами.

Загальним є те, що будь-яка графічна програма в ідеалі повинна надавати можливість нанесення розмірів та штрихування.

У процесі автоматизації креслярських та конструкторських робіт головну увагу приділяють: випуску точних креслень деталей, вузлів та зборок; інтерактивній роботі з розміщеною в ЕОМ моделлю проєктованого компонента чи системи для перевірки її механічних, електричних, теплових та інших властивостей. Як правило модель інтерпретують за допомогою моделювальної програми, що видає інформацію про поведінку системи оператору за дисплейним пультом для виконання чергових циклів проєктування та перевірки. По завершенні проєктування об'єкту допоміжні програми можуть провести постобробку проєктної БД для підготування переліків деталей та зведеної відомості необхідних матеріалів, сформувані дані для оброблювальних центрів.

Модель конструювання в 2D-просторі забезпечує конструктору необхідний сервіс для побудови контурів із використанням традиційної креслярської техніки. Конструктор може працювати в одній, двох або трьох довільно обраних проєкціях, використовуючи мову, максимально наближену до лексики тієї предметної галузі, що досліджується. Операції, реалізовані за допомогою цієї мови, приводять до побудови в пам'яті ЕОМ 2D-образу та 3D-моделі конструйованого об'єкту. Отриману 3D-модель зображують у вигляді каркасної (проволочної) моделі, поверхневої моделі чи моделі твердого тіла. В останньому випадку часто здійснюють двоетапний процес проєктування: на першому етапі конструктор оперує об'ємними примітивами; на другому етапі об'єкт, відображений за допомогою 3D-примітивів, переводять у нормальну напівтонову форму. Модуль параметричного конструювання дозволяє використовувати техніку задавання параметрів проєкто-

ваного виробу, формувати каталоги уніфікованих стандартних та нормованих деталей.

**Загальна характеристика прикладного математичного апарату інженерної комп'ютерної графіки та дизайну.** Навіть у багатотомному виданні неможливо подати весь концептуальний та математичний апарат, який становить собою теоретичний фундамент сучасної комп'ютерної графіки та такого її ключового напрямку, як інженерна комп'ютерна графіка та дизайн: викладенню цього формального апарату присвячені численні фундаментальні науково-технічні видання (книжки, статті в журналах, матеріали конференцій та семінарів). Тим більше неможливо вивчити та опанувати на практиці весь цей теоретичний матеріал у рамках курсу навчання в вищому навчальному закладі.

Розуміючи це, автори цієї книги поставили собі за мету сконцентрувати в ній лише ті математичні аспекти побудови реалістичних зображень за допомогою EOM, які, становлячи собою невід'ємний складовий елемент комп'ютерної графічної освіченості сучасного інженера, створюють теоретичну базу для подальшого самостійного вивчення математичного апарату комп'ютерної графіки та дизайну за такими навчальними посібниками, як [10-13, 33, 37, 39, 41, 43, 46, 57, 60, 61, 64, 79, 98, 109, 111, 119, 120, 124-128, 140-142, 154, 157, 158, 166, 173, 174] та іншою фундаментальною літературою.

До тієї частини вказаної теоретичної бази, що було викладено в попередніх главах, додатково зауважимо таке.

Ключовими напрямками математичного моделювання в інженерній комп'ютерній графіці є:

–  $2D$ -модельовання, тобто створення та редагування плоских креслень – двовимірних проєкцій просторових об'єктів;

–  $3D$ -модельовання, тобто створення, інженерний аналіз та редагування динамічних просторових геометричних моделей, що належать до трьох базових класів моделей – каркасних (полігональних), поверхневих та твердотільних, а також створення їх наочних та реалістичних графічних зображень, з можливістю автоматичного отримання потрібних проєкцій та перерізів утворених  $3D$ -об'єктів.

Ці ключові напрямки математичного модельовання в САПР базуються на таких математичних теоріях: аналітична та диференційна геометрія, векторна алгебра, нарисна геометрія та креслення, теорія графів, чисельні методи розв'язування математичних задач, математична логіка, методи оптимізації, теорія інформації, теорії прийняття рішень, штучного інтелекту та розпізнавання образів. У цьому зв'язку далі відзначимо таке.

Переважна більшість задач комп'ютерної (обчислювальної) геометрії та графіки полягає в необхідності створення аналітичного опису та подання в ЕОМ відрізків ламаних і кривих ліній та ділянок багатограних і криволінійних поверхонь у двовимірному, тривимірному або багатовимірному просторах, синтезу цих ліній та поверхонь, формоутворення згідно з заданими умовами, визначеними точно або за допомогою експертних оцінок (наприклад, таких, які враховують естетичні вимоги).

Традиційним математичним апаратом комп'ютерної геометрії та графіки є векторна алгебра, на базі якої розв'язують задачі на перетворення систем координат, афінні перетворення на площині та в просторі. При цьому отримання (ідентифікація)



координат точок найчастіше здійснюється на основі математичного апарату аналітичної геометрії.

Оскільки більшість об'єктів у техніці обмежені складними кривими поверхнями, для полегшення розрахунку, обробки та втілення в реальній практиці цих поверхонь, їх розбивають на так звані кінцеві елементи (КЕ) – невеликі чотирикутні або трикутні відсіки, з подальшим їх параметричним поданням (це дозволяє не прив'язувати КЕ до системи координат).

Загальна теорія розбиття проектованої поверхні на КЕ була запропонована С.А.Кунсом: основою цієї теорії є метод з'єднання чотирьох довільних граничних кривих єдиним гладким відсіком поверхні, що забезпечує неперервність градієнтів та кривизен між суміжними відсіками цієї поверхні.

Широко застосовується в інженерній комп'ютерній графіці метод Безье, що базується на многочленній апроксимації Бернштейна: згідно з цим методом, проектувальник задає незамкнену ламану або багатогранну поверхню, що апроксимується відповідною гладкою кривою або поверхнею.

Неможливо не відзначити величезну роль для математичного моделювання в САПР теорії сплайнів, методи якої дозволяють подавати будь-яку сплайнову криву або поверхню за допомогою фундаментальних сплайнів (*B*-сплайнів) таким чином, що можна здійснювати локальну модифікацію поверхні, не піддаючи перерахунку цю поверхню в цілому.

Також потрібно відзначити, що стрижнем математичного моделювання в сучасних САПР є метод геометричного векторно-параметричного моделювання.

Найтиповішими для подання графічної інформації в САПР є спискова структура та складні ієрархічні структури даних.

Найбільші труднощі в процесі геометричного та графічного моделювання в САПР виникають при розв'язуванні на ЕОМ метричних та позиційних задач, коли потрібно оцінювати взаємне положення геометричних фігур, що призводить до необхідності отримання все більш ефективних та швидкодіючих алгоритмів, які виключають перебір усіх можливих варіантів та здійснюють направлений пошук.

Важливим фактором при розв'язуванні геометричних задач на ЕОМ є стійкість застосованих алгоритмів.

**Особливості апаратно-програмної платформи інженерних САПР.** САПР – це та галузь, в якій комп'ютерні технології відіграють безпосередню роль в створенні товару та грошей. Тому в процесі проектування САПР некритичних або незначних моментів нема.

Розумний вибір апаратно-програмної платформи САПР, яка б гарантувала тривалий строк життя системи в цілому, є непростою задачею. Висока динаміка ринку апаратних засобів, зміщення пріоритетів у системному ПЗ та технологіях розробки прикладного ПЗ роблять цю задачу багатоваріантною. За цих умов вибір позиції переслідування сьогохвилинних переваг може призвести до збільшення витрат на етапі розгортання системи та до їх зростання в процесі експлуатації.

Про апаратне забезпечення САПР можна почути протилежні думки: максималістську, коли одноразово закуповується все найкраще та найдорожче; супереконномну, коли знижують вартість системи за рахунок низькоякісної апаратури, що не відповідає вимогам до продуктивності та надійності. Найчас-

тіше системи, створені на основі цих протилежних концепцій, однаково неефективні.

До особливостей засобів апаратно-програмної підтримки САПР належать: підвищені вимоги до пропускну здатності інтерфейсів, пов'язаних з передаванням даних (комп'ютерних шин, каналів, мереж тощо); те, що показники продуктивності на операціях із плаваючою точкою повинні бути незрівнянно вищими за показники офісно-побутової обчислювальної техніки.

Для визначення мінімальних припустимих значень смуги пропускання (в *MBps*) та продуктивності (в мільйонах операцій з плаваючою точкою – *MFLOPS*) необхідно проаналізувати основні специфічні для САПР програмні компоненти, в першу чергу – бібліотеку *OpenGL*, яку використовують у переважній більшості САПР та яка є фактичним стандартом.

У загальних рисах параметри, що передаються викликам *OpenGL*, можна охарактеризувати обсягом даних (кількістю байтів), які необхідні для моделювання таких абстрактних об'єктів: вершина 3D-моделі, колір та нормаль до елементу поверхні (3-24 байти), флаг належності до грані (1 байт), властивості матеріалу (8-172 байти), додаткові відомості про текстуру та інші (6-48 байтів).

Для подання простої геометричної моделі, що містить 50000 вершин, знадобиться від 2 до 20 *Mb* службової *OpenGL*-інформації. Для ретельного виведення цієї моделі на екран та здійснення з нею операцій в реальному часі, тобто для забезпечення частоти поновлення екрану 24 кадри/с, необхідна пропускна здатність приблизно 500 *MBps*. Якщо ввести мультиплікаційний поправочний коефіцієнт на складність моделі, визначений класом розв'язуваних задач, можна стверджувати

про таку мінімально необхідну пропускну здатність: для простих задач – 700 MBps; для комплексних задач – 5-7 GBps.

Щодо показників продуктивності на операціях із плаваючою точкою, то в розрахунку на ефективну роботу одного користувача мінімально необхідна продуктивність складе 150-200 MFLOPS.

Ці оцінки, зроблені для даних як динамічного компоненту САПР, визначають вимоги до обчислювальної частини апаратної платформи. Оцінка статичного аспекту даних САПР – розмірів робочих файлів – дозволяє уникнути помилок у процесі вибору методів та засобів зберігання, накопичування та обміну даними. Це важливо тому, що кількість файлів діючої САПР може становити від десятків тисяч до сотен мільйонів, а співвідношення їх розмірів – 100 млн. та вище.

**Стан ринку інженерних САПР.** Цей ринок пропонує великий асортимент ПЗ для автоматизації інженерної діяльності в проектних організаціях та на виробництві. Він має неабиякий попит серед вітчизняних промислових підприємств, які розглядають системи для комплексного автоматизованого управління виробництвом, автоматизації розробки та підготування виробництва нових виробів як ефективний засіб виходу на внутрішній та зовнішній ринок із конкурентоздатною продукцією світових стандартів.

Найактивніший пошук можливостей та засобів розв'язання цих задач здійснюють ті підприємства, що випускають засоби виробництва та товари народного споживання (машинобудівні, станкобудівні, приладобудівні підприємства та інші).

Повна автоматизація виробництва пов'язана з великими капітальними вкладеннями. Велика вартість багатфунк-

ціональних САПР високого рівня та обмежені можливості розробки складних виробів засобами 2D-САПР легкого рівня привели до розквіту САПР середнього рівня, розвиток яких підтримують всі розробники CAD/CAM-систем, та до популярності серед користувачів CAD/CAM/CAE-систем пірамідальної схеми автоматизації конструкторсько-технологічного підготування виробництва. Вона значно економить ресурси завдяки використанню на робочих місцях ПЗ виключно з необхідним набором функціональних можливостей: для концептуальної конструкторсько-технологічної проробки нового виробу – декілька робочих місць з САПР високого рівня; для детальної проробки виробу – декілька десятків або сотень робочих місць з САПР середнього рівня; для підготування конструкторсько-технологічної документації та робочих креслень – до тисячі робочих місць з САПР легкого рівня.

#### **Характерні особливості розробки інженерних САПР.**

Розробникам САПР доводиться враховувати такі особливості об'єктів автоматизації (підприємств та організацій): прийнято ділити об'єкти автоматизації на підрозділи, в кожному з яких розв'язують блок однотипових задач, а отримані результати є вхідними даними для іншого підрозділу (прикладом підрозділів є конструкторський відділ із дільницями розробки нових виробів, проектування пресформ і штампової оснастки та іншими, технологічний відділ із заготовчими, механо-оброблювальними, збиральними та іншими дільницями); робота базується на досвіді, що накопичувався роками та може не піддаватися узагальненню через наявність специфічних умов; необхідно враховувати всі особливості організаційної струк-

тури конкретного підприємства або організації в цілому та специфіки діяльності кожного відділу.

Для враховування цих особливостей розробники САПР використовують:

– модульний принцип організації великих та середніх САПР, згідно з яким окремі модулі повністю або частково підтримують типові задачі типових виробничих підрозділів (наприклад, САПР підприємства, що виконує листову штамповку, повинна містити модулі проектування різних видів штампів – розділювальних, витягувальних, згинальних тощо; САПР підприємства з випуску пластмасових виробів повинна містити модулі проектування прес-форм та аналізу лиття);

– створення додатків, відкритих для інтеграції з іншими програмами, та введення додаткових інструментальних засобів для адаптації та розвитку на їх основі існуючих програмних модулів (так намагаються врахувати ту специфіку виробництва, що не зводиться до розбиття задачі на множину підзадач);

– активний розвиток інструментальних об'єктно-орієнтованих середовищ та високорівневих спеціалізованих об'єктно-орієнтованих бібліотек, призначених для швидкої розробки додатків у галузі САПР без залучення кваліфікованих програмістів (цьому сприяє інтенсивний розвиток візуальних *ОО*-засобів програмування): ці додатки користувачів зможуть ефективно працювати з іншими аналогічними продуктами компанії.

**Інженерна комп'ютерна графіка, САПР та технологія віртуальної реальності.** Перспективним напрямком комп'ютерної графіки є віртуальна реальність – технологія, що забезпечує високо реалістичне моделювання 3D-простору та підтримує динамічну інтерактивну взаємодію з користувачем,

створюючи ефект природного занурення в той кіберпростір, який моделюють. Вона являє собою новий тип інтерфейсу, що якісно змінює способи взаємодії людини з комп'ютером.

Розвиток засобів віртуальної реальності почався з досліджень, що полягали в створенні авіаційних тренажерів. Далі цю технологію стали широко застосовувати для імітації функціонування літальних апаратів, космічних кораблів, автомобілів та інших складних систем. Зараз системи віртуальної реальності використовують ще ширше: в тренажерах, навчанні, комп'ютерних іграх, кіно, анімації, САПР, медицині, охороні здоров'я, маркетингу тощо (цьому сприяли розвиток апаратних засобів та їх здешевлення).

Розглянемо, як технологія віртуальної реальності сприяє підвищенню ефективності САПР, дослідивши можливості та вузькі місця застосування типового інтегрованого CAD/CAM/CAE-комплексу (системи *UG-EDS Unigraphics Fly-Through*). Цей комплекс забезпечує автоматизовану підтримку всіх етапів розробки складних виробів та конструкцій (проектування, інженерного аналізу, підготування до виробництва та інших): кожному етапу відповідає набір функціональних модулів, об'єднаних загальним інтерфейсом та базою даних, яка зберігає повний опис проєктованого виробу – головну модель (майстер-модель). Підтримане паралельне проєктування виробів (група фахівців різних профілів одночасно виконує всі етапи розробки деталей, вузлів та зборок), що сприяє скороченню строків робіт. Забезпечені актуальність головної моделі та паралельний доступ до бази даних у розподіленому середовищі проєктування.

САПР цього класу дозволяють виконувати деталізації та уточнювання, необхідні для успішного просування між етапами автоматизованого проектування, з самого початку роботи з системою. Але не виключено, що вже на ранніх етапах розробки будуть припущені помилки та недоліки, що стануть явними тільки при тестуванні зразків. Тобто вузьким місцем застосування САПР є етап виготовлення та тестування фізичних дослідних зразків та макетів, який хронологічно завершує попередні етапи проектування та підготування виробництва: якщо на цьому етапі будуть виявлені недоліки, для виправлення яких доведеться повертатися до початку робіт, це загальмує випуск виробів.

Ефективним засобом боротьби з цим недоліком є застосування систем віртуального макетування, програмне забезпечення яких базується на технологіях віртуальної реальності.

Віртуальний макет-прототип – це інтегроване цифрове подання виробу та його властивостей, що формується за даними головної моделі, відбиває просторову взаємодію компонентів та дозволяє оцінити працездатність конструкції в цілому.

Переваги САПР, які підтримують можливості віртуального макетування, такі: вони дозволяють замінити фізичний прототип виробу його віртуальним аналогом та в процесі комп'ютерного аналізу електронного зразку розв'язувати ті задачі, що раніше вимагали натурних випробувань.

Віртуальним макетуванням можна супроводити весь процес проектування виробів, що дозволяє проводити тестування паралельно з розробкою, тобто – своєчасно знаходити та виправляти помилки та недоліки: віртуальний прототип можна



створити одразу після напрацьовування основних вимог до виробу та формування його концептуальної моделі, а в разі деталізації головної моделі його буде модифіковано (на відміну від фізичного макету, що може бути виготовлений тільки після завершення всіх етапів проектування та підготування виробництва); в концептуальному проектуванні віртуальний макет допомагає провести аналіз альтернативних підходів та обрати правильне рішення; в конструюванні віртуальний макет допомагає оцінити зовнішні форми складових частин, їх взаємне стикування та узгодженість у рамках єдиного виробу; в проектуванні виробів у розподіленому середовищі корпоративної мережі віртуальний макет підвищує наочність та спрощує процес управління проектуванням; у підготуванні виробництва віртуальне моделювання допомагає проконтролювати всі технологічні етапи ланцюжка виготовлення вузлів та збірок, та оцінити якість розробленої оснастки в режимі реального часу.

Замовник може використовувати віртуальний макет вузла чи виробу, що є частиною його продукції, в інженерних потребах. Віртуальне макетування дозволяє використовувати майстер-моделі виробів у галузях маркетингу, продажу, супроводження, навчання.

**САПР, інтелектуалізація автоматизованого проектування та *Internet*.** Обговоримо тенденції розвитку САПР на основі застосування інтелектуальних САПР в *Internet*.

Класичні методи САПР вичерпали можливості ЕОМ. Це вимагає доповнення фундаментальної концепції систематичних алгоритмів методами теорії експертних систем (ЕС): суттєвий вплив на продуктивність праці інженера здійснюють інтелек-

туальні САПР, які функціонують в глобальній мережі *Internet*. Але алгоритми та способи відображення, необхідні для створення інтелектуальних моделей, баз знань та інформаційних запитів, є складними та мало дослідженими. Найсильніше недоліки традиційних методів у порівнянні з інтелектуальним процесом автоматизованого проектування відчутні в концептуальній базі САПР.

По-перше, інженер працює так, що його мета є в певній мірі нечіткою, досвід використовується евристично, а алгоритмічне наближення до рішення є відтворенням шляху його прийняття. Цей метод не відтворюється класичними програмами. Прикладом є САПР прототипів аналогових радіотехнічних пристроїв, електро-двигунів, трансформаторів: проектувальнику потрібна ЕОМ для підвищення ефективності складної праці (враховування численних обмежень часто перевищує людські можливості, тому прийняті рішення можуть бути неякісними або їх важко знайти); ці САПР неможливо повністю описати за допомогою програм (звичайний алгоритм не здатен урахувати всю складність задачі, а в разі розгляду всіх прийнятих до уваги випадків програми стають надто складними або непридатними для реалізації).

По-друге, в разі класичних алгоритмів ймовірно отримати помилкове рішення через неправильні дані (при використанні складних алгоритмів автоматичне визначення адекватності даних є важкою задачею, яку подекуди неможливо розв'язати та яку часто не передбачає ПЗ).

У методичному плані недоліки традиційних САПР такі.

По-перше, існуюче ПЗ не впорядковане, а САПР як об'єкт дослідження недостатньо вивчена через традиційну стратегію автономної розробки САПР у межах кожної галузі. Згідно з

нею вимагається ефективність на рівні конкретно розглянутих задач. Якщо ж потрібно поєднати декілька програмних або апаратних засобів, найчастіше з'являються непереборні не-сумісності, які при створенні складних систем стають визначальними. Ця проблема вирішується завдяки розробці єдиних стандартів.

По-друге, має місце розростання та еволюція ПЗ навколо проекту в процесі його розвитку до розмірів, не передбачених специфікацією.

По-третє, намагання користувачів використати програми САПР найефективніше часто залишаються марними через те, що великі лістинги найчастіше не розкривають використаних моделей (доводиться обмежуватися документацією).

В ергономічному аспекті є такі недоліки. Якщо початково використання розробником традиційних методів САПР здійснюється як випадкова взаємодія з ЕОМ та прийняття рішень на робочому місці, то надалі САПР повинні підтримувати: роботу зі спеціальними знаннями, що забезпечують перевірку дій інженера; накопичування досвіду для його використання всією спільнотою (з персоніфікацією проміжних рішень та висновків конкретної людини, оцінкою її індивідуального вкладу).

Концептуально схему інтелектуальної САПР в *Internet* подано на рис.12.1. Її база знань (БЗ) відображає спеціальні знання проєктувальників у сукупності правил, яким повинен бути підпорядкований об'єкт у процесі проєктування.

Ці правила містяться в явному вигляді, не залежать від стратегії дій, припускають звертання, модифікацію та поповнення, не вимагаючи модифікації програмного ядра САПР.

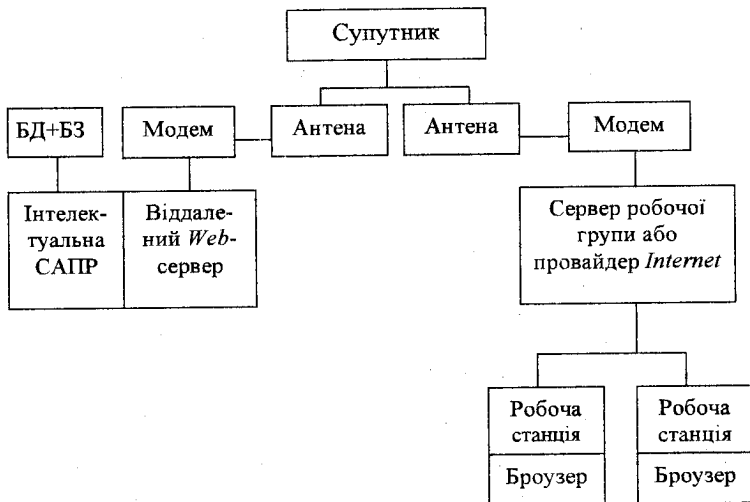


Рис. 12.1. Концептуальна схема інтелектуальної САПР в *Internet*

Природа цих правил така: обмеження, обумовлені існуючими нормами та виробничими процесами, специфікою використання конкретних пристроїв, економічними факторами; характер поведінки об'єктів та процесів проектування, що обумовлює відповідний тип математичної моделі; евристичні правила (наприклад, залежність методів розкрою листової сталі від особливостей виробничого процесу, механічні та ізоляційні обмеження, умови експлуатації).

Можливості інтелектуальних САПР покращують надійність технічних характеристик їх елементів, дозволяючи локалізувати помилкові рішення: невдалі спроби можна стерти, залишаючи кінцевий результат; можна запам'ятати ту частину спеціальних знань, що є принциповими проміжними рішеннями.

ЕС інтелектуальних САПР повинні бути гібридними, поєднуючи такі можливості: проведення складних обчислень при розв'язуванні задач у цифровій формі; проведення на ЕОМ логічних виводів, які базуються на обробці даних у символьній формі.

У разі тісних зв'язків між виробниками, успішність їх взаємодії визначає наявність моделей технологічних процесів, придатних для сумісного використання. В цьому плані доцільне використання інтелектуальних агентів, які дозволяють набагато краще зрозуміти можливості технологічного процесу, в рамках якого буде реалізовано проект (приводячи до прийняття рішення, краще підготованого до виробництва), та допомагають користувачам знайти потрібну інформацію.

Проблемою є відсутність загальноприйнятих методів обміну цією інформацією. В цьому плані *NIST (National Institute of Standards and Technology)* розробляє вимоги до специфікації технологічного процесу. Наприклад, ним створено базову архітектуру для інтеграції простих програмних агентів трансляції та розпізнавання з системою *Bentley Microstation Modeler*.

Інтеграція великої бази структурованих інженерних ресурсів та інтелектуального інструментарію агентів є надзвичайно складною дослідницькою задачею на стику інженерних дисциплін, штучного інтелекту, подання інженерних знань, автоматизованого проектування, мережних обчислень. Створення узагальненої архітектури агентів для підтримки основних технічних застосувань допомагає виробникам ПЗ створювати спеціалізовані агенти для вбудови в гетерогенні САПР.

Основним напрямком застосування *Internet* для САПР є сумісне використання інформації та співпраця інженерів. Але для САПР та відповідних додатків типові великі розміри. Виходячи з повільності протоколів *WWW* та не дуже ефективних засобів доступу до інформації, *Internet* не впорається з передаванням великих обсягів інженерних даних відкритою мережею з метою їх перегляду без нових технологій. Провідним напрямком застосування інтелектуальних агентів є фільтрація інформації для швидкого її постачання інженеру, не відволікаючи його від основних справ.

Для практичного застосування САПР із мережними функціями велику роль відіграють питання безпеки та довіри.

Наприклад, розглянемо виробничі сервіси, керовані замовником.

Якщо постачальник деталей опублікує електронний каталог параметризованих виробів, які користувачі САПР можуть обирати для розміщення в своїх проектах, у виробників замовлених компонентів можуть виникнути дві характерні ситуації щодо функціональності та поведінки їх виробів: вони не дуже важливі для проекту в цілому; вони критично важливі для проекту (наприклад, критичні вузли в аерокосмічній промисловості).

У другому з розглянутих вище випадків вимоги до каталогу повинні бути суворішими через можливі проблеми: через надзвичайну важливість деталі, її можуть застосувати в проектній організації до відправки виробнику заявки на неї (з метою аналізу, модифікації під конкретні потреби користувача та запуску в технологічний процес), але імовірно таке, що виробник не зможе виготовляти потрібну деталь, тобто створена конструкція не зможе відповідати потрібним вимогам.

Ще один наочний приклад – забезпечення інтелектуальної власності та авторських прав.

Деякі користувачі не погодяться розкривати себе навіть за умови мінімального ризику, остерігаючися втрати корпоративної інформації, фінансового краху, наслідків судового розгляду катастроф.

Багато виробників не використовують навіть активні параметричні моделі – деталі, що здатні самі підтримувати внутрішні обмеження та дозволяють виробнику слідкувати за тим, хто і коли ними користується.

Розподілені ЕС, не вирішуючи всіх проблем, розширюють можливості класичних САПР у будь-якій галузі наукових досліджень та матеріального виробництва, згладжуючи багато недоліків завдяки використанню модульних структур:

- алгоритмічні методи розв'язування задач доповнюються евристичними підходами;

- здійснюється вплив на загальну архітектуру систем, розширюючи її та роблячи гнучкішою, придатною для евристичних та алгоритмічних методів (це дозволяє об'єднати переваги алгоритмічної САПР та ЕС, зменшити витрати на підготування персоналу та скоротити обсяг робіт у різних галузях практичної діяльності);

- ці САПР є стійкими в розвитку, оскільки вони передбачають власну динаміку, підтриману надійністю правил виводу;

- спеціальні знання ясно висловлені та легко доступні, підприємство може безпосередньо керувати їх змістом, а розробник – персоналізувати власні спеціальні знання;

– можна зберігати інженерні дані САПР у корпоративних БД та забезпечувати доступ і маніпуляцію даними користувачам через *Web*-броузери чи додатки, що знаходяться на ПК.

Легкість переходу конструкторів та користувачів на модель функціонування САПР в *Internet* залежить від: традицій (організаційних, культурних та інших); необхідності виводити на ринок та вбудовувати в робоче середовище користувача нові інструменти; обмежень у застосуванні сучасних технологій САПР (евристичні та алгоритмічні методи важко зв'язати між собою, оскільки програмування ЕС засобами алгоритмічних мов є неефективним, а традиційні мови ЕС (Лісп, Пролог) малоефективні для розрахунків; інтерфейси між інженером та користувачем мають приватний характер та важкодоступні для інженера).

У зв'язку з виходом САПР в *Internet*, прогресивною тенденцією стала розробка додатків САПР на базі мови програмування *Java* та *Java*-технології.

Наочним прикладом є *bCAD API* – використання технології *Java* фірмою *ProPro Group* (Новосибірськ) за програмний інтерфейс до 3D-САПР середнього класу *bCAD* для *Windows/NT/XP*: користувачі та сторонні розробники можуть створювати додатки з використанням усього набору інструментів креслення та об'ємних побудов, доступних на поточний час в інтерактивному режимі; використання віртуальної *Java*-машини дозволяє суміщати гнучкість об'єктно-орієнтованого підходу, потужні мовні засоби, захищеність ядра та даних головної системи від помилок та збоїв додатків; при програмуванні можна використовувати всі



популярні системи розробки *Java*-додатків (*Microsoft JDK*, *Microsoft Visual J++*, *Symantec Visual Cafe* тощо); використання технології *JIT (Just In Time)* забезпечує високу ефективність виконання додатків, особливо для систем, побудованих на традиційних інтерпретаторах (*Visual Basic*, *LISP* та інших); побудову елементів інтерфейсу користувача можна здійснювати за допомогою пакету *AWT (Abstract Windows Toolkit)* або бібліотеки *Microsoft AFC (Application Foundation Classes)*; можливе не тільки виконання програм, написаних мовою *Java*, а й виклик процедур із *DLL*, розроблених із використанням інших мов (це дозволяє сполучати *bCAD* із високоефективними застосуваннями типу систем розрахунків на міцність, що можуть використовуватися паралельно на декількох процесорах та навіть окремих комп'ютерах під управлінням *Windows NT/XP* або *UNIX*).

Ще один приклад – ПЗ *Engineering Back Office* компанії *Bentley Systems Inc*, яке реалізоване в мережному середовищі з архітектурою клієнт/сервер та містить сімейство серверів додатків *ModelServers*: завдяки серверам *Publisher*, *TeamMate* та *Continuum* можна зберігати інженерні дані в корпоративних БД та забезпечувати доступ до них користувачів через *Web*-браузери чи додатки, що знаходяться на ПК.

Це ПЗ є дуже актуальним в умовах зростання інтересу до архітектури клієнт/сервер з боку тих відділів, які відповідають за впровадження інформаційних технологій при побудові інформаційних систем масштабу підприємства. Серія серверів додатків *Engineering Back Office* дозволяє застосовувати всі переваги архітектури клієнт/сервер в інженерних проектах.

ПЗ *Engineering Back Office* переносить деякі логічні принципи, властиві *MicroStation* та додаткам для нього, на сервери додатків. У підсумку середовище проектування є прозорішим та додає всі переваги архітектури клієнт/сервер до переваг *CAD* (гнучкості, надійності тощо) у взаємовідносинах між проектними відділами та підприємством у цілому.

Серія продуктів *MicroStation* компанії *Bentley Systems Inc* залишилась інструментарієм для клієнтів на ПК. У той час, як головною функцією продуктів серії *ModelServers* є обробка запитів, які відносяться до інженерних моделей.

Сервер *ModelServer Publisher*: конвертує файли формату *DGN* системи *MicroStation* та файли формату *DWG* системи *AutoCAD* в *Web*-формати (*SVF*, *CGM*, *VRML*); надає ПЗ для *Web*-серверів компанії *Netscape Communication Corporation* (*FastTrack Server* у разі фіксованої ліцензії та *Enterprise Server* у разі багатокористувацької ліцензії), призначене для зв'язку *Web*-сторінок у форматі *HTML* із реальними кресленнями.

*Web*-сервери, що використовують *ModelServer Publisher*, забезпечують користувачам, які мають в розпорядженні лише *Web*-броузер, прямий доступ до креслень (рис. 12.2).

Безсумнівні переваги цієї серверної архітектури в порівнянні з простим використанням *Web*-броузерів, які працюють з *DGN*- та *DWG*-файлами, такі.

По-перше, користувачу не потрібно завантажувати спеціальні додатки для обробки форматів файлів *CAD*-систем: використовуються формати графічних файлів, які є стандартними для *Internet*.

По-друге, в традиційному варіанті, якщо недостатньо простого перегляду креслень форматів *DGN* або *DWG*, дово-

диться завантажувати зв'язані дані (файли посилань, файли ресурсів тощо). Використовуючи *ModelServer Publisher*, *Web*-сервер передає графічну інформацію не до всього файлу креслення, а виключно до запитаного виду *CAD*-моделі. Це зменшує навантаження на мережу та підсилює безпечність (інженерні дані залишаються на сервері та не можуть бути змінені).

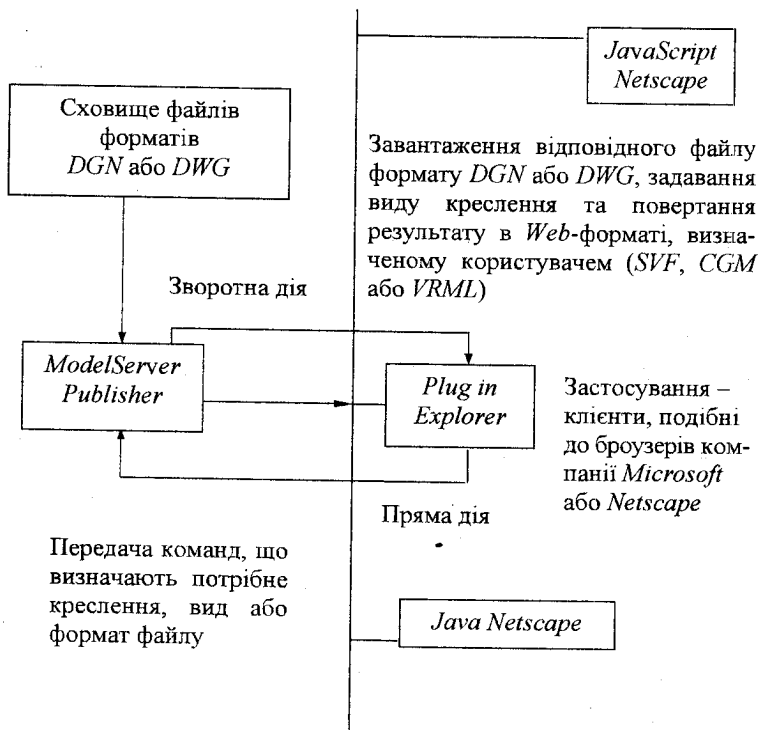


Рис.12.2. Схема функціонування *ModelServer Publisher*

*ModelServer TeamMate* є розширеною версією *MicroStation TeamMate* для серверу, призначеного для ведення докумен-

тообігу та управління проектом, яка забезпечує користувачів *MicroStation* та броузерів *Internet* уніфікованими засобами доступу до файлів креслень, перегляду внесених змін тощо.

Якщо використовувати *ModelServer TeamMate* в якості серверу застосувань, то доступ до інформації про проекти, що міститься в файлах, легко розподіляється між усіма клієнтами відповідно заданій ієрархії.

*ModelServer TeamMate* надає середовище, що легко настроюється на корпоративні стандарти. Воно є зрозумілим для партнерів через те, що засобами взаємодії в процесі створення проекту є стандартні *HTML*-форми.

Домашня сторінка проекту, що є доступною всім членам робочої групи, може містити спеціальні поля *ModelServer TeamMate* для перегляду виконаних змін, файлів креслень, специфікацій, пошуку файлів проекту тощо.

*ModelServer TeamMate* надає засоби імпорту та експорту файлів різних форматів (*DWG*, *DXF*, *IGES*, *SAT*, *DGN* тощо) відносно бази даних проекту, наприклад, автоматичне конвертування файлу формату *DGN* для *MicroStation* у файл формату *DWG* для *AutoCAD* та навпаки.

Завдяки виконанню операцій конвертування сервером, виключена можливість внесення помилок користувачами-членами проектної групи.

*ModelServer Continuum* надає можливість тим клієнтам, які використовують *MicroStation*, зберігати графічну та текстову інформацію проекту не в файлах креслень, а в реляційних БД.

Щодо сеансу редагування цієї інформації користувачем, то: спочатку він формує запит до БД, з якої відокремлюється

інформація, необхідна для створення тимчасового комплексу файлів креслень (його застосовують в подальшій роботі); основний сеанс роботи є традиційним для середовища *MicroStation*; наприкінці роботи *ModelServer Continuum* поміщає змінену інформацію до БД.

*ModelServer Continuum* не вимагає логічного чи геометричного розподілу даних по окремим файлам, надаючи суттєві переваги в роботі з великими проектами, особливо з геоінформаційними. Прикладом додатку типу "клієнт" для *ModelServer Continuum* є *MicroStation Geographics*.

Завдяки стандартним *SQL*-запитам та форматам даних, сервер *ModelServer Continuum* забезпечує рівноправний доступ до графічних даних як тих клієнтів, які приймають участь у створенні проекту, так і тих, які безпосередньо не приймають у цьому участі.

Графічні характеристики зберігаються в БД серверу *ModelServer Continuum* як записи реляційної БД разом із тими атрибутами, що до них відносяться.

Із записами може бути пов'язаний будь-який документ, що дозволяє легко керувати не тільки графічними характеристиками, а й файлами принципових схем, керівництв із експлуатації, растрових зображень тощо.

Для забезпечення користувачів програм-клієнтів повним доступом до засобів *Engineering Back Office*, компанією *Bentley Systems* надаються засоби розробки додатків для кожного елементу *Model-Server*: специфікації під загальною назвою *Open Engineering Connectivity* дозволяють користувачам та розробникам налаштовувати та розширювати можливості існуючих клієнтів та створювати нових клієнтів для *ModelServer* на базі

будь-якого середовища програмування (*C++ Builder, Visual C++, Java, Visual J++, Visual Basic* тощо).

### Питання для самоперевірки

1. Дайте визначення та коротко охарактеризуйте практичне застосування інженерної комп'ютерної графіки.
2. Поясніть призначення загальноприйнятих скорочених термінів *CAD, CAM, CAE, AEC, GIS, PDM*.
3. Сформулюйте особливості апаратно-програмної платформи інженерних САПР.
4. Дайте загальну характеристику прикладного математичного апарату інженерної комп'ютерної графіки.
5. Яку роль в інженерній комп'ютерній графіці відіграють технологія віртуальної реальності та інтелектуальні САПР?

### 12.3. Огляд базових класів інженерних САПР

**Машино-, приладо-, судно- та авіабудівні САПР.** До вказаної категорії САПР належать системи, призначені для розробки станків, агрегатів, автомобілів, мотоциклів, велосипедів, приладів, суден, літаків, аерокосмічних апаратів та іншої численної продукції або її складових компонентів. Для більшої лаконічності та узагальненості назви цих систем, надалі будемо вживати для них термін "САПР машинобудівного профілю".

САПР машинобудівного профілю характеризуються великим спектром представників (від простих "комп'ютерних кульманів" до великомасштабних систем) та динамічністю [79, 119, 120].

Основною тенденцією розвитку САПР машинобудівного профілю є перехід від багатомодульних до багатокомпонентних систем, сформованих із окремих програм незалежних постачальників.

Це викликано: зростанням потреб виробництва в невеликих, добре споряджених інженерних центрах та в максимальному скороченні виробничих витрат на всіх етапах створення продукції для зниження її собівартості; тим, що багатокомпонентні системи часто надають рівноцінні або більші можливості за значно меншу вартість.

Але великомасштабні САПР машинобудівного профілю ще не зійшли з арени, а в деяких галузях застосування є беззаперечними лідерами завдяки розповсюдженню в світі, тривалому терміну розробки та експлуатації програм (гаранту надійності ПЗ), багатоплатформовості. Наприклад, для моделювання складних вузлів та виробів, які містять сотні та тисячі деталей, просто необхідні декілька великомасштабних САПР та потужних RISC-комп'ютерів.

Лідером великомасштабних САПР машинобудівного профілю є програмний комплекс *ProEngineer* фірми *PTC*. Іншим європейським лідером є подібна система *Euclid* фірми *Matra Datavision*. Конкурує з фірмою *PTC* САПР-продукція американської компанії *SDRC* (наприклад, система *I-DEAS*), яка має беззаперечні переваги в авто-, авіа- та суднобудуванні.

Оскільки в цілому можливості великомасштабних САПР машинобудівного профілю однакові, зупинимося на перевагах систем-лідерів.

САПР *ProEngineer* – одна з найкращих у моделюванні великомасштабних збираних виробів. На всіх рівнях моделювання в ній реалізовано можливість параметризації, тобто заміни в геометричній моделі деталі конкретних координат або розмірів символічними іменами-параметрами (можна породжувати низку деталей виключно шляхом змінювання значення однієї величини-параметру, аналогічно підпрограмам у програмуванні). Функціональні можливості базової системи можна розширяти за допомогою декількох десятків спеціалізованих модулів, забезпечуючи максимум вимог машинобудівного виробництва. Високі стабільність та якість системи привели до багатомільйонних контрактів на її закупівлю з боку фірм *Lucent Technology*, *Tetra Pack* та *Lockheed Martin Astronautics*, контрактів із багатьма іншими фірмами.

Комплекс САПР *Euclid* фірми *Matra Datavision* за більшістю функціональних можливостей подібний до системи *ProEngineer*, але версія *Euclid Quantum* має вищу конкурентну здатність за ціновим показником (у першу чергу – базових модулів). Про переваги цієї системи свідчать давній контракт *Matra Datavision* із автомобільною фірмою *Renault*, великий контракт із корпорацією *British Telecom* та інші.

Система *I-DEAS* компанії *SDRC* має потужні модулі *CAE*, а саме: один із найкращих генераторів сітки кінцевих елементів, модулі лінійного та нелінійного кінцево-елементного аналізу, унікальні підсистеми обробки експериментальних даних для настроювання розрахункових програм. Проте



відсутні багато модулів, пов'язаних безпосередньо з виробництвом – САМ-підсистеми.

Усі великомасштабні САПР машинобудівного профілю мають суттєві недоліки: високу вартість, що обмежує область їх застосування; високу складність; суттєві обмеження можливостей.

Основний недолік цих систем пов'язаний зі складністю, дискомфортною та нестійкістю до помилок (наприклад, через втрату точності може викривитися форма виробу), важкістю переходу між поверхневими та твердотільними моделями. Необхідність цього переходу пов'язана з тим, що поверхнєве та твердотільне моделювання реалізовані по-різному: для роботи з об'єктами, що обмежені складними поверхнями, в складі САПР є окремі модулі поверхневого моделювання; твердотільне моделювання реалізоване набором булевих операцій над 3D-примітивами.

Ці недоліки призводять до фінансових негараздів виробників великомасштабних САПР машинобудівного профілю, незважаючи на солідні корпоративні замовлення. Ситуація стала особливо гострою зі зростанням продуктивності дешевих ПЕОМ та розвитком нової галузі в наданні високотехнологічних послуг – *Design On Demand* (проектування на замовлення).

Оскільки малі виробничі та проектні фірми утворили масовий та ємкий ринок послуг, вимогливий до співвідношення між ціною та ефективністю САПР, то популярності набули активні новаторські компанії, що пропонують САПР машинобудівного профілю середнього рівня.

Найкращим прикладом масової системи середнього рівня є САПР *AutoCAD* фірми *Autodesk* [9, 17, 125-128, 140, 161, 164], а фірм-виробників систем цього класу – *Autodesk* та *Bentley Systems* (з її інтегрованими рішеннями на базі ядра *MicroStation*).

На титул лідера ринку САПР машинобудівного профілю середнього рівня також претендують: *SolidWorks* та *SolidEdge* фірми *Intergraph*, *PT Modeller* фірми *PTC*, *Mechanical Desktop* фірми *Autodesk*. Між ними є значні розбіжності. Наприклад, найдорожчими є *PT Modeller* та *Mechanical Desktop* (відповідних видатних якостей вони не мають). На тестах побудови твердотільної 3D-моделі складної конфігурації найкращий час показали *SolidEdge* та *SolidWorks*, користувачі *Mechanical Desktop* стикнулися з проблемою моделювання схилів, а користувачі *PT Modeller* – із проблемами неінтуїтивної процедури побудови тонкостінних тіл та неможливості створення заокруглень між елементами різних поверхонь. Непогані показники демонструє САПР *VX* на задачах розробки повністю параметризованих моделей з урахуванням та видаленням можливих колізій: час роботи – такий, як у *SolidEdge* та *SolidWorks*, при цьому дві третини часу йдуть на формування системи параметрів та відношень.

Вибір САПР не визначається тільки ім'ям, статусом або популярністю компанії-виробника ПЗ. Слід звертати увагу і на системи нові або такі, що не заявляли про себе.

Так, американо-італійська САПР *Eureca Gold* фірми *Cadlab* із досвідом роботи на європейському ринку активно завойовує американський та азійський ринки. Ця система, на відміну від *SolidWorks*, призначена для конструкторів, а не для

дизайнерів. За функціональними можливостями вона не поступається лідерам своєї категорії, забезпечуючи підсистеми: 2D-креслення з підтримкою асоціативності між моделлю та кресленням, 3D-моделювання з повною параметризацією, збиране моделювання, роботу з поверхнями, ескізне моделювання, роботу з листовим металом тощо.

Претендує на роль лідера і система *IronCad* однойменної фірми. Вона поступається САПР *Eureca* в зручності роботи з поверхнями, але базується на принципово нових прогресивних запатентованих технологіях формування 3D-моделей.

Ще одним претендентом на лідерство є САПР *Varimetrix* (популярна в Великобританії та Канаді). Вона є мультиплатформовою та має такі особливості: побудова ядра 3D-моделювання на базі уніфікованої параметричної геометрії (УПГ), що робить рівноцінними всі форми подання моделі та забезпечує виконання переходів між поверхневими та твердотільними моделями однією командою; потужні засоби роботи зі складними поверхнями; вбудована підсистема моделювання збірок; модуль формування керувальних програм для 2-5-координатних станків із ЧПУ.

Найгостріша проблема розробників машинобудівних САПР середнього рівня – залежність від ліцензованих компонентів: ліцензійні відрахування збільшують вартість продуктів, а можливості внесення принципових змін в якісні показники систем, подібні до швидкості, залежать від власників компонентів.

Наприклад, вихід нового ядра 3D-моделювання типу *ACIS* дозволив суттєво підвищити показники *AutoCAD* та швидкодію ще багатьох САПР, які базуються на технології *ACIS*

(бібліотеці геометричних програм фірми *Spatial Technology*). З іншого боку, поява 3D-ядра *ParaSolid* з високою швидкістю та можливостями, відсутніми в *ACIS*, привела до очевидних переваг у моделюванні складних поверхонь для САПР, які базуються на *ParaSolid*.

У вигідному становищі знаходяться ті компанії-розробники САПР, що застосовують 3D-підсистеми власної розробки: користувачі цих САПР отримують такі переваги, як порівняно невисокі ціни, гарний рівень підтримки та навіть більші можливості ПЗ.

Серед конструкторських додатків для розв'язування загальних та спеціальних задач проектування слід виділити САПР компаній *Rebis* та *Orange Technologies*, призначені для проектування заводів, дослідних та промислових установок.

Найвідомішими *CAM*-системами, тобто системами автоматизації виробництва на етапі підготування керувальних програм для станків із ЧПУ, є: *SURFCAM* компанії *SURFWARE*; *hyperMILL* компанії *OPEN MIND Surfware Technologies*; *AdvaNC* російських виробників.

Щодо такого етапу проектування, як розрахунок та аналіз конструкцій виробів, то широко відомими є *CAE*-системи американської корпорації *MSC* (*Nastran*, *Patran*, *Fatigue*, *Dytran*, *SuperForge*, *ABAQUS*, *Mvision*, *InCheck* тощо).

Цікавими розробками країн СНД є *CAD/CAM*-системи "КОМПАС" та "КРЕДО", *CAD/PDM*-система *T-FLEX CAD*, *CAM/PDM*-система "ТехноПро", *CAD*-системи *bCAD* та *ZCAD*, універсальні системи технологічного проектування "СІАП-ТТ" та "Темп", інтегрований програмний пакет оперативного планування "Фобос" та інші розробки (наприклад, під *AutoCAD*).

**Архітектурні та будівельні САПР.** Для розуміння важливості для розвитку економіки САПР цього класу, систематизуємо вимоги, що висуває практика до ПЗ для автоматизації архітектурно-будівельного моделювання та проектування.

Через численність цих САПР, у першу чергу їх порівнюють за загальними характеристиками, а також за ступенем підтримки специфіки та ключових етапів архітектурно-будівельного проектування, тобто – за можливостями розв'язування задач: проектування промислових будівель та споруд, житлових та суспільних будівель, проектування та розстановки меблів, створення інтер'єрів, реставрації пам'ятників архітектури, створення робочої документації. Ці задачі вимагають від архітектурно-будівельних САПР таких функціональних можливостей [34, 79, 119, 120, 155]:

– *моделювання та проектування стін* (розбиття, маркування та оброзмірювання помешкань, відрисовування стін, підпорних стінок та пандусів, обробка відрисованих стін, ведення бібліотек стилів стін, штрихування та редагування стін), *стінних отворів та пройомів* (оброзмірювання, відрисовування чвертей), *дверних, віконних та балконних пройомів* (відрисовування компонентів пройомів, їх маркування, редагування, ведення бібліотек пройомів та їх окремих компонентів, створення відомостей), *сходів, ескалаторів та ліфтів* (параметричне відрисовування сходів, перил, балюстрад, ескалаторів та ліфтових вузлів, розрахунок параметрів сходів, ведення бібліотеки блоків ліфтових вузлів), *покрівель* (загальне настроювання, відрисовування стандартних типів покрівель, побудова та обробка поверхонь покрівель складної конфігурації, штрихування простих та складних поверхонь

покрівель, розрахунок конструкцій дахів, відрисовування, розрахунок, специфікації стропильних конструкцій, редагування покрівель), *конструктивних елементів* (стелі, підлоги, балки, колони, каміни, димові труби);

– *об'ємне моделювання (тіні, перспектива, аксонометрія, анімаційні фліки)*;

– *моделювання та проектування поверхів* (встановлення, контролювання, змінювання поточного режиму видимості), *3D-фасадів* (оформлення, швидке отримання), їх *2D-розгортки по стінах* (різні способи отримання результатів проектування вікон та дверей, виставлення позначок) та *розрізів* (2D- та 3D-, з позначками та маркуванням), *приміщень* різних форм та маркувань (створення, обробка, редагування, корегування площі, конвертування до стін, штрихування, формування в списки);

– *створення текстових документів* (підтримка різних способів формування та генерації відомостей, специфікацій та експлікацій з можливістю попереднього корегування здобутої інформації; підтримка різних форматів отримання даних);

– *підтримка розрахунків та перевірок площ та об'ємів*;

– *підтримка роботи з бібліотеками* (розвинена система ведення параметричних та символічних (блочних) бібліотек).

Повинна бути надана можливість враховування специфіки: різних типів (стилів) помешкань, стін, підпорних стінок, пандусів, стінних отворів та проїомів, дверних, віконних та балконних проїомів, сходів, ескалаторів, ліфтів, дахів та покрівель, конструктивних елементів; місць їх розташування в будівлі.

*Розглянемо перелік найтиповіших представників архітектурно-будівельних САПР.* Формування цього переліку обумовили ті фактори, що дозволяють класифікувати та розставити пріоритети серед численних архітектурно-будівельних САПР: апаратно-програмна база, на якій може функціонувати САПР (у першу чергу – тип комп'ютеру та операційної системи), рівень універсальності (комплексності) САПР, її ціна та розповсюдження на практиці.

За типом комп'ютеру та ОС виділяють пакети, що орієнтовано: на ПК із ОС *DOS* та *WINDOWS*; на потужні дорогі графічні станції, наприклад, *Sun* із ОС *UNIX*, *Apple* з ОС *Macintosh*.

Універсальність (комплексність) інженерних архітектурно-будівельних САПР – це можливість виконання в межах одного програмного комплексу робіт, які належать до різних класів задач проектування, одночасного виконання розрахункових та графічних задач.

Фактор розповсюдження цих САПР залежить від використаної графічної платформи, рівня комплексності, локалізованості та перспектив розвитку пакету, надійності супроводження.

По-перше, відзначимо широко відомі потужні САПР *ArchiCAD* та *CADdy* [34, 79, 119, 120, 155], які вважаються безперечними лідерами.

По-друге, відзначимо системи, що інтенсивно завойовують свою частку світового ринку – САПР-пакети фірми *Intergraph*: *Architron*, *AllPlan*, *UniCAD*, *CADKey*, *Architectural Office*. Вони заслуговують високої оцінки та інтенсивно застосовуються зарубіжними користувачами, але на вітчиз-

няному ринку не дають прийняттого співвідношення між ціною та можливостями та не мають шансів до широкого розповсюдження там, де повсюди є *AutoCAD*.

По-третє, важливим класом архітектурно-будівельних пакетів є САПР на базі системи *AutoCAD* для ПК. Наприклад, це європейські та американські лідери *ACADMAP* та *ArchPro* американської фірми *Eagle Point*, *AutoArchitect* американської фірми *SOFTDESC*. Цей стан справ обумовлений: розповсюдженістю пакету *AutoCAD*; його статусом фактичного світового стандарту САПР; кількістю вказаних систем та багатими можливостями прикладних програм, які працюють у середовищі *AutoCAD*, для підтримки архітектурних додатків; доступністю цін. Найбільшу цінність мають САПР на базі тих версій *AutoCAD*, в яких у повній мірі реалізовано об'єктно-орієнтований підхід до роботи з додатками.

Відзначимо найвідоміші розробки на базі *AutoCAD* країн СНД: "МАЕСТРО-А" (м. Київ), *ARKO* (фірма "АПИО-ЦЕНТР", м. Москва), "АРФА-КАД" (фірма *VIKS*, м. Нальчик).

Ще одним напрямком архітектурно-будівельних САПР є пакети для широкого вжитку, наприклад, відомі системи *3D Home Architect*, *FloorPlan* та *FloorPlan Plus 3D*.

Особливо відзначимо пакети для архітектурного та промислового дизайну. Їх яскравим представником є *3D Studio VIZ*, який продовжив лінію *3D Studio* фірми *Autodesk*.

**САПР електротехніки та електроніки.** Ці системи отримали на практиці велику кількість узагальнюючих назв, які відбивають різні аспекти їх застосування: САПР електричних ланцюгів, схем та мереж; САПР електричних, електронних та радіоелектронних схем, пристроїв та приладів; радіотехнічні та



радіоелектронні САПР; САПР електронної, радіотехнічної та радіоелектронної апаратури тощо [79, 119, 120]. Надалі будемо дотримуватись єдиної термінології, що є найбільш лаконічною, місткою та найчастіше застосовується в фахових комп'ютерних книжках та періодичних виданнях – назви *САПР електротехніки та електроніки*.

*Автоматизоване проектування електротехніки та електроніки* – це процес, який використовує апаратне та програмне забезпечення комп'ютеру для специфікації, проектування, моделювання, верифікації та документування електричних та електронних виробів.

Широко відомими та розповсюдженими на практиці програмними системами САПР електротехніки та електроніки є: *P-CAD, OrCAD, Electronics Workbench, MicroCap*.

У цілому, САПР електротехніки та електроніки можна умовно поділити на такі групи.

*По-перше, це системи аналогового проектування.* Більшість комерційних та дорогих систем цієї групи є інтерактивними надбудовами над відмінною розрахунковою програмою *PSpice* (розробка університету Берклі), які доповнено бібліотеками моделей промислових електронних компонентів (у цій групі найкращою з точки зору зручності роботи та відповідності результатів моделювання практиці вважають систему *MicroCap*).

*По-друге, це системи цифрового проектування.* В цій галузі існує близько тисячі різноманітних комерційних та *free-ware* програмних комплексів, які можна стисло охарактеризувати, як "VHDL, верифікація, імітація, синтез" (було прослідковано тенденцію рішучого переходу систем цієї групи на плат-

форму *Windows* та такого ж рішучого відступу на попередні *Unix*-позиції).

Прогресивним напрямком розвитку підприємств радіоелектронної промисловості стало впровадження сучасних систем автоматизованого проектування електроніки – *EDA* (*Electronic Design Automation*).

### *Розглянемо історію розвитку САПР електроніки.*

Автоматизація проектування електроніки почалася на рубежі 60-70-х років ХХ сторіччя: були створені перші промислові системи, що дозволяли розробляти печатні плати та інтегральні схеми, але недостатня потужність комп'ютерів та обмежені можливості ПЗ ще не дозволяли здійснювати проектування належним чином.

Процес проектування тривав багато місяців завдяки трудомісткості його окремих етапів та організаційних неузгоджень. Дійсно: на першому етапі розробляли дуже детальну специфікацію, що містила ретельний опис характеристик створюваного виробу; на другому етапі на основі специфікації готували схематичний опис проекту; на наступних етапах виконували копітку роботи з інтерактивного трасування та підбирання елементів виробу. Для верифікації печатної плати виготовляли її фізичний макет та тестували його вручну. Якщо ж виріб працював не згідно з поставленими до нього вимогами, то процес повторювали: трасування, підбирання елементів, виготовлення прототипу.

У процесі проектування, за різні його етапи (розробка схеми, трасування, виготовлення макету) відповідали різні відділи, документація між якими просувалася через спеціальне бюро. Це бюро перевіряло документацію на відповідність стан-

дартам та прийнятим на підприємстві обмеженням (рекомендований перелік компонентів та матеріалів, конструкторсько-технологічні обмеження тощо) тривалий час, який міг перевищувати час проектування.

Восьмидесяті роки ХХ сторіччя ознаменувалися різким зростанням складності проектів, значним скороченням життєвого циклу електронних виробів та жорсткою конкуренцією в електронній промисловості. Це привело до необхідності докорінного змінювання підходів розробників до проектування. А саме – до створення та застосування апаратно-програмних засобів проектування як систем нового покоління, що акумулюють досвід провідних розробників електронних компонентів та забезпечують розв'язування таких задач: різке скорочення циклу проектування та часу виходу на ринок завдяки використанню нових програмно-технічних рішень та вбудованих засобів управління процесом паралельного проектування; підвищення якості виробів, які випускають; повна автоматизація процесу проектування на базі наскрізної інтеграції ПЗ; можливість створення віртуального прототипу та ПЗ для його тестування та відлагодження, що скорочує витрати, пов'язані з виготовленням фізичного макету; створення розвинених БД та бібліотек стандартних компонентів.

Саме у 80-х роках сформувалися ті визначальні риси ринку ПЗ систем *EDA*, що забезпечили сучасний рівень розвитку засобів обчислювальної техніки та виробів електроніки.

*Розглянемо короткий огляд стану та структури ринку програмних комплексів САПР електротехніки та електроніки.* Це – величезний сектор світового програмного ринку. За-

пропоновані ним програмні комплекси САПР електротехніки та електроніки можуть задовольнити найвибагливішого клієнта.

Існуючі різноманітні системи *EDA* покривають весь спектр задач проектування електричних та електронних виробів і можуть бути умовно розділені на такі класи:

- засоби функціонального опису в процесі автоматизованого проектування електричних компонентів та електронних компонентів (*CAE* – високорівневий опис, вентиляно-рівневий опис);

- структурне моделювання електричних та електронних компонентів (імітаційне моделювання засобами мов *VHDL*, *Verilog*, *HDL* вентиляного рівня; моделювання електричних ланцюгів (аналогових та аналогово-цифрових));

- синтез електронних компонентів (синтез *ASIC*, програмованих логічних та вентиляльних матриць);

- проектування печатних плат та багатокристальних модулів (розміщення та трасування);

- проектування інтегральних схем (розміщення та трасування, порівнявє планування, проектування замовлених інтегральних схем);

- інструменти для розробки блоків;

- спеціалізовані інструменти та утиліти (верифікаційні та аналітичні інструменти, засоби автоматичного тестування та управління процесом проектування).

Серед САПР електротехніки та електроніки є як глобальні інтегровані системи, так і програмні утиліти-модулі, що реалізують окремі функції проектування (наприклад, тільки трасування печатних плат або тільки логічний синтез). Багато інтегрованих систем містить набори програмних модулів, які

надають засоби наскрізного проектування електронних виробів. Так що справа тільки в тому, щоб обрати доступний за ціною програмний продукт.

Коротко охарактеризуємо місце систем *EDA* в світі програмного бізнесу та напрямки діяльності їх провідних розробників.

За даними Міжнародної асоціації розробників систем автоматизованого проектування електроніки *EDAC (Electronic Design Automation Companies)*, обсяг ринку електронних САПР (прибуток від продажу програмних продуктів, консультацій та сервісних послуг) та оборот десяти найбільших компаній вимірюються мільярдами доларів. Згідно світового рейтингу, лідерами в світі систем *EDA*, з великим відривом від інших, є міжнародні компанії *Cadence Design Systems* та *Mentor Graphics*.

*Cadence* є однією з небагатьох компаній, які випускають інтегровані програмні комплекси з реалізацією повного маршруту проектування для всього спектру електронних приладів та компонентів: від будь-якого типу інтегральних схем до комп'ютерних систем, мережного та телекомунікаційного обладнання.

Головною відмінністю програмних комплексів *Cadence* від продукції інших компаній, що практикують обмін даними між окремими додатками на рівні стандартизованих зовнішніх інтерфейсів, є інтеграція нових ідей та набутих технологій до основного ядра (для цього переробляють відповідні програмні модулі, доповнюють та модифікують за необхідності внутрішню структуру обміну даними).

Ця стратегія обумовлена такими міркуваннями. Можна мати набір чудових модулів, які ефективно розв'язують локальні задачі (моделювання схем, розміщення та трасування

компонентів на платі, розрахунок теплових та перехресних перешкод тощо), але якщо вони є окремими програмами, то витрати часу на передавання даних із одного пакету програм в інший можуть зробити процес проектування недостатньо ефективним.

Глобальний підхід до реалізації кожного рішення вивів багато з програмних розробок фірми в ранг фактичних стандартів для виробників електронної апаратури. Прикладом є мова опису високого рівня *Verilog*.

Фірму *Cadence Design Systems* характеризує те, що: в центральному офісі фірми в США та в численних філіалах по всьому світу працюють декілька тисяч висококваліфікованих фахівців; фірма активно співробітничала з провідними розробниками стратегічно важливих *EDA*-технологій, об'єднуючися з найперспективнішими (наприклад, з фірмою *Cooper&Chyan Technology*, що є розробником трасувальнику *SPECCTRA* – невід'ємного компоненту багатьох популярних пакетів фізичного проектування, створених третіми фірмами); до переліку клієнтів фірми входять найкращі компанії сфери високих технологій (*Motorola, Siemens, Apple Computer, IBM, AT&T, Intel, NEC, Philips, Sun Microsystems, Silicon Graphics, Toshiba, Fujitsu*); засобами програмних пакетів фірми розроблено багато популярних високотехнологічних електронних пристроїв (процесори сімейств *Alpha AXP* та *PowerPC*, комп'ютери *Apple* та *UNISYS* тощо).

Компанія *Mentor Graphics* пропонує комплексне рішення для кожного з типів мікроелектронних пристроїв. Вона є ініціатором та розробником методології проектування інтегрованих систем *ISD*, яка дозволяє створювати програмно-апаратні

рішення "систем на кристали" (SOS) та "систем на платі" (SOB). Ця компанія має штаб-квартиру в США та декілька тисяч висококваліфікованих фахівців в офісах по всьому світу.

Показником активності на світовому ринку САПР електротехніки та електроніки в США є *Needham index* – сума вагових часток, які відображають середні вартості акцій шістнадцяти провідних компаній, що спеціалізуються на розробці вказаних САПР. Згідно з цим показником, лідирують фірми *Cadence Design Systems*, *Epic Design Technology*, *IKOS Systems*, *Mentor Graphics*, *Synopsys*, *VIEWlogic*, *OrCAD*, *Compass Design Automation*, *ZyCAD* та інші, що постійно вдосконалюють свої системи, вкладаючи велику частку прибутку в науково-дослідну роботу та маркетинг (приблизно в такій пропорції: чистий прибуток – 26 %, НДДКР – 23 %, маркетинг – 51 %).

Фірма *Epic Design Technology* з резиденцією в США та декількома сотнями фахівців початково була самостійним розробником програм для проектування, аналізу та верифікації інтегральних схем, становлячи особливий інтерес своїми програмними інструментами логічної верифікації на всіх рівнях абстракції. Потім вона з'єдналася з компанією *Synopsys*.

Фірма *IKOS Systems* має резиденцію в США та розгалужену мережу дистриб'юторів в Європі та Азії. Вона створює програмні системи, що призначені для високопродуктивного моделювання проектів, описаних засобами мов *Verilog* та *VHDL*, та дозволяють виконувати змішано-рівневе моделювання та верифікацію в процесі розробки інтегральних схем.

Компанія *Synopsys* є лідером у технології синтезу – фундаменті методології високорівневого проектування. Вона просу-

ває системи високорівневого цифрового моделювання (*Verilog*, *VHDL*) та логічного синтезу інтегральних схем.

Компанія *VIEWlogic Systems* розробляє програмне забезпечення на платформах *UNIX* та *MS Windows* для проектування широкого асортименту електронної апаратури, пропонуючи користувачам псевдоповний маршрут проектування (за вибірковими розділами).

Компанія *OrCAD* створює *EDA*-додатки під *OS Windows* для проектування програмованих логічних матриць (*FPGA*), комплексних програмованих логічних пристроїв (*CPLD*) та печатних плат.

Фірма *Compass Design Automation* із центральними офісами в США та у Франції має близько сотні патентів на свої продукти та технології, що належать до класу бібліотек та програмних інструментів розробки напівзамовлених (*ASIC*, *ASSP*) та замовлених інтегральних схем, пропонуючи повний та жорсткий маршрут проектування інтегральних схем глибоко субмікронної технології.

В цілому для ринку САПР електротехніки та електроніки характерна швидка динаміка та націленість великих фірм-розробників на забезпечення повного циклу проектування для кожного типу електронних пристроїв.

Коротко розглянемо декілька найпопулярніших класичних САПР електротехніки та електроніки – *OrCAD*, *P-CAD* та *ACCEL EDA*, *Electronics Workbench*.

**Охарактеризуємо систему *OrCAD*, призначену для проектування електричних схем та електронних пристроїв.** Пакет *OrCAD* фірми *OrCAD Incorporated* – це інтегрований комплекс програм для розробки електричних схем та елект-



ронних пристроїв [123]. Він постачається разом із захистом у вигляді електронного ключа та складається з декількох незалежних модулів, сумісних на рівні початкових файлів.

**Модуль *Schematic Design Tools (SDT)*** призначено для складання електричних та електронних схем проєктованого пристрою та виведення їх на принтер або плотер, а отриманий файл є початковим для обробки іншими модулями пакету. Цей модуль містить бібліотеку елементів та комплектуючих, яку можуть поповнювати фірма-виробник та користувачі: до неї заносять, окрім умовного позначення та найменування, функціональні параметри цифрових мікросхем (відповідність стану входів зі станом виходів, часові затримки тощо).

**Модуль *Digital Simulation Tools (DST)*** призначено для моделювання роботи цифрового пристрою на основі файлу зі схемою, складеною за допомогою модуля *SDT*. Використовуючи бібліотеки функціональних параметрів мікросхем та враховуючи часові затримки сигналів, модуль *DST* моделює роботу пристрою та креслить діаграми напруг у контрольних точках. Аналізуючи їх, конструктор може судити про працездатність схеми.

**Модуль *PC Board Layout Tools (PCB)*** призначено для розведення та трасування печатних плат на основі файлу зі схемою, яку було складено за допомогою модуля *SDT*. Розробник задає розміри печатної плати, кількість шарів та компоновку деталей на платі. Модуль *PCB* оптимізує розведення печатних провідників, відрисовуючи їх на різних шарах різними кольорами. Після цього конструктор може скорегувати розташування комплектуючих, наприклад, для зменшення кількості навісних перемичок.

*Модуль PCB* може виводити дані в стандарті, придатному до використання в ЧПУ-обладнанні для виготовлення печатних плат (свердління отворів, виготовлення фотошаблонів тощо).

*Модуль Programmable Logical Design Tools (PLD)* призначено для проектування пристроїв на базі програмованих логічних інтегральних схем (ПІІС або *PLD*) – мікросхем, які являють собою матриці тригерів та логічних елементів, з'єднаних провідниками (перепалюючи непотрібні провідники, як це робиться при програмуванні ПЗП, можна отримати функціональний вузол, який в разі дискретної технології складався б із декількох корпусів мікросхем).

*Охарактеризуємо систему для розробки електронних пристроїв P-CAD для DOS та її версію ACCEL EDA для Windows.*

Будь-який фахівець у галузі електроніки знає про систему проектування печатних плат *P-CAD*. Її було розроблено компанією *Personal CAD Systems*, впроваджено наприкінці 1980-х років на всіх підприємствах електронної промисловості, що мали відношення до печатних плат, а в 1995 році – придбано та розвинено компанією *ACCEL Technologies*.

Програмний комплекс *P-CAD* фірми *ACCEL Technologies* призначено для розробки електронних пристроїв, починаючи від складання їх принципових схем та завершуючи підготуванням документації на них [122, 153]. Він містить: редактори принципових схем та багат шарових печатних плат; програми автоматичного розміщення компонентів на печатних платах, трасування з'єднань, видачі креслень на пристрій виведення (принтер, плотер, фотоплотер), передавання даних на сверд-

лильні станки з ЧПУ; допоміжні програми обслуговування та складання звітів.

Цією ж фірмою випущено пакет *ACCEL EDA – Windows*-версію програми *P-CAD*, яка поєднує в собі можливості ООП-версії *P-CAD* (дозволяє працювати з попередньо підготованими стандартними компонентами) та наочний інтерфейс *Windows* (втомлюючи послідовність вибору пунктів у великих багаторівневих меню замінено натисканням однієї кнопки з відповідною піктограмою або одиничної клавіші).

Вдосконалені ООП-можливості дозволяють модифікувати об'єкти навіть на завершальних стадіях роботи (раніше доводилося повертатися до етапу побудови об'єктів, а потім вручну наново створювати їх образи). *ACCEL EDA* дозволяє швидко переносити виправлення, що були внесені до підготованої печатної плати, до принципової схеми, та навпаки.

Система *P-CAD* призначена переважно для розробки печатних плат цифрових пристроїв. Додаткові можливості *ACCEL EDA* враховують особливості аналогових та змішаних аналогово-цифрових пристроїв.

У системі *ACCEL EDA* інтегровані найкращі властивості пакетів *TangoPRO* (редактор схем, менеджер бібліотек) та *P-CAD* (засоби розробки печатних плат).

Коротко оглянемо можливості пакету *ACCEL EDA*:

– дозволяє виконувати повний цикл проектування печатних плат, який містить графічне введення схеми, упакування схеми на печатну плату, ручне розміщення компонентів, трасування провідників (ручне, інтерактивне та автоматичне), контроль помилок у схемі та печатній платі, випуск документації;

– підтримує механізм *Engineering Change Order (ECO)* для перенесення змін у печатній платі на схему та навпаки;

– підтримує англійську та метричну системи одиниць, а завдяки застосуванню 32-розрядної арифметики забезпечує дискретність вимірювання лінійних та кутових розмірів та можливість зміни системи одиниць на будь-якій стадії проекту без втрати точності;

– підтримує загальноприйняті в *CAD*-додатках текстові формати *DXF* та *PDIF* опису баз даних, що дозволяє легко обмінюватись інформацією з *AutoCAD*, *OrCAD*, *Viewlogic*, попередніми версіями *P-CAD* та іншими розповсюдженими пакетами;

– дозволяє використовувати на схемах українські та російські літери.

Комплекс *ACCEL EDA (P-CAD for Windows)* містить такі основні функціональні модулі:

– модуль *ACCEL EDA Library Manager* дозволяє готувати інтегровані бібліотеки, що містять інформацію про типові елементи, з яких може складатися схема (до цих бібліотек можна заносити графічні символи елементів, можливі варіанти корпусів, додаткову текстову інформацію);

– модулі *ACCEL Schematic* та *ACCEL P-CAD PCB* – це графічні редактори принципів схем та печатних плат, які за своїми можливостями аналогічні відповідним компонентам *OrCAD* та *P-CAD*, проте мають привабливіший дружній інтерфейс у стилі *Windows*-додатків;

– модулі *SPECCTRA*, *ACCEL Autorouters (Quick Route, PRO Route)* здійснюють трасування печатних плат як в автоматичному, так і в інтерактивному режимі;

– набір утиліт призначений для обміну з іншими додатками, генерації звітів, підготування завдань для станків із ЧПУ та фотоплотерів.

Пакет *P-CAD* працює в середовищі *DOS, Windows/NT/XP* та вимагає не менше 8 *Mb* оперативної пам'яті, а в разі використання утиліт *ACCEL PRO Route* та *SPECCTRA* – не менше 16 *Mb*.

**Охарактеризуємо програмний комплекс *Electronics Workbench* для розробки електричних ланцюгів та радіоелектронного обладнання.** В процесі розробки радіоелектронних пристроїв неможливо обійтися без комп'ютерних технологій у зв'язку зі складністю та великим обсягом робіт, вимогами високої точності та глибокого аналізу. В цьому плані система *Electronics Workbench* надає такі можливості:

– галузями її застосування є підприємства, що виконують розробку електричних ланцюгів, а також вищі навчальні заклади, що займаються вивченням та розробкою радіоелектронних пристроїв (*Electronics Workbench* застосовують у більшості вищих навчальних закладів світу);

– її застосовують як заміну дорогому обладнанню;

– може виконувати велику кількість аналізів радіоелектронних пристроїв, які в разі стандартних методів розробки потребують багато часу;

– містить велику кількість моделей радіоелектронних пристроїв від найвідоміших виробників (*Motorola* та інші);

– може працювати з великою кількістю комп'ютерної периферії та імітувати її роботу;

– не має аналогів по кількості виконуваних функцій та простоті інтерфейсу (є дуже простою в роботі, не вимагає

глибоких знань в комп'ютерній техніці, а її інтерфейс можна опанувати за декілька годин).

*Electronics Workbench* являє собою сучасний засіб програмної розробки та імітації електричних ланцюгів, вивчення роботи принципів схем радіоелектронних пристроїв.

**Геоінформаційні системи (ГИС, GIS).** Важливість цих систем є безсумнівною: переважна більшість інформації, з якою стикається людина в оточуючому світі, має територіальну прив'язку, описуючи ті об'єкти, для яких важливу роль відіграють їх просторове положення, форма та взаємне розташування.

*GIS* тісно пов'язані з поняттям географічної карти – складного та насиченого багатим змістом креслення, що є джерелом інформації про земну поверхню та важливим інструментом досліджень.

Користувачами картографічних матеріалів (географічних карт та знімків, результатів накопичення, систематизації та аналізу відповідних фактичних даних) є: вчені з тих галузей науки та практики, які вивчають біосферу, літосферу, атмосферу тощо (геологія, геофізика, екологія, медичинська географія, соціологія, історія, економіка тощо); державні діячі та службовці; бізнесмени, менеджери, будівельники, агрономи, лісоводи, геологи, синоптики; мореплавці, льотчики, космонавти, автомобілісти, військові тощо.

Нинішній період розвитку картографії пов'язаний з впровадженням в обробку географічних даних обчислювальної техніки та з досягненнями в галузі космічних технологій.

Геоінформаційна система — це програмно-апаратний комплекс для збирання, зберігання, управління, аналізу та відобра-

ження просторово-розподіленої інформації (інформаційна система з географічно організованою інформацією).

**САПР промислових підприємств.** В останнє десятиріччя в країнах СНД впроваджують широко розповсюджені зарубіжні системи автоматизованого повномасштабного 3D-проекування промислових підприємств від таких фірм-розробників, як *CADCentre*, *Intergraph*, *Matra*, *Data Vision*, *Rebis* (наприклад, система *OMNI-Series*) та інших.

Ці системи дозволяють створювати 3D-моделі трубопроводів, обладнання, будівельних конструкцій, опалювальної та вентиляційної систем, розводки електричних кабелів тощо, а також автоматично генерувати потрібні монтажні креслення та проектну документацію.

**Службові програмні комплекси САПР.** Це достатньо широкий спектр ПЗ.

З одного боку, це не критичні стосовно функціонування САПР в цілому, стадартні програми офісного призначення.

З іншого боку, це життєво важливі для будь-якої САПР компоненти, що належать до наведених нижче двох категорій.

По-перше, це програми векторизації (перетворення растрового формату в векторний), що дозволяють швидко та недорого перетворювати архіви в модулі, придатні для повторного використання (наприклад, популярна в Європі система *TsokTrix*, яку експлуатують *Volvo* та *Soab*, визнана в США продукція компанії *Arbor Image*).

По-друге, це системи управління конструкторською документацією та проектами. У великих САПР для цього існують відповідні модулі. В середніх САПР цю проблему вирішують засобами сторонніх виробників.

Наприклад, *SolidWorks* дозволяє здійснювати імпорт сторонньої моделі в ту поточну, з якою працює конструктор, за допомогою технології *Drag-and-Drop* (з потрібної *HTML*-сторінки, відображеної стандартним браузером).

По-третє, це СУБД, що орієнтовані на зберігання специфічних для САПР даних. Тобто – даних дуже великого діапазону, обсягів та високої складності опису (відповідно, запитів до них), що робить неефективним застосування реляційних СУБД та виводить на рівень перспективних системи типу *Adabas* фірми *Softwase AG* та об'єктно-орієнтовані СУБД типу *Poet* фірми *Poet Software*.

**Універсальне відкрите інтегроване середовище проектування *CADdy* західно-германської фірми *ZIEGLER-Informatics*.** Ця система охоплює широке коло галузей практичного застосування та розв'язуваних задач. Її модулями загального призначення є: базовий пакет; електронний архів креслень; база даних виробів; модуль перегляду та друку креслень; графічна інформаційна система; графічна система управління базами даних; компілятор мови програмування *C* та бібліотека відповідних функцій; інтерфейс із графічними функціями, тривимірне світлотіньове розфарбовування.

*CADdy* підтримує такі задачі:

– у галузі машинобудування – двовимірне проектування, БД кріпильних елементів, параметризація, кінематика, твердотільне 3D-моделювання, вигинка листового металу, розгортка трубних з'єднань, розкрій та розміщення деталей, технічні ілюстрації;



– у галузі промислових установок – розробка принципів схем, розводка трубопроводів в ізометрії, БД обладнання та арматури;

– у галузі електротехніки – принципові схеми та схеми підключень, БД виробів, компоновка обладнання, архів проєктів;

– у галузі електроніки – принципова та монтажна схеми, проєктування печатних плат, авторозміщення, автотрасування;

– у галузі геодезії – обрахунок та перетворення даних вимірювань, урівнювання геодезичних мереж, топографія та картографія, цифрова модель рельєфу та розрахунок об'ємів, векторизація, архів перерізів карт (гарні можливості для створення геодезичної основи проєктування);

– у галузі інженерних мереж – проєктування водопровідних та каналізаційних мереж, експлуатація каналізації, проєктування газових мереж, проєктування та розмітка доріг, планування земельних робіт;

– у галузі архітектури – 2D/3D-проєктування, розрахунок площ помешкань, проєктування та розрахунок конструкцій покрівель, статичні будівельні розрахунки, армування конструкцій, БД профілів, 3D-проєктування металоконструкцій, планування території;

– у галузі обладнання будівель – системи опалення, водопостачання, каналізації, кондиціонування та вентиляції, електропостачання, слабострумові системи.

*CADdy* дозволяє ефективно здійснювати проєктування та моделювання охоронно-пожежних систем.

За допомогою *CADdy* успішно розв'язують задачі обладнання "інтелектуальної будівлі", комплексного проєктування та

встановлювання. Наприклад, для фірм, які пропонують на вітчизняному ринку мережні технології, *CADdy* надає всі можливості для проектування та встановлювання корпоративних інформаційних систем на принципах побудови "інтелектуальної будівлі", включаючи локальні обчислювальні мережі на базі обладнання *Cabletron Systems* та *Cisco*, телефонні системи з встановленням офісних АТС, *AT&T* та системи гарантованого безперебійного енергопостачання.

**Універсальна САІР *Visio* фірми *Shapeware*.** Є одним із найуніверсальніших, найпотужніших та простих у використанні прикладних інструментальних засобів для побудови: технічних креслень, мережних топологій, проектів житлових та офісних приміщень, гістограм, графіків, бізнес-діаграм та маркетингових діаграм (із підтримкою *UML* – уніфікованої мови моделювання *Unified Modeling Language*), структурних схем, топографічних карт, рисунків та багатьох інших можливостей. САІР *Visio* є гарним інструментом для підготування як простих діаграм, так і складних креслень.

**Комплекс програм електронного документообігу *RxEDM*.** Ця система дозволяє розв'язувати критичні задачі в стислі строки.

У муніципальному господарстві *RxEDM*, наповнену конкретним змістом, зручно використовувати за довідкову систему по тим помешканням, які розташовано на території адміністративної територіальної одиниці (району, округу, області тощо), для: швидкого знаходження оперативної інформації про стан фонду нежилих помешкань, надання замовнику потрібних йому для перегляду зображень приміщень, організації перевірок "суперечливих" приміщень тощо.

Застосування *RxEDM* виключає можливість зловживань та неплатежів орендної плати за використання приміщень, а тому приводить до поповнення міського бюджету.

Надана *RxEDM* текстово-графічна БД має безперечні переваги в порівнянні з традиційною текстовою: подана інформація є наочнішою; можна легко користуватися топологічною інформацією (встановлювати суміжність помешкань, створювати або ліквідувати пройоми тощо), що обумовлює розмір орендованої площі та можливість пошуку додаткових орендарів.

У галузі управління будівництвом *RxEDM* дозволяє покращувати взаємодію між субпідрядниками завдяки: скороченню втрат часу на транспортування та копіювання креслень, контроль процесу обміну кресленнями; мінімізації кількості помилок, пов'язаних із наявністю декількох версій та копій креслень.

У промисловості *RxEDM* дозволяє ефективно вирішувати проблему невідповідності виведеного на друк креслення тому кресленню, що залишилося в розробника за архівне (невідповідність виникає через існуючу практику внесення оперативних змін у креслення виробів, які вже виготовляють або запускають у виробництво, поспіхом).

**Універсальне середовище автоматизованого проектування *Microstation* компанії *Bentley Systems*.** Ця система надає чудові можливості *CAD*-технології для професійної роботи в різних галузях: твердотільне параметричне моделювання в машинобудуванні (*Microstation Modeler*) та додатки для машинобудівного проектування в середовищі *Microstation*, а саме *COSMOS/Designer* для інженерно-технічних розрахунків

та структурного аналізу з застосуванням МКЕ (методу кінцевих елементів), *ESPRIT/MS* для підготовки програм для 2-5-координатних ЧПУ-станків, *ADAMS/MS* для кінематичних розрахунків; архітектура, будівництво та ландшафтне проектування; проектування промислових установок та підприємств; поєднання САПР та ГІС із потужністю *Microstation Geographics*; обробка растрових зображень за допомогою *Microstation Descartes*.

*Microstation* – це мультиплатформова система (наприклад, її підтримують *PC*, *Mac* та *UNIX*-станції).

*Microstation* – це програма 3D-моделювання об'єктів (хоча для плоского креслення вона теж є достатньо ефективною), основними напрямками використання якої є механічні додатки (конструювання, розрахунки, виробництво), архітектура, дизайн (для інтер'єрів, індустріальний тощо), геоінформаційні системи та багато іншого.

З іншого боку, *Microstation* – це ядро комплексу програм-надбудов із аналогічною назвою, наприклад, *Modeler* призначено для твердотільного моделювання, *Descartes* – для роботи з растровими об'єктами, *Masterpiece* – для високоякісної візуалізації.

Є полегшена версія ядра *Microstation* – *PowerDraft* (для роботи з 2D та частково з 3D, без засобів роботи з вільною поверхнею).

*Microstation* – це система для професійних проектувальників будь-яких спеціальностей, яку орієнтовано на найвищі інтелектуальний та професійний рівень користувачів та їх вимоги. Сімейство *Microstation* націлене на надання користу-

вачам усього арсеналу засобів сучасного автоматизованого проектування.

*Microstation*, як універсальна базова графічна система, має незрівняні за широтою, повнотою та рівнем реалізації функціональні можливості, основними з яких є: засоби роботи з векторною 2D-графікою; можливості 3D-моделювання; засоби підготування вихідних креслень; бібліотеки типових графічних фрагментів; засоби роботи з растровою графікою; інтерфейс з БД; можливості візуалізації та анімації.

### Питання для самоперевірки

1. Наведіть класифікацію інженерних САПР.
2. Які функції виконують САПР машинобудівного профілю?
3. Сформулюйте переваги архітектурно-будівельних САПР.
4. Охарактеризуйте функціональні можливості САПР електротехніки та електроніки.
5. У чому полягає практичне призначення ГІС ?
6. Які можливості надають САПР промислових підприємств?
7. Охарактеризуйте роль інших класів САПР.

## 12.4. Комплексне застосування САПР

Багато вдалих проєктів у різних галузях промисловості базуються на ПЗ фірми *Autodesk*. Часто обирають систему *AutoCAD* – завдяки її широкому розповсюдженню та великій кількості створених для неї спеціалізованих прикладних програм (для успішної роботи однієї системи *AutoCAD* недостатньо, оскільки ефективне застосування засобів автоматизації базується на використанні прикладного, проблемно-орієнтованого ПЗ, що працює під управлінням *AutoCAD*).

Розглянемо наочні приклади комплексних типових застосувань САПР із використанням продукції *Autodesk* та інших провідних фірм.

**Управління національними лісами та ресурсами.** Початково агентство *US Forest Service* Департаменту сільськогосподарства США (біля 200 млн. акрів лісів та лугів, півмільйону миль доріг та інші ресурси) використовувало трудомістке та малопродуктивне ручне проектування та геодезичну зйомку, якій перешкоджала велика кількість лісових доріг. Потрібна була ефективна автоматизована система проектування з можливістю інтеграції БД та електронних карт.

На даний час фахівці з управління земельними ресурсами, архітектори, цивільні інженери, картографи агентства використовують комп'ютерні технології, що значно полегшують їх працю. Застосоване ПЗ фірм *AutoDesk* (*AutoCAD*, *3D Studio*, *Animator Pro*, *Auto Vision*) та *SoftDesk* (*SoftDesk Modules*). Розширення *ADE* пакету *AutoCAD* дозволяє легко здійснювати пошук, редагування, зв'язування великої кількості

рисунків (креслень), виготовлених за допомогою *AutoCAD*, *3D Studio*, *Animator Pro* та *Auto Vision* допомагають підготувати презентації та наочні посібники для навчальних семінарів.

**Відображення стану та проектування геотермальних джерел.** Енергетична компанія *Northern California Power Agency (NCPA)* виробляє 90 % енергії шляхом переробки природних ресурсів (наприклад, енергії гейзерів, розташованих на території 3800 акрів), постачаючи енергією 12 міст.

З переходом *NCPA* до автоматизованого проектування, з використанням ПЗ фірм *AutoDesk (AutoCAD, ADE)*, *Eclipse (Facade)*, *Lettering Systems (Letterease)* та *Schreiber Instruments (Quick Surf)*, усі роботи завершують швидше (наприклад, проекти – за 1 день замість тижня) та з вищою якістю.

За допомогою *ADE* здійснюють інтеграцію проектів із картографічною інформацією для створення карт спеціального призначення.

**Індивідуальне та комерційне проектування інтер'єрів та розташовування обладнання.** Фірма *RGP (Pallas)*, одна з найкращих дизайнерських компаній США, почала автоматизацію для виживання в умовах сильної конкуренції, а вже через п'ять років обладнала площу в 3,2 млн. кв. футів.

Конструктори та дизайнери цієї фірми застосовують для проектування інтер'єрів та планування залів магазинів, супермаркетів, торгівельних центрів ПЗ фірм *AutoDesk (AutoCAD, 3D Studio, Auto Vision)*, *Informative (Graphics Myriad)*, *IST Nova Fill*, *Softdesk ASG (ASG Architectural)*, *Intel (Proshave)*.

Завдяки цьому проекти надаються замовникам вдвічі швидше, на вищому презентаційному рівні та зі зниженням вартості проектування.

*AutoCAD* використовують для 2D/3D-проекування інтер'єрів, *3D Studio* та *AutoVision* – для підготування презентацій, створення фотореалістичних зображень, проведення віртуальних екскурсій по помешканнях, які проектують тощо.

**Реставрація монументу "Статуя Свободи", перепроекування музею та оточуючого простору.** Компанія *National Park Service* відповідає за стан: статуї Свободи та пов'язаного з нею музею (*Statue of Liberty, Ellis Island National Monument*), які щорічно відвідує більше мільйону осіб; службових та адміністративних споруд загальною площею більше 50 акрів.

Свого часу для прискорення реставраційних робіт із перепроекування головної споруди (перетворення її в триповерховий музей площею 250 тис. кв. футів), було здійснено перехід до автоматизованих методів із застосуванням системи *AutoCAD* фірми *Auto Desk*. Це підвищило ефективність роботи на 70 %.

**Перепроекування та розвиток аеропорту Лас-Вегасу.** Інженерна та конструкторська групи фірми *Clarc Country Department of Aviation* використала продукти фірм *AutoDesk (AutoCAD)* та *Softdesk (Softdesk Civil Survey)* для архітектурного проектування та цивільного будівництва на 33 тис. акрів площі аеропорту Лас-Вегаса, що є одним із найзавантаженіших аеропортів на заході США, в процесі виконання всіх робіт: від реконструювання терміналів (моделювання терміналів, включаючи розрахунки несучих конструкцій, електрики, трубопроводів тощо) до будівництва шляхів.



**Презентація розроблюваних архітектурних проєктів.** Фірма *MHTN Architects* досягла успіхів у проєктуванні, моделюванні та реалістичному відображенні архітектурних проєктів за допомогою 3D-моделювання та анімації, що надають діячам архітектури та мистецтва засоби реалістичного подання їх витворів, на базі ПЗ фірм *AutoDesk (AutoCAD, 3D Studio)*, *Adobe Systems (Adobe Premiere)*, *Robert Meneel Associates (Accrender)*.

Спочатку *MHTN* застосовувала в проєктуванні *VersaCAD*.

Надалі стали потрібними твердотільне моделювання та анімація. Для їх реалізації *MHTN* спробувала використовувати *Microstation*, але знайшла її не найкращим рішенням, оскільки архітектори та дизайнери *MHTN* часто працюють разом із архітект-інженерами та консультантами з інших фірм, які використовують в основному *AutoCAD*.

Не менших успіхів досягла фірма *Gaviota Graphics*, яка займається: проєктуванням зовнішнього вигляду інтер'єрів будинків; підбиранням матеріалів; розробкою реалістичних зображень, які дозволяють клієнту переглядати споруду при сході та заході сонця та здійснювати "прогулянки" всередині та ззовні цієї споруди при різному освітленні.

*Gaviota Graphics* спеціалізується на створенні фотореалістичних зображень архітектурних конструкцій для їх аналізу та презентації.

Автоматизація робіт на базі систем *AutoCAD* та *3D Studio* фірми *Autodesk*, *3DSurf* та *QuickSurf* фірми *Schreiber Instruments* суттєво підвищила їх ефективність та знизила їх вартість (без автоматизації діяльність фірми *Gaviota Graphics* була неефективною, а її продукція надто дорого коштувала).

Часто систему *AutoCAD* фірми *Autodesk* використовують у процесі виконання індивідуальних архітектурних проєктів.

**Проектування механічних виробів.** Фірма *Planar America* проєктує та виготовляє пласкі екрани відображення інформації для медичних та промислових потреб.

Оскільки ручне проектування цих виробів було неефективним та вимагало багато часу, було впроваджене автоматизоване проектування на базі ПЗ фірми *AutoDesk* (*AutoCAD*, *AutoCAD Designer*). Бажаної ефективності було досягнуто шляхом створення наскрізної системи, що автоматизує технологічний процес від проектування до виготовлення.

Розробка та модифікація креслень на базі *AutoCAD* забезпечила підвищення продуктивності, а параметризація креслень на базі *AutoCAD Designer* спростила процес проектування.

*AutoCAD* використовують для конструювання компонентів пласких екранів, спеціальні програми на базі мови *AutoLisp* – при конструюванні фотомасок, *AutoCAD Designer* – для детального конструювання та моделювання.

**Механічне конструювання, моделювання та анімація в нафтовій та газовій промисловості.** Фірма *Hydril Company* та *Hydril Controls* використовують ПЗ фірми *AutoDesk* (*AutoCAD*, *3D Studio*, *Animator Pro* та спеціальне ПЗ, розроблене на базі *AutoCAD*) для підвищення ефективності проектування.

*AutoCAD*, який було обрано завдяки простоті та популярності, використовують для 2D-конструювання та 3D-твердотільного моделювання, *3D Studio* та *Animator Pro* – для маркетингу та при поданні документації.

**Конструювання та виробництво велосипедів та мотоциклів.** Фірма *Bushido Racing* є лідером у проектуванні та

виробництві велосипедів та їх компонентів, проектує та виготовляє унікальні картінги та шасі для мотоциклів.

Для підвищення ефективності діяльності, *Bushido Racing* використовує ПЗ фірми *AutoDesk* (*AutoCAD*, *AutoSurf*, *AutoCAD Designer*, *3D Studio*, *AutoVision*, спеціальне ПЗ). Завдяки цьому типовий цикл проектування скоротився більше, ніж втричі, а обсяг робочих файлів – майже на порядок.

Автоматизація почалася з впровадження автоматизованого випуску креслень, де *AutoCAD* отримав перевагу над системами *CAD Key Computervision* завдяки широкому розповсюдженню та простоті використання. Далі було впроваджене автоматизоване проектування 3D-виробів.

У процесі проектування картінгів використовується ПЗ *AutoCAD* із внутрішніми стереографічними та 3D-можливостями, а кожен елемент прототипу двигуна будують за допомогою програм твердотілого моделювання *AutoCAD*.

**Навчальний курс графічних програм для студентів.** Відділення технічної графіки *Purdue University* розробило та впровадило унікальну навчальну програму з комп'ютерної графіки для студентів, які спеціалізуються в різних галузях науки та техніки (графічні аніматори, комерційні художники, фахівці в галузі САПР та технічної графіки).

Початково курси з дисциплін циклу САПР включали *CAD Key*, *VersaCAD* та *MathCAD*. На даний час студенти в процесі роботи з комп'ютерними ілюстраційними зображеннями та інженерною графікою, при підготованні науково-технічної документації та розробці мультимедійних проектів використовують весь набір програм фірми *Autodesk*.

## Питання та завдання для самоперевірки

1. Наведіть приклади великомасштабних проектів, в яких було реалізоване комплексне використання систем інженерної комп'ютерної графіки.

2. Чому продукція фірми *Autodesk* має найбільш високий попит при розробці великомасштабних проектів?

3. Проаналізуйте, які системи наукової та інженерної комп'ютерної графіки та для розв'язування яких задач використовуються на практиці на підприємствах, в установах та організаціях за вашим місцем проживання, навчання або роботи. Висуньте ваші пропозиції щодо вдоконалювання тієї практики, що склалася.

4. За матеріалами комп'ютерної періодики наведіть декілька показових прикладів практичного застосування комп'ютерної графіки в Україні та зарубіжжі.

5. За матеріалами провідних комп'ютерних журналів проаналізуйте останні досягнення в галузі практичних застосувань комп'ютерної графіки та дизайну за минуле та поточне півріччя (нові технології, програмні й апаратні системи та сфери їх впровадження).

## Навчальний практикум до гл.12

**1-а група завдань.** Використовуючи інструментальні засоби систем наукової комп'ютерної графіки, виконайте наведені нижче тренувальні вправи на застосування цих систем, із графічною візуалізацією процесу та результатів цього виконання (конкретні функції та значення параметрів до цих вправ вам призначить викладач).

1. Визначити, обчислити для заданих значень аргументу та візуалізувати функції користувача (наприклад, інтегральні функції, комплексні функції, сходишкові функції дійсного аргументу, функцію знаходження власного значення квадратної матриці, функції алгебри логіки).

2. Вивести емпіричну формулу (наприклад, встановити функціональний зв'язок між двома наборами емпіричних числових даних за допомогою методу найменших квадратів).

3. Обчислити суму числового ряду з заданою точністю та для заданого переліку значень параметру табуляції.

4. Обчислити інтеграл (наприклад, криволінійний, поверхневий, потрійний).

5. Виконати табуляцію інтегралів із заданою точністю (наприклад, невластних або еліптичних інтегралів).

6. Виконати диференціювання певного класу функцій (наприклад, диференціювання неявної функції або диференціювання інтегралу по параметру).

7. Здійснити чисельний розв'язок диференційного рівняння.

8. Виконати графічне диференціювання та інтегрування функції (наприклад, поліному  $n$ -го ступеня) на заданому інтервалі.

9. Здійснити заміну змінних (наприклад, у диференційних виразах, у подвійних інтегралах).

10. Розв'язати рівняння (наприклад, рівняння виду  $f(x) = 0$ , де  $f(x)$  є поліномом  $n$ -го ступеня, або лінійне однорідне/неоднорідне диференційне рівняння  $n$ -го порядку з постійними коефіцієнтами, що приводиться до характеристичного поліному  $n$ -го ступеня), враховуючи початкові наближення коренів цього рівняння та визначену точність їх обчислення.

11. Розв'язати систему лінійних рівнянь в матричному вигляді з контролем сингулярності матриці коефіцієнтів.

12. Розв'язати систему векторних рівнянь.

13. Розв'язати систему порівнянь за заданим модулем.

14. Розв'язати за допомогою методу розв'язувальних блоків: задачі лінійного/нелінійного програмування, пошуку розв'язків систем функціональних нерівностей або систем параметричних нелінійних рівнянь/нерівностей; цілочисельні задачі (пошуку розв'язків у цілих числах систем лінійних рівнянь, погано визначених систем лінійних рівнянь, систем нелінійних рівнянь в комплексній області).

15. Обчислити екстремум функції багатьох змінних.

16. Розв'язати варіаційну задачу (наприклад, здійснити наближений пошук екстремуму функціоналу при заданих крайових умовах наближеним методом Рітца).

17. Здійснити чисельний гармонійний аналіз та тригонометричну інтерполяцію функції.

18. Розв'язати фізичну задачу (наприклад, розрахувати баланс енергії при змінюванні поверхні рідини внаслідок злиття її

крапель, обчислити так званий еквівалентний рівень звуку, скласти погодинний графік змінювання рівня ґрунтових вод у заданих точках системи водопостачання комплексу свердловин, обчислити параметри процесу деформування ґрунту внаслідок його ущільнення, обчислити параметри процесів теплообміну та руху газів у топці та газоходах).

**2-а група завдань.** Використовуючи інструментальні засоби систем інженерної комп'ютерної графіки, потрібно виконати такі тренувальні вправи.

19. Необхідно розробити на базі графічного пакету, що належить до класу систем інженерної комп'ютерної графіки (наприклад пакету *AutoCAD* та вбудованої в нього мови програмування *AutoLisp* фірми *Autodesk*) один із базових модулів комплексного програмного засобу "АРМ інженера-конструктора", адаптованого для інженерів-машинобудівників, який здійснює налаштування цього графічного пакету на специфіку діючих стандартів та передбачає введення до ЕОМ параметричних програм рисування типових елементів креслень. Нижче наведено перелік функцій, підтримку яких потрібно реалізувати в рамках АРМ конструктора-машинобудівника.

Ф 1. Вибір з меню та автоматизоване креслення стандартних форматів  $A1$ ,  $A2$ ,  $A3$ ,  $A4$  із наступними типами штампів:  
а) високим; б) низьким;

Ф 2. Автоматизоване креслення позначень: а) шорсткості обробки; б) покриття швів зварних з'єднань; в) паяних з'єднань; г) клеєних з'єднань; д) стандартних клепаних з'єднань.

Ф 3. Автоматизоване креслення позначень допусків:  
а) прямолінійності; б) площинності; в) округлості; г) циліндричності, д) профіля продольного перерізу; е) паралельності;

ж) перпендикулярності; з) нахилу; і) співосності; к) симетричності; л) позиційного перетину осей; м) радіального биття; н) торцевого биття; о) биття в заданому напрямку; п) повного торцевого биття; р) форми профілю; с) форми поверхні.

Ф 4. Автоматизоване креслення типових отворів на кресленнях: а) з фаскою; б) з двома фасками; в) з різьбою; г) з цековкою; д) з цековкою та фаскою; е) з цековкою та різьбою; ж) з конусом.

Ф 5. Автоматизоване креслення стандартних кріпильних деталей: а) болтів (з шестигранною голівкою; з шестигранною зменшеною голівкою; з шестигранною зменшеною голівкою, полем допуску  $h_9$  та короткою нарізною частиною; з шестигранною голівкою для шарнірних з'єднань; зі шліцем у шестигранній голівці; тих, які не випадають та мають шліц у шестигранній голівці); б) гвинтів (тих, які не випадають та мають напівкруглу голівку; з циліндричною голівкою; з контрувальним отвором у циліндричній голівці; з потайною голівкою; з потайною голівкою, полем допуску  $h_9$  та короткою нарізною частиною; з потайною голівкою та хрестоподібним шліцем; з напівкруглою голівкою; з напівкруглою голівкою, полем допуску  $h_9$  та короткою нарізною частиною; з плоскоопуклою голівкою; з плоскоопуклою голівкою та хрестоподібним шліцем; для люків; з напівпотайною голівкою та кутом 90 градусів; з напівпотайною голівкою та кутом 120 градусів; з циліндричною голівкою та шестигранним заглибленням під ключ; установочних із конічним кінцем; установочних із плоским кінцем; установочних із циліндричним кінцем); в) шпильок; г) гайок (шестигранних; шестигранних низьких; шестигранних проріз-



них; шестигранних прорізних низьких); д) штифтів (циліндричних; конічних); е) шайб; ж) шайб пружинних.

Ф 6. Програми простановки розмірів відповідно діючим стандартам: а) горизонтальних лінійних розмірів; б) вертикальних лінійних розмірів; в) радіусів; г) діаметрів.

Ф 7. Стилзовані шрифти для випуску плакатів.

Ф 8. Набір програм розміщення та редагування тексту на полі креслення для випуску таблиць та технічних вимог: а) програма імпорту/експорту тексту між текстовим файлом та кресленням; б) текстова "лінза" (програма збільшення тексту для перегляду); в) програма редагування тексту на кресленні стандартними текстовими редакторами; г) програма для полегшення випуску текстових плакатів; д) програма розбивки текстових рядків на окремі слова в полі креслення.

Ф 9. Автоматичне креслення оригінальних кріпильних деталей.

Ф 10. Бібліотека параметризованого кріпежу, що містить довідкові таблиці характеристик міцності кріпильних деталей, в яких для різних матеріалів наведено: припустимі навантаження на розрив різьби, припустимі зусилля та моменти затягування різьби, припустимі навантаження на зріз стрижня тощо.

Це завдання складене на базі матеріалів статей в журналі "Компьютер Пресс" пана Б.Ткаченка (м. Дніпропетровськ), присвячених питанням створення АРМ для інженерів-машинобудівників.

20. Використовуючи інструментальні можливості графічного пакету, що належить до класу систем інженерної комп'ютерної графіки (наприклад, САПР *AutoCAD* та мови програмування *AutoLisp*), виконайте 2D-креслення та наочне реаліс-

тичне 3D-моделювання для одного з наведених далі типів конструкцій одношхдинкових редукторів [173]: а) циліндричний горизонтальний двопотоковий з двома швидкохідними валами; б) конічний з вертикальним тихохідним валом; в) циліндричний вертикальний з верхнім розташуванням шестерні; г) черв'ячний з боковим розташуванням черв'яка; д) циліндричний горизонтальний з нахиленим роз'ємом корпусу; е) черв'ячний з вертикальним валом черв'яка; ж) циліндричний вертикальний з нижнім розташуванням шестерні (конструкція 1); з) черв'ячний з верхнім розташуванням черв'яка (конструкція 1); і) циліндричний горизонтальний; к) черв'ячний з верхнім розташуванням черв'яка (конструкція 2); л) циліндричний вертикальний з верхнім розташуванням шестерні; м) черв'ячний з нижнім розташуванням черв'яка (конструкція 1); н) циліндричний з вертикальними валами; о) конічний горизонтальний; п) циліндричний вертикальний з нижнім розташуванням шестерні (конструкція 2); р) черв'ячний з нижнім розташуванням черв'яка (конструкція 2); с) конічний з вертикальним швидкохідним валом; т) циліндричний горизонтальний двопотоковий з двома тихохідними валами.

21. Необхідно здійснити: вибір тих архітектурно-будівельних САПР, які якнайкраще відповідають одному з наведеного нижче переліку критеріїв і параметрів оцінки та вибору програмного забезпечення для автоматизації архітектурного проектування; виконати за допомогою інструментальних засобів однієї з обраних САПР відповідну задачу проектування.

С 1. Орієнтація САПР на специфіку та стадії архітектурного проектування: а) проектування та розташовування

меблів; б) інтер'єри; в) проектування житлових та суспільних будівель; г) реставрація пам'яток архітектури; д) проектування промислових будівель та споруд; е) робоча документація.

С 2. Можливості САПР, які забезпечують моделювання та проектування стін.

С 2.1. Робота з розбивочними осями: а) розбивка осей (лінійних, полярних, дугоподібних); б) маркування осей (автоматичне, напівавтоматичне); в) оброзмірювання осей (автоматичне).

С 2.2. Відрисовування стін: а) різні способи відрисовування стін (з прив'язкою до осі, без прив'язки до осі, з центральною прив'язкою, з прив'язкою по краю стіни); б) відрисовування типових конфігурацій стін; в) відрисовування стін змінної висоти; г) відрисовування криволінійних стін; д) відрисовування похилих стін; е) відрисовування стін складного профілю; ж) відрисовування стін із вертикальними членуваннями; з) відрисовування стін із замкнених поліліній; і) відрисовування стін із заданим обробленням для всіх поверхонь; к) відрисовування підпорних стінок та пандусів.

С 2.3. Обробка стін, які вже відрисовано: а) нішами та пілястрами; б) нішами та пілястрами з арочними завершеннями; в) рустами (облицьовуванням); г) карнизами; д) протягуванням профілю вздовж 2D/3D-маршруту; е) пристінними полицями з брусками.

С 2.4. Бібліотеки стилів стін: а) ведення бібліотеки стилів стін; б) створення легенди тих стін, які використовуються в кресленні; в) створення таблиці всіх стилів стін у бібліотеці; г) видача інформації про стиль обраної стіни; д) маркування та специфікації стін.

**С 2.5.** Виконання штрихування стін: а) штрихування вертикальних площин стін із можливістю редагування; б) штрихування горизонтальних перерезів стін із можливістю редагування.

**С 2.6.** Редагування стін: видалення, сполучення, обробка сполук, обробка перетинів та зрощування стін; заміна стилю стін на той, що було обрано з бібліотеки; обрізання стіни до вказаної точки; подовження стіни до об'єкту та до покрівлі.

**С 3.** Можливості САПР, які забезпечують моделювання проїомів (віконних, дверних тощо).

**С 3.1.** Робота з проїомами та отворами в стінах: а) оброзмірювання по ходу відрисовування; б) відрисовування чвертей (тих, які видно в плані; тих, які видно в розрізі).

**С 3.2.** Робота з дверними проїомами (прямокутними; з арочними завершеннями; в прямолінійних, криволінійних та похилих стінах): а) відрисовування дверних полотниць; б) відрисовування столярки; в) відрисовування наличників; г) бібліотека дверних проїомів (розміри, найменування); д) бібліотека дверних столярок; е) балконні двері з вікнами; ж) маркировка; з) автоматична маркировка при відрисовуванні проїомів; і) відомості по дверях.

**С 3.3.** Робота з віконними проїомами: а) відрисовування козирків; б) відрисовування підвіконь (прямокутні; з арочними завершеннями; в прямолінійних, криволінійних та похилих стінах); в) відрисовування столярки; г) відрисовування наличників; д) стрічкове оскління; е) суцільне оскління; ж) вітражі; з) еркери (прямокутні, криволінійні); і) вікна в покрівлях; к) вікна в кутах будівель; л) бібліотека віконних проїомів (розмір, маркирування); м) бібліотека віконних столярок; н) генератор

створення стільця для бібліотеки; о) маркування (автоматичне маркування при відрисовуванні проїомів); п) відомості по вікнах.

С 3.4. Редагування дверних та віконних проїомів: а) копіювання проїомів через вказаний інтервал; б) копіювання проїомів за прикладом; в) заміна стилю вікна з бібліотеки або за прикладом; г) параметричне редагування вікон; д) видалення проїому; е) пересування проїому вздовж прямої; ж) пересування проїому по дузі.

С 4. Можливості САПР, які забезпечують моделювання та проектування сходів, ліфтів, ескалаторів: а) параметричне відрисовування сходів (одномаршевих, з лінією розриву, без лінії розриву, дугоподібних внутрішніх, дугоподібних зовнішніх, двомаршевих, з розташуванням маршів на одній прямій, без зазору між маршами, з простором між маршами, з розташуванням маршів під кутом 90 градусів один до одного, з довільним кутом між маршами, тримаршевих, багатоплощадочних, спіральних, з напівкруглою площадкою між маршами, сходів по сектору), відрисовування перил, відрисовування балюстради; б) засоби розрахунку параметрів сходів; в) параметричне відрисовування ескалаторів; г) параметричне відрисовування ліфтових вузлів; д) бібліотека блоків ліфтових вузлів.

С 5. Можливості САПР, які забезпечують моделювання та проектування покрівель.

С 5.1. Створення покрівель: а) загальне настроювання відрисовування покрівель; б) односкатна покрівля; в) двоскатна, трискатна, чотирикатна вальмова покрівля; г) двоскатна, трискатна, чотирикатна мансардна покрівля; д) шатрова

покрівля; е) покрівля зі складною конфігурацією; ж) купол; з) пласка покрівля; і) пласка покрівля з парпетом; к) обробка *3DFACE*-граней для побудови складних покрівель (обрізання граней до перетину, подовження граней, сполучення граней, розрив граней на дві частини, відрисовування певної грані між двома вказаними гранями); л) створення площин покрівлі складної конфігурації командою *PFACE*; м) обробка складних *PFACE*-поверхонь (обрізання та подовження поверхонь, приховування граней складних поверхонь); н) штрихування *3DFACE*-поверхонь покрівлі; о) штрихування складних *PFACE*-поверхонь покрівлі; п) типи штрихувань для покрівель; р) додаткові установки, що прискорюють побудову складних покрівель (установка кута перехрестя по нахилу покрівлі тощо); с) розрахунок конструкцій дахів; т) будівельні конструкції (відрисовування, розрахунок, специфікації); у) вікна в покрівлях; ф) слухові вікна в покрівлях; х) постановка кутів нахилу скатів покрівлі.

С 5.2. Редагування покрівель: а) перетворення трискатної покрівлі в двоскатну; б) пристиковування двоскатної покрівлі до вказаного скату покрівлі; в) змінювання кутів нахилу покрівлі.

С 6. Можливості САПР, які забезпечують моделювання та проектування конструктивних елементів (підлога, стеля, балки, колони тощо).

С 6.1. Моделювання та проектування балок: параметричне відрисовування балок із прокатних профілів.

С 6.2. Моделювання та проектування підлоги та стелі: а) відрисовування підлоги/стелі; б) штрихування підлоги/стелі;

в) розкладка панелей перекриття; г) тип підлоги для селектованих приміщень; д) тип підлоги для вказаних помешкань.

С 6.3. Моделювання нульових підлог: план підлог.

С 6.4. Моделювання та проектування колон: а) параметричне відрисовування колон; б) відрисовування одиночної колони; в) автоматичне відрисовування колон у місцях перетину осей; г) відрисовування колон із огороджувальною конструкцією.

С 6.5. Інші вироби та елементи: а) параметричне відрисовування каміну; б) параметричне відрисовування димових труб.

С 7. Можливості САПР, які забезпечують роботу з об'ємною моделлю, поверхами, фасадами та розрізами.

С 7.1. Робота з об'ємною моделлю: а) побудова тіней об'ємної моделі на поточну КСК; б) швидке отримання перспективи; в) засоби корекції отриманої перспективи; г) засоби для отримання 2D-перспективи (аксонометрії) на основі тривимірної; д) засоби для отримання аксонометрії; е) засоби отримання анімаційного фліку.

С 7.2. Робота з поверхами: а) встановлення поверху; б) копіювання поверхів; в) вмикання/вимикання видимості поверхів.

С 7.3. Робота з фасадами та розгортками по стінах: а) швидке отримання 3D-фасаду та 3D-розрізу; б) отримання 2D-фасаду з 3D-фасаду через допоміжний файл; в) отримання 2D-фасаду з 3D-фасаду методом проектування обраних об'єктів; г) вікна для 2D-фасаду (на основі одиничних блоків); д) двері для 2D-фасаду (на основі одиничних блоків); е) простановка позначок на 2D-фасаді.

С 7.4. Символи для оформлення фасадів: а) люди; б) дерева; в) транспорт.

*С 7.5.* Розрізи: а) швидке отримання 3D-розрізу; б) отримання 2D-розрізу; в) вікна для 2D-розрізу; г) двері для 2D-розрізу; д) відрисовування позначок для 2D-розрізу; е) відрисовування маркування вузла; ж) відрисовування сполучення перекриття та стіни в розрізі.

*С 8.* Можливості САПР, які забезпечують роботу з приміщеннями та підготування текстових документів.

*С 8.1.* Робота з приміщеннями: а) створення приміщень різних форм та маркування обраного з бібліотеки виду; б) маркування приміщень; в) перенесення приміщення; г) копіювання приміщень; д) корегування площі приміщень; е) фіксування зв'язків між приміщеннями та маркуванням; ж) створення списку приміщень усього відділу; з) штрихування приміщень відділу; і) штрихування приміщень за іменем; к) конвертування приміщень в стіни; л) присвоєння приміщенням типу обробки; м) тип обробки для вказаних приміщень.

*С 8.2.* Відомості, специфікації, експлікації: а) формування на основі файлів шаблонів; б) формування на основі підготовлених форм; в) генератор довільних форм специфікацій; г) можливість корегування вилученої інформації до формування з неї таблиці; д) перевизначення номерів для автоматично виставлених маркувань; е) урахування кількості поверхів при витязі інформації.

*С 8.3.* Формат отримання даних: а) у вигляді таблиці в кресленні; б) у вигляді текстового ASCII-файлу; в) у вигляді DXF- та DBF-файлів.

*С 9.* Можливості САПР, які забезпечують роботу з бібліотеками.



С 9.1. Система ведення бібліотек: а) система ведення бібліотек (параметрична, база блоків); б) доступ до символів бібліотеки (з розділів випадаючого графічного меню САПР; крізь діалогове вікно *DCL* менеджера символів); в) формування бібліотек (за допомогою програм, які є зовнішніми по відношенню до САПР; всередині САПР); г) можливості з редагування (додавання елементів, видалення елементів, пересування символів між розділами, копіювання символів).

С 10. Склад бібліотеки.

С 10.1. Склад параметричної бібліотеки (для відрисовування та заповнення маркування): а) вікна; б) двері; в) вбудовані шафи; г) перемички (відповідно до ДЕСТ); д) підлоги; е) типи обробок приміщень; ж) підвіконні плити; з) рандбалки; і) забивні сваї; к) плити перекриттів; л) колони; м) рігелі; н) діафрагми; о) панелі; п) елементи конструкцій.

С 10.2. Склад символної бібліотеки (в вигляді блоків типу ".DWG"): а) меблі (суспільного призначення, для житлових приміщень, кухонні та обладнання, конторські, спальні); б) меблеві аксесуари; в) стільці та крісла; г) оргтехніка; д) світильники; е) автомобілі; ж) рослини; з) дерева; і) транспорт.

С 10.3. Робота з символами: а) автоматичне розташовування мийок з полицею; б) автоматичне розташовування санвузлів; в) автоматичне розташовування пісуарів; г) вставка символу за прикладом; д) конвертування 3D-символів у 2D- та навпаки; е) заміна символу на той, що вказаний у кресленні або обраний з бібліотеки; ж) створення легенди символів, вставлених до креслення; з) редагування легенди символів, вставлених до креслення.

Це завдання складене на основі матеріалів статей-оглядів архітектурно-будівельних САПР, опублікованих журналом "Компьютер Пресс".

**3-я група завдань.** Здійснюючи комплексне використання систем інженерної комп'ютерної графіки, виконайте для того підприємства, де ви проходите виробничу практику або працюєте, такий проект.

22. Промисловий дизайн.

23. Архітектурний проект.

24. Повномасштабний завершений елемент графічної діалогової системи (ГДС) для підтримки виробничих процесів.

25. Цілісний продукт графічних технологій виробничого призначення, розроблений із застосуванням готових ГДС.

### Висновки до розд.3

Базові галузі практичного застосування графічних комп'ютерних технологій становлять ділова, наукова, ілюстраційна та інженерна комп'ютерна графіка та дизайн.

В третьому розділі було розглянуто такі важливі аспекти цих прикладних галузей: визначення, в якому сформульоване їх основне практичне призначення; сфери використання та ключові прикладні задачі, які вони охоплюють; класифікацію провідного програмного забезпечення та його користувачів; базові форми графічного подання даних у процесі розв'язування типових прикладних задач; можливості створення, обробки та *2D/3D*-моделювання графічних зображень (побудови, редагування, художнього оформлення (дизайну), аналізу, модифікації, трансформації, деформації, отримання проєкцій та перерізів, анімації, реалістичної візуалізації, створення спецефектів тощо), їх мультимедіа-дизайну, відеодизайну та *Web*-дизайну, а також їх документування засобами провідних графічних пакетів у процесі розв'язування типових прикладних задач; можливості отримання друкованої графічної продукції.

---

## ПІСЛЯМОВА

---

Комп'ютерна графіка та дизайн являє собою таку галузь знань, в якій: з одного боку, накопичено солідний багаж теоретичних знань та практичного досвіду; з іншого боку, здійснюється постійний розвиток апаратного та програмного забезпечення, методів, технологій та практичних застосувань.

Зрозуміло, що в одному, не дуже великому за обсягом виданні, авторам не вдалося розкрити всі аспекти сучасної комп'ютерної графіки та дизайну. Їх метою було: викласти той навчальний матеріал, який становитиме собою основу для подальшого самостійного вивчення комп'ютерної графіки та дизайну; позбавитися від необхідності розглядати цей матеріал на лекціях (і таким чином здобути можливість викладати на лекціях більш складні питання сучасної комп'ютерної графіки та дизайну, що дійсно потребують їх пояснення викладачем); розв'язати актуальну проблему вирівнювання того початкового обсягу знань, з яким студенти приходять на прослуховування лекцій.

Автори планують не зупинятися на досягнутому, а випустити серію книжок з комп'ютерної графіки, де ключові питання цієї галузі знань буде розглянуто детальніше та з розкриттям складнішого математичного апарату.

---

**ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ**

---

<i>2D</i>	– <i>Two dimensions</i>
<i>3D</i>	– <i>Three dimensions</i>
<i>API</i>	– <i>Application Programming Interface</i>
<i>APS</i>	– <i>Active Pixel Sensor</i>
<i>ASP</i>	– <i>Advanced Signal Processor</i>
<i>AWT</i>	– <i>Abstract Windows Toolkit</i>
<i>CAD</i>	– <i>Computer Aided Design, Drafting and Drawing</i>
<i>CAE</i>	– <i>Computer Aided Engineering</i>
<i>CAM</i>	– <i>Computer Aided Manufacturing</i>
<i>CATV</i>	– <i>Cable TeleVision</i>
<i>CCD</i>	– <i>Charge Coupled Device</i>
<i>CD-DA</i>	– <i>Compact Disc-Digital Audio</i>
<i>CDRAM</i>	– <i>Cached Dynamic Random Access Memory</i>
<i>CMOS</i>	– <i>Complementary Metal-Oxide-Semiconductor</i>
<i>CMYK</i>	– <i>Cyan, Magenta, Yellow, black</i>
<i>CSS</i>	– <i>Cascading Style Sheets</i>
<i>CTP</i>	– <i>Computer-to-Plate</i>
<i>DBS</i>	– <i>Direct Broadcast Satellite</i>
<i>DCT</i>	– <i>Discrete Cosine Transform</i>
<i>DDR</i>	– <i>Double Data Rate</i>
<i>DPCM</i>	– <i>Differential Pulse Code Modulation</i>
<i>DPI</i>	– <i>Dots per inch</i>
<i>DRAM</i>	– <i>Dynamic Random Access Memory</i>
<i>DSP</i>	– <i>Digital Signal Processor</i>
<i>DTD</i>	– <i>Document Type Definition</i>

---

<i>EDA</i>	– <i>Electronic Design Automation</i>
<i>EDO</i>	– <i>Extended Data Out</i>
<i>EMBM</i>	– <i>Environment Mapped Bump Mapping</i>
<i>ESDRAM</i>	– <i>Enhanced SDRAM</i>
<i>FCRAM</i>	– <i>Fast Cycle Random Access Memory</i>
<i>FeRAM</i>	– <i>Ferroelectric Random Access Memory</i>
<i>FM</i>	– <i>Frequency Modulation</i>
<i>FPM</i>	– <i>Fast Page Mode</i>
<i>FPS</i>	– <i>Frames Per Second</i>
<i>GIS</i>	– <i>Geo Information System</i>
<i>HDTV</i>	– <i>High-Definition TeleVision</i>
<i>HFS</i>	– <i>Hierarchical File Structure</i>
<i>HTML</i>	– <i>Hyper Text Markup Language</i>
<i>ICD</i>	– <i>Installable Client Driver</i>
<i>IRC</i>	– <i>Internet Relay Chat</i>
<i>ITU</i>	– <i>International Telecommunication Union</i>
<i>LV</i>	– <i>Laser Vision</i>
<i>MAFC</i>	– <i>Microsoft Application Foundation Classes</i>
<i>MCD</i>	– <i>Mini Client Driver</i>
<i>MCI</i>	– <i>Media Control Interface</i>
<i>MDRAM</i>	– <i>Multibank Dynamic Random Access Memory</i>
<i>MIDI</i>	– <i>Musical Instruments Digital Interface</i>
<i>MRAM</i>	– <i>Magnetic Random Access Memory</i>
<i>NTSC</i>	– <i>National Television Standarts Committee</i>
<i>OCR</i>	– <i>Optical Character Recognition</i>
<i>PAL</i>	– <i>Phase ALternative</i>
<i>PDA</i>	– <i>Personal Digital Assistant</i>
<i>PDF</i>	– <i>Portable Document Format</i>
<i>PDM</i>	– <i>Project Document Manufacturing</i>

---

<i>RAM</i>	– <i>Random Access Memory</i>
<i>RDRAM</i>	– <i>Rambus Dynamic Random Access Memory</i>
<i>RGB</i>	– <i>Red, Green, Blue</i>
<i>RIP</i>	– <i>Rasterising Image Processor</i>
<i>SDRAM</i>	– <i>Synchronous Dynamic Random Access Memory</i>
<i>SGRAM</i>	– <i>Synchronous Graphics Random Access Memory</i>
<i>SIMD</i>	– <i>Single Instruction Multiple Data</i>
<i>SGI</i>	– <i>Servers, Graphics, Insight</i>
<i>SGML</i>	– <i>Standard Generalized Markup Language</i>
<i>TCL</i>	– <i>Transform, Clipping, Lighting</i>
<i>TCP</i>	– <i>Transform Control Protocol</i>
<i>UDP</i>	– <i>User Datagram Protocol</i>
<i>UML</i>	– <i>Unified Modeling Language</i>
<i>URL</i>	– <i>Uniform Resource Locator</i>
<i>VR</i>	– <i>Virtual Reality</i>
<i>VRML</i>	– <i>Virtual Reality Modeling Language</i>
<i>WMF</i>	– <i>Window Meta File</i>
<i>WORM</i>	– <i>Write Only Random Memory</i>
<i>WRAM</i>	– <i>Windows Random Access Memory</i>
<i>WT</i>	– <i>Wave Table</i>
<i>WYSIWYG</i>	– <i>What You See Is What You Get</i>
<i>WWW</i>	– <i>World Wide Web</i>
<i>XML</i>	– <i>eXtensible Markup Language</i>
АЗ	– Апаратне забезпечення
АЦП	– Аналогово-цифровий перетворювач
БД	– База даних
БЗ	– База знань
ЕПТ	– Електронно-променева трубка
ЕС	– Експертна система

ІЧ	– Інфра-червоний
КГ	– Комп'ютерна графіка
КЕ	– Кінцевий елемент
МКЕ	– Метод кінцевих елементів
ОЗП	– Оперативний запам'ятовувальний пристрій
ООП	– Об'єктно-орієнтоване програмування
ОС	– Операційна система
ПЗ	– Програмне забезпечення
ПЗЗ	– Пристрій (прилад) із зарядовим зв'язком
ПЗП	– Постійний запам'ятовувальний пристрій
ПК	– Персональний комп'ютер
ПЛІС	– Програмована логічна інтегральна схема
САПР	– Система автоматизованого проектування
СУБД	– Система управління базами даних
ФНА	– Фотонабірний автомат
ЦАП	– Цифро-аналоговий перетворювач
ЦП	– Центральний процесор
ЦПК	– Центральний процесор комп'ютеру



## ЛІТЕРАТУРА

1. *Adobe Illustrator*. Учебник. – К.: Диасофт, 2002. – 368 с.
2. *Overview of the MPEG4 Standard// ISO/IEC JTC1/SC29 /WG11 //1998. – October 1998. – p.5-48.*
3. *Microsoft Excel 2000*: Справочник /Под ред. Ю.Колесникова. – СПб.: Питер, 2000. – 480 с.
4. *Microsoft Office 2000*: справочник/Под ред. Ю.Колесникова. – СПб.: Питер, 2000. – 540 с.
5. *Microsoft Word 2000*: Справочник /Под ред. Ю.Колесникова. – СПб.: Питер, 2000. – 352 с.
6. *Rob Koenen. MPEG-4: Multimedia For Our Time //IEEE Spectrum. – 1999.– February. – p.20-82.*
7. Абраш М. Тайнства программирования графики. – К.: ЕвроСИБ, 2002. – 512 с.
8. Акоста Р. Внутренний мир *World Wide Web*. – М.: ДМК, "Радио и связь", 2002. – 530 с.
9. Алексеев А. *AutoCAD*: Спец. справочник. – СПб.: Питер, 2003. – 688 с.
10. Аммерал Л. Интерактивная трехмерная машинная графика. – М.: Сол Систем, 1992. – 317 с.
11. Аммерал Л. Машинная графика на персональных компьютерах. – М.: Сол Систем, 1992. – 232 с.
12. Аммерал Л. Принципы программирования в машинной графике. – М.: Сол Систем, 1992. – 224 с.
13. Аммерал Л. Программирование графики на Турбо Си. – М.: Сол Систем, 1992. – 221 с.

14. Архангельский А.Я. Библиотека *C++ Builder*. 60 управляющих компонентов. – М.: Бином, 2002. – 268 с.
15. Бартенев О.В. Графика *OpenGL*. Программирование на Фортране. – М.: Диалог-Мифи, 2000. – 368 с.
16. Бартенев О.В. *Visual Fortran*. Новые возможности. – М.: Диалог-Мифи, 2002. – 304 с.
17. Барчард Б. Внутренний мир *AutoCAD*. – К.: Диасофт, 2003. – 688 с.
18. Белунцов В. *Flash*. Анимация в Интернете. – М.: Десс-Ком, 2002. – 325 с.
19. Березин С., Раков С. *Internet у вас дома*. 2-е изд. – СПб.: BHV-Санкт-Петербург, 2002. – 736 с.
20. Библия *Web-мастера*. – К.: Диалектика, 2003. – 916 с.
21. Биркмайер К. Общество Плоской Земли //Мульти-медиа.: Цифровое видео. – №4. - 1998. – с.90-112.
22. Блатнер Дж. Использование *MS Excel 2000*: Спец. издание. – М.: ДМК, "Радио и связь", 2000. – 620 с.
23. Бобровский С. Программирование на языке *QBASIC* для школьников и студентов. – М.: Десс; Инфорком-Пресс, 1999. – 208 с.
24. Бобровский С. Самоучитель программирования на языке *C++* в системе Borland *C++ Builder*. – М.: Десс; Инфорком-Пресс, 2002. – 312 с.
25. Божко А.Н. *Adobe Frame Maker*. Сложная верстка. – М.: ДМК, 2002. – 576 с.
26. Борланд Р. Эффективная работа с *Word*. – СПб.: Питер, 2002. – 960 с.
27. Браун П. *Adobe Web-дизайн*: Энциклопедия. – М.: ДМК, "Радио и связь", 2002. – 423 с.

28. Брой Д. Информатика. – М.: ДМК, "Радио и связь", 2002. – 512 с.
29. Броумбах-Дунка Н., Шерил М. *Flash* на примерах. – М.: Вильямс, 2002. – 368 с.
30. Бурлаков А. Самоучитель по компьютерной графике. – М.: ДМК, "Радио и связь", 2002. – 436 с.
31. Вагнер Р. *Java Script*. Энциклопедия пользователя. – К.: Диасофт, 2002. – 464 с.
32. Вайк А. *Java Script* в примерах. – К.: Диасофт, 2000. – 304 с.
33. Васильков Ю.В. Компьютерные технологии вычислений в математическом моделировании: учебное пособие. – М.: ДМК, "Радио и связь", 2000. – 376 с.
34. Велихов С. *Archicad*. Руководство по работе с программой. – М.: Мир, 2002. – 416 с.
35. Водески Р. *Графика для Web*: Библия дизайнера. – К.; М.: Диалектика, 2002. – 651 с.
36. Все, что вы хотите знать об *Internet*. – Москва: МИК, 2003. – 537 с.
37. Фокс А., Пратт М. Вычислительная геометрия. – М.: Мир, 1982. – 304 с.
38. Габбасов Ю. *Internet*. – СПб.: BHV-Санкт-Петербург, 2003. – 462 с.
39. Гардан Й., Люка М. Машинная графика и автоматизация конструирования. – М.: Мир, 1987. – 270 с.
40. Гарнаев А. Использование *MS Excel* и *VBA* в экономике и финансах. – М.: ДМК, "Радио и связь", 2002. – 512 с.

41. Геометрическое моделирование и машинная графика в САПР: Учебник /В.Е.Михайленко, В.Н.Кислоокий, А.А.Лященко и др. – К.: Выща шк., 1991. – 374 с.
42. Гилберт С. *Visual C++ 6*. Самоучитель. В примерах. – К.: Диасофт, 2000. – 496 с.
43. Гилой В. Интерактивная машинная графика. – М.: Мир, 1981. – 380 с.
44. Гончаров П. Компьютер для менеджера. – М.: ДМК, "Радио и связь", 2000. – 415 с.
45. Горячев А. Практикум по компьютерным технологиям. – М.: ДМК, "Радио и связь", 2002. – 297 с.
46. Графики функций: Справочник / Н.А.Вирченко, И.И.Ляшко, К.И.Швецов – К.: Наук. думка, 1979. – 320 с.
47. Гук М. Аппаратные средства РС: Энциклопедия. – СПб.: Питер, 2003. – 834 с.
48. Гультияев А. Визуальное моделирование в среде *Mathlab*. – СПб.: Питер, 2002. – 432 с.
49. Гультияев А.К. Уроки *Web*-мастера. – СПб.: Корона принт, 2003. – 480 с.
50. Гурский Ю. Эффективная работа с *Photoshop*. Трюки и эффекты. – СПб.: Питер, 2003. – 384 с.
51. Денисов А. *Microsoft Internet Explorer*: Справочник. – СПб.: Питер, 2002. – 468 с.
52. Додж М., Кината К., Стинсон К. Эффективная работа с *MS Excel 2000*. – СПб.: Питер, 2000. – 1072 с.
53. Дьяконов В. *MathCAD 2003*. Учебный курс. – СПб.: Питер, 2003. – 624 с.
54. Дьяконов В. *MathCAD 2000*. Учебный курс. – СПб.: Питер, 2000. – 592 с.

55. Дьяконов В. *Mathematica*. Учебный курс. – СПб.: Питер, 2003. – 656 с.
56. Дьяконов В. *MathLab*. Учебный курс. – СПб.: Питер, 2002. – 560 с.
57. Дубина Л. Машиностроительные расчеты в среде *Excel*. – М.: ДМК, "Радио и связь", 2002. – 283 с.
58. Дюк Д. Обработка данных на ПК в примерах. – М.: ДМК, "Радио и связь", 2000. – 627 с.
59. Евсеев Г. *Maya*. Анимация и специальные эффекты. – М.: ДЕСС-КОМ, 2002. – 448 с.
60. Завьялов Ю.С., Леус В.А., Скороспелов В.А. Сплаины в инженерной геометрии. – М.: Машиностроение, 1985. – 224 с.
61. Зозулевич Д.М. Машинная графика в автоматизированном проектировании. – М.: Машиностроение, 1976. – 240 с.
62. Зубков С.В. *Assembler* для *DOS*, *Windows* и *Unix*. – М.: ДМК-пресс, 2000. – 608 с.
63. Информатика. Базовый курс / Симонович С.В. и др. – СПб.: Питер, 2003. – 640 с.
64. Иванов В.П., Батраков А.С. Трехмерная компьютерная графика / Под ред. Т.М.Полищука. – М.: Радио и связь, 1995. – 223 с.
65. Кайл Л. *Web-профессионалам: Flash*. – К.: BHV, 2003. – 304 с.
66. Карлашук Д. Электронная лаборатория на *IBM PC*. – М.: ДМК, "Радио и связь", 2000. – 371 с.
67. Карлберг П. Бизнес-анализ в *Excel*. – М.: ДМК, "Радио и связь", 2000. – 412 с.
68. Карпов Б. *Microsoft Visio 2000*: Краткий курс. – СПб.: Питер, 2001. – 256 с.

69. Кент П. *World Wide Web*. – М.: Компьютер: ЮНИТИ, 2002. – 452 с.
70. Кирсанов Д. Веб-дизайн. – М.: ДМК, "Радио и связь", 2002. – 541 с.
71. Кишик А.Н. *Adobe Photoshop*. Эффективный самоучитель. – К.: Диасофт, 2003. – 304 с.
72. Кобурн Ф., Маккормик П. Эффективная работа с *Corel-DRAW*. – СПб.: Питер, 2003. – 948 с.
73. Козье Д. Электронная коммерция. – М.: ДМК, "Радио и связь", 2002. – 457 с.
74. Комягин В.Б. *Flash 5*. Самоучитель. – М.: Триумф, 2003. – 480 с.
75. Конопка Р. Создание оригинальных компонент в среде *Delphi*. – К.: НИПФ – "ДиаСофт ЛТД", 2001. – 512 с.
76. Коноховский Д. Экономическая информатика: Учебник. – М.: ДМК, "Радио и связь", 2000. – 516 с.
77. Корриган Д. Компьютерная графика: Секреты и решения. – М.: ИЧП "Энтроп", 2001. – 385 с.
78. Крамер Э. HTML. Наглядный курс *Web*-дизайна. – М.: Вильямс, 2001. – 304 с.
79. Красильникова А. Автоматизация инженерно-графических работ: Учебник. – М.: ДМК, "Радио и связь", 2002. – 478 с.
80. Крейнак Дж., Хейбрекен Дж. Интернет: Энциклопедия. – СПб.: Питер, 2002. – 612 с.
81. Кристиансон Т. *Perl*. Библиотека программиста. – СПб.: Питер, 2001. – 736 с.
82. Кроудер Д. Разработка *Web*-узлов для "чайников". – М.: Вильямс, 2003. – 336 с.

83. Круглински Д. Программирование на *Visual C++ 6.0*. – СПб.: Питер, 2003. – 864 с.
84. Кузьменко В.Г. *VBA 2000*. Самоучитель. – М.: Бином, 2000. – 408 с.
85. Кулагин Б.Ю. *3D Studio MAX 4.0*: от объекта до анимации. – СПб.: БХВ – Петербург, 2003. – 448 с.
86. Кэнту М. *Delphi* для профессионалов. – СПб.: Питер, 2003. – 944 с.
87. Леонтьев Ю. Самоучитель *Word 2000*. – СПб.: Питер, 2000. – 436 с.
88. Либерти Д. *C++*. Энциклопедия пользователя. – К.: Диасофт, 2002. – 592 с.
89. Лит М. *Flash* для "чайников". – М.: Вильямс, 2003. – 320 с.
90. Лью Р. Цифровая кинематография. – М.: ДМК, "Радио и связь", 2002. – 318 с.
91. Макарова В. Информатика: учеб. пособие. – М.: ДМК, "Радио и связь", 2002. – 552 с.
92. Мак-Клеланд Д. *Photoshop* для "чайников". – М.: Вильямс, 2003. – 368 с.
93. Маров М. *3D Studio MAX*: Справочник. – СПб.: Питер, 2002. – 672 с.
94. Маров М. *3D Studio MAX*: Учебный курс. – СПб.: Питер, 2003. – 608 с.
95. Мартинер У. *VB* для *Web*. Руководство разработчика. – К.: BHV, 2002. – 384с.
96. Марченко А.И. Программирование в среде Турбо Паскаль 7.0. – К.: Век+, 2000. – 464 с.

97. Маслов В. Основы программирования на языке *Java*. – М.: *Telecom*, 2003. – 132 с.
98. Математика и САПР: В 2-х кн. Кн.1./ Шенен П., Коснар М., Гардан И. и др. – М.: Мир, 1988. – 204 с.
99. Матчо Д., Фолкнер Д. *Delphi*. – М.: БИНОМ, 1999. – 464 с.
100. Мелихов. Интернет: Энциклопедия. – 3-е изд. – М.: ДМК, "Радио и связь", 2000. – 516 с.
101. Мец К. *Web* по индивидуальному заказу // *PC Magazine / Russian Edition*. – 2000. – № 10. – С.76-84.
102. Мешков А. *Visual C++* и *MFC*. – СПб.: *BHV*, Петербург, 2003. – 1040 с.
103. Миронов Д. *CorelDRAW*: Учебный курс. – СПб.: Питер, 2003. – 464 с.
104. Мураховский В. Сборка, настройка, апгрейд современного компьютера. – М.: Десс; Инфорком-Пресс, 2001. – 291с.
105. Муртазин Д. Интернет: Учебник. – М.: ДМК, "Радио и связь", 2002. – 376 с.
106. Немнюгин С. *Pascal*: Учебный курс. – СПб.: Питер, 2003. – 542 с.
107. Нидерст Д. *Web-мастеринг* для профессионалов. – СПб.: Питер, 2001. – 576 с.
108. Новейший самоучитель работы на компьютере/Под ред. С.Симоновича – М.: Десс; Инфорком-Пресс, 2003. – 656 с.
109. Ньюмен У., Спрулл Р. Основы интерактивной машинной графики. – М.: Мир, 1976. – 574 с.
110. Ойер К. Техническая поддержка *Photoshop*. – К.: Комиздат, 2002. – 256 с.



111. Осипов В.А. Машинные методы проектирования непрерывно-каркасных поверхностей. – М.: Машиностроение, 1979. – 248 с.
112. Павловская Т.А. *C/C++*. Программирование на языке высокого уровня. – СПб.: Питер, 2001. – 464 с.
113. Панкратова Т. *Photoshop*. Учебный курс. – СПб.: Питер, 2031. – 480 с.
114. Пекарев Л.Р. Самоучитель *3D Studio MAX 4*. – СПб.: BHV, 2001. – 304 с.
115. Пол А. ООП на C++. – М.: Бином, 1999. – 462 с.
116. Полевой Р. *3D Studio MAX* для профессионалов. – СПб.: Питер, 2001. – 848 с.
117. Пономаренко С. *CorelDraw*. Изучаем вместе с *BHV*. – СПб.: BHV, 1998. – 352 с.
118. Потемкин В.Г. *MathLab*. В 2-х т. – М.: Диалог-МИФИ, 1999. – 304 с.
119. Потемкин А. Инженерная графика: просто и доступно. – М.: ДМК, "Радио и связь", 2000. – 537 с.
120. Прикладна геометрія та інженерна графіка: Міжвідомчий науково-технічний збірник. – К.: КНУБА, 2000-2003.
121. Пфаффенбергер Б. Эффективная работа с *Microsoft Internet Explorer*. – СПб.: Питер, 2002. – 452 с.
122. Разевич В.Д. Система *P-CAD*. – М.: СОЛОН-Р, 1999. – 720 с.
123. Разевич В.Д. Система проектирования цифровых устройств *OrCAD*. – М.: СОЛОН-Р, 2000. – 160 с.
124. Роджерс Д., Адамс Дж. Математические основы машинной графики. – М.: Машиностроение, 1980. – 240 с.

125. Бадаєв Ю.І., Аушева Н.М. Розробка САПР на базі Автокад-технології.: Навч. посібник. – К.: ІСДО, 1993. – 176 с.
126. Романычева Э.Т. *AutoCAD*. – М.: ДМК, 1999. – 480 с.
127. *AutoCAD*. Практическое руководство /Э.Т.Романычева, Т.М.Сидорова, С.Ю.Сидоров. – М.: ДМК, "Радио и связь", 2002. – 480 с.
128. Романычева Э.Т. Трехмерное моделирование в *AutoCAD*. – М.: ДМК, "Радио и связь", 2003. – 386 с.
129. Рубенкинг Н. Программирование в *Delphi* для "чайников". – К.: Диалектика, 2000. – 302 с.
130. Руденко А. Практичний курс інформатики. – К.: ВНУ, 2002. – 527 с.
131. Роуз К. Освой самостоятельно *Adobe Photoshop* за 24 часа. – М.: Вильямс, 2000. – 416 с.
132. Рудаков П.И. Обработка сигналов и изображений *MathLab*. – М.: Диалог-МИФИ, 2000. – 416 с.
133. Рудометов Е. Аппаратные средства и мультимедиа: Справочник. – СПб.: Питер, 2002. – 483 с.
134. Рычков В. Самоучитель *Excel 2000*. – СПб.: Питер, 2000. – 336 с.
135. Самофалов С.П. *Java на Web* – модно или необходимо? //Компьютеры + программы. – 1998. – № 11. – с.35-39.
136. Симонович С. Информатика для юристов и экономистов. – СПб.: Питер, 2002. – 389 с.
137. Симонович С., Евсеев Г., Алексеев А. Общая информатика. – М.: АСТ-ПРЕСС; Инфорком-Пресс, 2002. – 592 с.
138. Симонович С., Евсеев Г. Практическая информатика: универсальный курс. – М.: АСТ-ПРЕСС; Инфорком-Пресс, 2002. – 480 с.

139. Симонович С., Евсеев Г., Алексеев А. Специальная информатика: универсальный курс. – М.: АСТ-ПРЕСС; Информ-Ком-Пресс, 2002. – 480 с.
140. Соен Ф. Внутренний мир *AutoCAD*. – К.: Диасофт, 1997. – 704 с.
141. Справочник по машинной графике в проектировании / Под ред. В.Е.Михайленко, А.А.Лященко. – К.: Будівельник, 1984. – 184 с.
142. Бронштейн И.Н., Семендяев К.А. Справочник по математике для инженеров и учащихся втузов – М.: Наука, гл. ред. физ.-мат. лит.-ры, 1981. – 720 с.
143. Степаненко А. Персональный компьютер: учебный курс. – М.: ДМК, "Радио и связь", 2003. – 612 с.
144. Стоцкий Ю. Самоучитель *Office 2000*. - СПб.: Питер, 2000. – 576 с.
145. Стразницкас М. Эффективная работа с *Photoshop*. – СПб.: Питер, 2002. – 704 с.
146. Страуструп Б. Язык программирования C++. – М.: Бином, 2001. – 1099 с.
147. Тайц А.М. *Adobe in Design* в подлиннике. – СПб.: BHV – СПб, 2002. – 704 с.
148. Тайц А.М., Тайц А.И. *Adobe Illustrator*: Учебный курс. – СПб.: Питер, 1999. – 608 с.
149. Тайц А.М., Тайц А.И. *CorelDRAW*. Краткий курс. – СПб.: Питер, 2002. – 328 с.
150. Тайц А.М. *Corel Draw 10*. Самоучитель. - СПб.: BHV, 2001. – 640 с.
151. Тимофеев В.В. C/C++. Программирование в среде C++ *Builder 5*. – М.: Бином, 2000. – 368 с.

152. Уайт Д. *Visual C++ 6 и MFC*. Энциклопедия пользователя. – К.: Диасофт, 2000. – 720 с.
153. Уварова А. *P-CAD 2000 ACCEL EDA*. Учебный курс. – СПб.: Питер, 2001. – 320 с.
154. Уокер Б.С., Гурд Дж., Дроник Е.А. Интерактивная машинная графика. – М.: Машиностроение, 1980. – 171 с.
155. Фелистов Э. *Archicad*. Справочник. – М.: Познательная книга плюс, 2002. – 192 с.
156. Фигурнов П. *IBM PC для пользователей*. – М.: ДМК, "Радио и связь", 2000. – 617 с.
157. Фокс А., Пратт М. Вычислительная геометрия. – М.: Мир, 1982. – 304 с.
158. Фоли Дж., Вэн Дэм А. Основы интерактивной машинной графики: В 2-х кн. – М.: Мир, 1985. – Кн.1. – 367с.; Кн.2. – 386 с.
159. Фоминов О. Мультимедиа и сети //Мультимедиа и цифровое видео. – 1997. – №№ 5-8.
160. Франка П. *C++*. Учебный курс. – СПб.: Питер, 2001. – 528 с.
161. Фрей Д. *AutoCAD на примерах*. – К.: Юниор, 2002. – 544 с.
162. Хабибуллин И. Самоучитель *Java*. – СПб.: BHV, 2001. – 464 с.
163. Халпери П. Стандартная библиотека *C++* на примерах. – М.: Вильямс, 2001. – 336 с.
164. Хаус Р. Использование *AutoCAD*. Специальное издание: Учеб. пособие. – М.: Издат. дом Вильямс, 2000. – 832 с.

165. Хеллер Д. Мультимедийные презентации в малом бизнесе. – К.: *BHV*, 2002. – 272 с.
166. Хокс В. Автоматизированное проектирование и производство: Пер. с англ. – М.: Мир, 1991. – 296 с.
167. Холмогоров А. Основы *Web*-мастерства. – М.: ДМК, "Радио и связь", 2002. – 527 с.
168. Хольценлаг М.Э. Использование *HTML 4*. – М.: Вильямс, 2001. – 1008 с.
169. Хольцнер С. *Dynamic HTML*. Руководство разработчика. – К.: *BHV*, 2002. – 400 с.
170. Чекотовський Д. Графічні методи в статистиці на основі *Excel 97/2000*. – К.: *BHV*, 2000. – 531 с.
171. Шафрин Э. Информационные технологии. – М.: ДМК, "Радио и связь", 2000. – 457 с.
172. Шафрин Э. Создание *Web*-страниц: Самоучитель. – СПб.: Питер, 2001. – 320 с.
173. Шейнблит А.Е. Курсовое проектирование деталей машин: Учеб. пособие для техникумов. – М.: Высш. шк., 1991. – 432 с.
174. Шикин Е.В., Боресков А.В., Зайцев А.А. Начала компьютерной графики. – М.: ДИАЛОГ-МИФИ, 1993. – 138 с.
175. Ширяев А. *Web*-дизайн для всех: свое имя в сети // *Hard and Soft*. – 1998. – №10. – С.48-51.
176. Шпунт Я.Б. Сканирование: лучшие программы, полезные советы. – М.: ДМК, 2000. – 432 с.
177. Эбнер М. Руководство разработчика *Delphi 5*. – К.: *BHV*, 2000. – 480 с.
178. Юров В. *Assembler*: Практикум. – СПб.: Питер, 2001. – 400 с.

---

179. Юров В. *Assembler*: Учебник. – СПб.: Питер, 2001. – 624 с.

180. Якушин Е. Изучаем *Internet*, создаем *Web*-страничку. – СПб.: Питер, 2000. – 256 с.

---

---

## ВІДОМОСТІ ПРО АВТОРІВ

---

---

**ВЕСЕЛОВСЬКА Галина Вікторівна** – кандидат технічних наук, доцент кафедри інформаційних технологій факультету кібернетики Херсонського державного технічного університету.

Адреса: Україна, м. Херсон, Бериславське шосе, 24, ХДТУ, корп.3, кімн.316; р.т. 55-17-31; д.т. 22-19-53; *Internet: E-mail: kstu@tlc.kherson.ua.*

**ХОДАКОВ Віктор Єгорович** – заслужений діяч науки і техніки України, доктор технічних наук, професор, завідувач кафедрою інформаційних технологій факультету кібернетики Херсонського державного технічного університету.

Адреса: Україна, м.Херсон, Бериславське шосе, 24, ХДТУ, корп.3, кімн.316; р.т. 55-17-31, 32-69-66; *Internet: E-mail: kstu@tlc.kherson.ua, hod@ist.com.ua.*

**ВЕСЕЛОВСЬКИЙ Віктор Михайлович** – доцент кафедри фізики та електротехніки факультету кібернетики Херсонського державного технічного університету.

Адреса: Україна, м.Херсон, Бериславське шосе 24, ХДТУ, корп.1, кімн.127; р.т. 32-69-23; д.т. 22-19-53; *Internet: E-mail:kstu@tlc.kherson.ua.*





**Веселовська Галина Вікторівна  
Ходаков Віктор Єгорович  
Веселовський Віктор Михайлович**

# **КОМП'ЮТЕРНА ГРАФІКА**

**Навчальний посібник  
для студентів вищих навчальних закладів**

Підписано до друку 05.08.2008 р.  
Формат 84×108/32. Папір офс. № 1  
Ум. друк арк.30,66. Наклад 300 прим. Дог. № 01/04

Видавництво "ОЛДІ-плюс"  
м.Херсон, вул.Кірова, 24, оф.207  
Тел./факс (0552) 24-12-20; e-mail: oldi@public.kherson.ua  
Ліцензія сер. ХС № 2 від 16.08.2000 р.