

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ

до виконання практичних і лабораторних робіт

«Основи комп'ютерної графіки. Частина 1»

з курсу «Комп'ютерна графіка»
для студентів спеціальностей
151 «Автоматизація та комп'ютерно-інтегровані технології»,
172 «Телекомунікація та радіотехніка»

Рекомендовано
редакційно-видавничою
радою університету,
протокол №1 від 19.02.2020

Методичні вказівки до виконання практичних і лабораторних робіт «Основи комп'ютерної графіки. Частина 1» з курсу «Комп'ютерна графіка» для студентів спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології», 172 «Телекомунікація та радіотехніка» денної та заочної форм навчання / уклад. А. О. Зуєв, О. М. Євсеєнко, В.А. Крилова. – Харків : НТУ «ХП». – 51 с.

Укладачі А. О. Зуєв
О. М. Євсеєнко
В.А. Крилова

Рецензент І.В. Григоренко

Кафедра інформаційно-вимірювальних технологій і систем

ЗМІСТ

Вступ	4
1 ЗАГАЛЬНІ ВІДОМОСТІ	5
1.1 Растрова графіка	6
1.2 Векторна графіка	7
1.3 Тривимірна графіка	8
2 КОЛІР І СИСТЕМИ КОДУВАННЯ КОЛЬОРУ	11
2.1 Загальні терміни та визначення	11
2.2 Апаратно-залежне відображення кольору	13
2.3 Апаратно-незалежне відображення кольору	16
2.4 Перетворення кольору між різними колірними моделями	19
3 ТРИВИМІРНА ГРАФІКА	23
3.1 Подання моделей об'єктів в 3D графіці	23
3.2 Створення моделі об'єкта	25
4 ПРОЦЕС ВІЗУАЛІЗАЦІЇ (РЕНДЕРИНГУ)	34
4.1 Системи координат в 3D графіці	34
4.2 Перетворення координат	36
4.2.1 Масштабування	37
4.2.2 Переміщення	38
4.2.3 Обертання	39
4.3 Світова система координат	40
4.4 Система координат спостерігача і камера	41
4.4.1 Відсікання	42
4.4.2 Проеціювання	44
4.5 Растеризація	45
Список літератури	50

ВСТУП

Подання даних на моніторі комп'ютера в графічному вигляді вперше було реалізовано в середині 50-х років для великих ЕОМ, які застосовувалися в наукових і військових дослідженнях. Відтоді графічний спосіб відображення даних став невід'ємною частиною комп'ютерних систем.

Комп'ютерна графіка – це розділ інформатики, який вивчає методи створення та обробки графічних зображень за допомогою обчислювальної техніки.

Комп'ютерна графіка використовується майже в усіх наукових та інженерних дисциплінах для збільшення наочності та поліпшення сприйняття інформації людиною. Без комп'ютерної графіки неможливо уявити собі не тільки комп'ютерний, але і звичайний, цілком матеріальний світ. Незважаючи на те, що комп'ютерна графіка є, по суті, інструментом, її структура і методи засновані на передових досягненнях фундаментальних і прикладних наук: математики, фізики, хімії, біології, статистики, програмування тощо. Це дійсно як для програмних, так і для апаратних засобів створення та обробки зображень на комп'ютері.

Тому комп'ютерна графіка як галузь інформатики активно розвивається і багато в чому виступає рушієм усієї комп'ютерної індустрії.

Найважливішою частиною комп'ютерної графіки є візуалізація – загальна назва прийомів подання інформації або явищ у вигляді, зручному для зорового спостереження та аналізу. Візуалізація даних застосовується в різних сферах людської діяльності: у медицині (комп'ютерна томографія), військовій справі (тренажерні комплекси), наукових дослідженнях (візуалізація будови речовини, векторних полів та інших даних), моделюванні тканин та одягу, дослідно-конструкторських розробках, індустрії розваг.

Методичні вказівки призначені для вивчення основ комп'ютерної графіки та її застосування у створенні систем візуалізації. Розглянуто питання кодування кольору, растеризації, подання та візуалізації тривимірних об'єктів, особливості програмно-апаратних систем тривимірної графіки.

Методичні вказівки містять також індивідуальні завдання для виконання лабораторних і практичних робіт, які допоможуть в ефективному освоєнні курсу комп'ютерної графіки.

1. ЗАГАЛЬНІ ВІДОМОСТІ

Комп'ютерна графіка (КГ) вивчає методи обробки і синтезу зображень за допомогою засобів обчислювальної техніки. Результатом роботи КГ є зображення, яке може використовуватися в різних сферах діяльності людини.

Особливості КГ:

а) найбільш природний для людини засіб спілкування з комп'ютером;
б) добре розвинений у людини механізм розпізнавання образів дозволяє дуже швидко й ефективно сприймати та обробляти різні види даних.

У КГ розглядаються наступні завдання:

- а) подання зображень;
- б) підготовка зображень до візуалізації;
- в) створення зображень;
- г) маніпуляції із зображеннями.

При обробці інформації, пов'язаної із зображеннями, виділяють три основних напрямки:

- а) розпізнавання образів (комп'ютерний зір);
- б) обробка зображень;
- в) машинна (комп'ютерна) графіка.

Основне завдання розпізнавання образів полягає в розпізнаванні наявного зображення і перетворення його в "мову символів". Це сукупність методів, що дозволяють виділяти суттєві частини зображень і віднести зображення до певного класу.

Обробка зображень розглядає завдання:

- а) усунення дефектів зображень (видаленням шумів);
- б) поліпшення зображень (контрастування, підвищення різкості);
- в) перехід від одного виду зображень до іншого.

Машинна графіка застосовується в тому випадку, коли вихідними даними є не зображення, а експериментальні дані, опис об'єкта або явища, результати математичного моделювання.

Наприклад, візуалізація експериментальних даних у вигляді графіків або гістограм, виведення графіки в комп'ютерних іграх або синтез навколишнього оточення для тренажерів.

За основним типом синтезованих зображень розрізняють три основних види КГ: растрова графіка, векторна графіка і тривимірна графіка.

1.1 Растрова графіка

Растровою графікою називається розділ КГ, який оперує з плоскими двовимірними зображеннями – растрами.

Растрове зображення – двовимірна матриця, кожна клітина якої "зафарбована" кольором, що в сукупності утворює малюнок.

Піксель – основний елемент растрових зображень, мінімальний елемент зображення. Це може бути окрема точка на екрані монітора, окрема точка, надрукована на принтері, або окремий елемент растрового зображення в пам'яті комп'ютера.

Призначення растрової графіки: передача зображень об'єктів реального світу, зберігання виведених зображень, отриманих іншими видами графіки.

Будь-який растровий малюнок має певну кількість пікселів у горизонтальних і вертикальних рядах. Поширені розміри: 320x240, 640x480, 800x600, 1024x768, 1280x1024, 1600x1200. Добуток цих двох чисел дає загальну кількість пікселів зображення.

Колір кожного пікселя растрового зображення зберігається у вигляді комбінації бітів. Чим більше бітів використовується, тим більше відтінків кольорів можна отримати (зберігати). Кількість бітів, що використовуються для кодування кольору пікселя, називається бітовою глибиною зображення, наприклад:

1 - чорно-біле зображення;

8 - зображення в градаціях сірого кольору (або індекс кольору в палітрі);

16, 24, 32 - повнокольорове зображення;

48 - кольорове зображення високої точності.

Переваги растрової графіки:

а) простота відображення на основних пристроях виведення, моніторах і принтерах;

б) растр ефективно подає реальні образи об'єктів.

Недоліки:

а) неможливість (складність) масштабування растрових зображень без втрати якості;

б) великий обсяг зображення. Чим більше в зображенні пікселів, тим детальніше воно виглядає але займає значно більший обсяг пам'яті. Така залежність від розмірів зображення квадратична.

Розмір файла растрової графіки значно залежить від формату, обраного для збереження зображення. За інших рівних умов, таких як розміри зображення і бітова глибина, істотне значення має схема (або

формат) стиснення зображення. Наприклад, файл формату BMP або TGA має зазвичай великі розміри у порівнянні з файлами форматів PNG і GIF, в яких використовується стиснення без втрат якості. А файли в форматах PNG і GIF зазвичай більші, ніж файли в форматі JPEG, для якого використовується стиснення з втратою якості.

1.2 Векторна графіка

Вид КГ, в якій зображення подається з використанням математичних описів об'єктів на площині, називається векторною графікою. Основним елементом векторної графіки є відрізок лінії. Таке подання дозволяє комп'ютеру обчислювати і поміщати в потрібному місці реальні точки при візуалізації цих об'єктів.

Призначення векторної графіки: графіки та діаграми, технічні ілюстрації і креслення, стилізовані малюнки і піктограми.

Особливість подання зображень векторної графіки дає їй ряд переваг перед растровою, наприклад, вільне масштабування, але в той же час є причиною її недоліків. Векторну графіку часто називають креслярською графікою.

Як і всі об'єкти, відрізки ліній мають властивості: форма лінії, її товщина, колір, характер лінії (суцільна або пунктирна). Замкнені лінії називаються контурами і можуть мати фактуру та колір для заповнення.

Прості об'єкти – як кола, еліпси, прямокутники і багатокутники, криві Безьє, сплайни – називаються графічними примітивами і використовуються при синтезі більш складних об'єктів. У векторній графіці складні об'єкти створюються шляхом комбінації декількох графічних примітивів.

Для виведення на монітор або принтер для векторних зображень потрібна попередня растеризація.

Растеризація – це процес перетворення зображення, описаного у векторному вигляді, в растр для виведення на дисплей або принтер.

Переваги:

- а) вільне масштабування без втрати якості зображення;
- б) невеликий обсяг займаної пам'яті, який не залежить від розміру зображення;
- в) відносна простота автоматичної генерації зображення;
- г) виведення на спеціальні пристрої, такі як графобудувачі та плотери без додаткового перетворення.

Недоліки:

- а) складність передачі зображень об'єктів реального світу;

- б) необхідність перетворення в растр при виведенні зображення на основні пристрої виведення: монітори і принтери (растеризація);
- в) ступінчастість фрагментів зображень (аліасінг) при перетворенні в растр.

Основною проблемою при растеризації є ступінчастість зображення - або аліасінг. *Аліасінг* виникає в тому випадку, коли частини векторних об'єктів ідуть під кутом до вертикальної або горизонтальної осі зображення. Він проявляється у вигляді "рваних країв" об'єктів, частин тонких ліній і об'єктів, які пропадають або блимають - розмір яких можна порівняти з розміром пікселя. Існує безліч різних методів антиаліасінгу, які видаляють або маскують цей неприємний ефект, але вони створюють додаткове навантаження на обчислювальну систему і вимагають додаткової пам'яті для зберігання та підготовки растра.

1.3 Тривимірна графіка

Вид КГ, в якій зображення будується за допомогою математичних описів об'єктів у просторі, називається тривимірною графікою. Основним елементом тривимірної графіки є трикутник.

Призначення: створення моделей реальних об'єктів і відображення їх взаємодії в просторі.

Тривимірна графіка вивчає методи створення моделей об'єктів у просторі, які максимально відповідають реальним. Спочатку створюється каркас об'єкта, потім його поверхню покривають матеріалами, візуально схожими на реальні. Після цього налаштовується освітлення. Отримані моделі можна обертати і розглядати з усіх боків. Усі тривимірні об'єкти, джерела освітлення та камери, а також допоміжні елементи утворюють тривимірну сцену.

Для задання позицій об'єктів і вершин трикутників у просторі використовуються *Декартові системи* координат, зазвичай, лівостороння або правостороння. Для подання і моделювання сукупності трьох тривимірних об'єктів що задють рисунок, зазвичай достатньо трьох систем координат:

- а) локальна система об'єкта, задає положення частин об'єкта;
- б) світова система, задає взаємне розташування об'єктів;
- в) система спостерігача, визначає видимі об'єкти з позиції спостерігача.

Сукупність систем координат і всіх об'єктів, які беруть участь у синтезі зображення, називається *сценою*.

Для опису системи координат використовуються перетворення: масштабування, обертання і зміщення. Для задання координат вершин

використовуються трикомпонентні вектори. Перетворення між системами координат здійснюється шляхом послідовного додавання перетворень, які описують систему координат, до вектору координат вершини.

Для отримання об'єктів на площині використовується операція проєціювання за допомогою центральної або паралельної проєкції.

Центральна (перспективна) проєкція приводить до візуального ефекту, подібного тому, який дає зорова система людини. При цьому спостерігається ефект перспективного укорочування, коли розмір проєкції об'єкта змінюється обернено пропорційно відстані від центру проєкції до об'єкта.

У **паралельних проєкціях** відсутнє перспективне укорочування, унаслідок чого зображення виходить менш реалістичним і паралельні прямі завжди залишаються паралельними.

Отримане після проєціювання векторне зображення rasterизується і виводиться на екран. Таким чином, змінюючи позицію спостерігача (або систему спостерігача), можна отримувати зображення трьовимірних об'єктів, які утворюють трьовимірну сцену.

Переваги:

- а) можливість синтезувати зображення об'єктів будь-якої форми і деталізації;
- б) невід'ємна частина отримання інформації при моделюванні об'єктів і процесів реального світу.

Недоліки:

- а) великі вимоги до продуктивності обчислювальної системи для перетворень і розрахунку освітлення тривимірних об'єктів;
- б) необхідність подвійного перетворення при виведенні зображення: спочатку в об'єкти на площині, а потім у растр;
- в) проблеми подібні до тих, що є у векторній графіці (аліасинг, складність отримання реалістичних зображень об'єктів).

Контрольні питання

1. Що таке комп'ютерна графіка?
2. Призначення і завдання комп'ютерної графіки.
3. Що таке машинна графіка?
4. Які завдання розглядає обробка зображень?
5. Призначення машинного зору.
6. Три основні види комп'ютерної графіки.
7. Що таке растрова графіка?

8. Що таке піксель?
9. Що таке растр?
10. Як розрахувати розмір пам'яті, займаної зображенням, в байтах? Що необхідно врахувати при цьому?
11. Переваги і недоліки растрової графіки.
12. Що таке векторна графіка?
13. Чим векторна графіка відрізняється від растрової?
14. Основний елемент векторної графіки і його властивості.
15. Що таке бітова глибина зображення?
16. Що таке графічні примітиви і для чого вони застосовуються?
17. Що таке растеризація?
18. Перерахуйте переваги і недоліки векторної графіки.
19. Що таке аліасинг? Коли він виникає?
20. Що таке тривимірна графіка?
21. Основний елемент тривимірної графіки.
22. Що таке система координат? Які системи використовуються в тривимірній графіці?
23. Що таке сцена в тривимірній графіці?
24. Для чого використовується світова система координат?
25. Для чого використовується система координат спостерігача?
26. Що таке проекція і для чого вона використовується?
27. Чим центральна проекція відрізняється від паралельної?
28. Що відбувається з трьохвимірними об'єктами в результаті проєкціювання?
29. Яка проекція дає результати, близькі до того, якій дає зорова система людини?
30. Переваги та недоліки тривимірної графіки.
31. Які особливості спільні для векторної і тривимірної графіки?
32. Для чого потрібен спостерігач в тривимірній графіці?

2. КОЛІР І СИСТЕМИ КОДУВАННЯ КОЛЬОРУ

2.1 Загальні терміни та визначення

Світло – це випромінювання, яке випускається нагрітим тілом або речовиною, яка знаходиться в збудженому стані.

Колір – це якісна характеристика світла (випромінювання), яка визначається на основі виникаючого зорового відчуття людини. Сприйняття кольору багато в чому залежить від людини: її фізичного і психологічного стану. Сітківка ока людини має 3 види рецепторів світла, відповідальних за кольоровий зір (колбочки). Кожен вид колб реагує на певний діапазон видимого спектру світла.

Колірний простір являє собою модель подання кольору, яка базується на використанні колірних координат і будується таким чином, щоб будь-який колір подавався точкою з певними координатами, для того щоб одному набору координат відповідав один колір.

Колірний простір можна визначити як Евклідов простір, якщо задати координати x , y і z як значення, що відповідають відгуку різних видів колбочок: довгохвильового (L), середньо-хвильового (M) і короткохвильового (S) діапазонів оптичного спектру. Початок координат (S, M, L) = (0, 0, 0) представлятиме чорний колір. Білий колір не матиме чіткої позиції в даному визначенні кольорів (рис. 2.1).

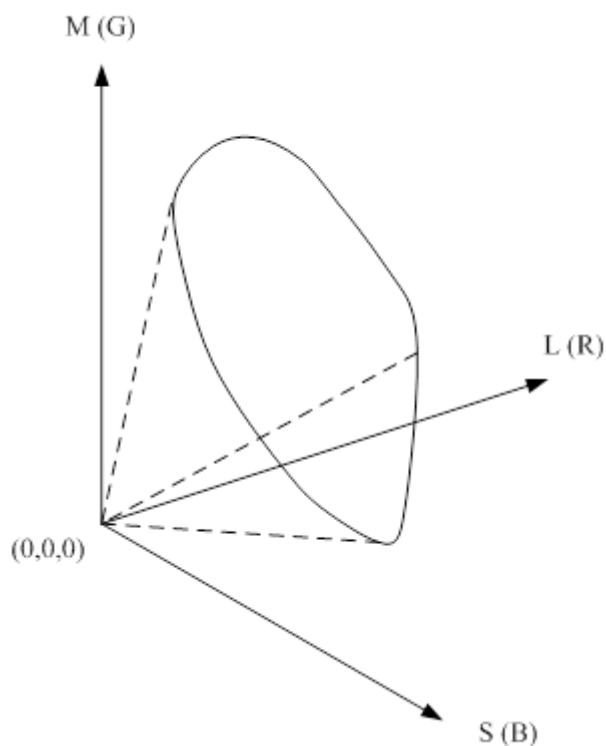


Рисунок. 2.1 – Видимий людиною колірний простір

Такий колірний простір матиме властивість адитивного змішування – сума двох колірних векторів (координат) буде відповідати кольору, отриманому змішуванням відповідних двох кольорів.

Зведене зображення дозволяє задавати колір будь-якої інтенсивності – починаючи від чорного кольору і до безкінечності. Однак на практиці зорові рецептори можуть перенасититись. Тому, дану модель, не можна застосовувати для опису кольору в умовах дуже високої або дуже низької інтенсивності випромінювання. До того ж, при низькій інтенсивності використовується інший механізм сприйняття кольору.

Білий колір визначається за допомогою колірної температури.

Колірна температура – це температура абсолютно чорного тіла, при якій воно випускає випромінювання того ж колірного тону, що й випромінювання, яке розглядається.

Колірна температура (табл. 2.1) визначає як буде виглядати та сприйматися колір при даних умовах освітлення.

Таблиця 2.1 – Розподіл джерел світла за колірною температурою

Колірна температура, К	Штучне джерело	Природне джерело
20000	-	Небо в полярних широтах
7500–6500	Люмінісцентна лампа	Хмарне небо
6000	Галогенні лампи	-
5500–5000	Електрична дуга	Сонце опівдні
4500–3500	Лампа денного світла	Небо вранці і ввечері
3000	Галогенні лампи (біле світло)	-
2800–2200	Лампа розжарювання (40–100 Вт)	-
2000	Полум'я свічки	Небо на заході і сході

Температура 5000 К – це колір, якого набуває абсолютно чорне тіло при нагріванні його до 5000 К. Колірна температура помаранчевого кольору – 2000 К. Це означає, що абсолютно чорне тіло необхідно нагріти до температури 2000 К, щоб воно набуло помаранчевого кольору світіння.

Колірна температура є характеристикою джерела світла. Будь-який

видимий нами колір має колірну температуру і не важливо, який це колір: червоний, малиновий, жовтий, пурпурний, фіолетовий, зелений, білий.

Для отримання максимально коректного кольорового зображення на всіх стадіях друку зображення зазвичай підтримують стандартну колірну температуру від 5000 К до 5500 К. В інших випадках білий колір може бути лише наближений до білого за допомогою балансу білого кольору. *Баланс білого кольору* – це установка значення колірної температури для всього зображення. При правильній установці кольори будуть відповідати тому зображенню, яке бачить людина. Якщо переважають сині та блакитні тони кольору, значить, кольори холодні, встановлена занадто низька колірна температура, необхідно її підвищити. Якщо ж переважає червоний тон – кольори теплі, встановлена занадто висока температура, необхідно її знизити.

Таким чином, у комп'ютерній графіці колір являє собою набір (вектор) чисел-координат у колірній системі.

2.2 Апаратно-залежне подання кольору

Для зберігання і обробки кольору використовуються різні методи та системи, обумовлені властивостями людського зору. Найбільш поширені системи кодування кольору, такі як: *RGB для моніторів і CMY(K) для друку*. Такі системи називаються апаратно-залежними, оскільки є "рідними" для двох найбільш поширених типів пристроїв виводу зображень : моніторів (проекторів) і принтерів.

Для найбільш точного подання кольору використовуються системи представлення кольору з більшим, ніж три, числом компонент. Це дозволяє максимально точно кодувати спектр відбиття або випромінювання джерела світла.

1) **Система RGB** (Red, Green, Blue – червоний, зелений, синій) – адитивна кольорова модель, яка описує спосіб отримання кольору. Вибір основних кольорів обумовлений особливостями сприйняття кольору людським оком. Колірна модель RGB знайшла широке застосування в техніці. У телевізорах, моніторах і проекторах застосовуються три електронні гармати (світлодіода або світлофільтрів) для червоного, зеленого і синього каналів. Кольори в цій моделі отримують за рахунок додавання випромінювання до чорного кольору.

Наприклад, якщо колір екрана, підсвіченого кольоровим прожектором, позначити як $rgb1$, а колір того ж екрана, освітленого іншим прожектором, $rgb2$, то при освітленні двома прожекторами колір екрана буде позначатися як $r1 + r2, g1 + g2, b1 + b2$.

Зображення в даній колірній моделі складається з трьох каналів. Усі кольори, доступні в даній моделі, можна отримати при змішуванні основних кольорів моделі в різних пропорціях.

Наприклад, якщо змішати в рівних пропорціях синій (B) і червоний (R), вийде пурпурний (M magenta), змішавши зелений (G) і червоний (R), отримаємо жовтий (Y yellow), а змішавши зелений (G) і синій (B) – небесний колір (C cyan). При змішуванні всіх трьох колірних компонент виходить білий колір (W white).

Усі кольори, які можна передати за допомогою цієї моделі, знаходяться всередині одиничного колірного куба (рис. 2.2).

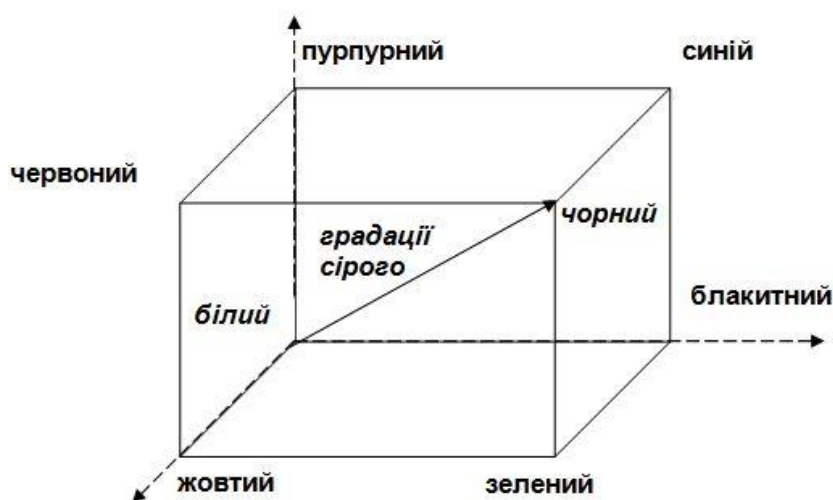


Рисунок 2.2 – Колірний куб для системи RGB

2) Система CMY(K) (цмік) (Cyan, Magenta, Yellow, Key) – субтрактивна схема кодування кольору, яка використовується в поліграфії для друку. Модель CMY(K), як правило, має порівняно невелике колірне охоплення (у порівнянні з RGB). Робота цієї моделі (на відміну від RGB) заснована не на випромінюванні, а на *відбиванні та поглинанні світла*. Таким чином, для "роботи малюнка" CMY(K), необхідна наявність зовнішнього джерела випромінювання.

Оскільки CMYK застосовують в основному в поліграфії при кольоровому друкуванні, а папір та інші друковані матеріали є поверхнями, що відбивають світло, зручніше рахувати, скільки світла відбилося від поверхні. Таким чином, якщо віднімати від білого послідовно три первинні кольори моделі RGB, вийде трійка кольорів CMY. Така модель є *субтрактивною* – від білого віднімаються первинні кольори RGB.

У CMYK використовуються чотири кольори, перші три в аббревіатурі названі за першою літерою кольору, а четвертим використовується чорний колір. Кольори називають так: небесно-блакитний (ціан), пурпурний

(маджента), жовтий і чорний.

Колір в СМУК залежить не тільки від спектральних характеристик барвників та способу їх нанесення, але й їх кількості, характеристик паперу та інших факторів. Код кольору в СМУК є лише набором даних для друку і не визначає колір однозначно і точно.

Використовується кілька стандартизованих процесів офсетного друку для крейдованого і некрейдованого паперу. Для кожного з цих процесів розроблені стандартизовані папір і фарби, створені відповідні колірні моделі СМУК, які використовуються в процесах відтворення кольору.

Чорний колір теоретично можна отримувати змішуванням у рівній пропорції пурпурного, небесного і жовтого барвників, але з певних причин (чистота кольору, розмокання паперу) такий підхід не застосовується.

Основні причини використання додаткового чорного кольору:

- а) на практиці змішування реальних пурпурного, блакитного і жовтого кольорів дає брудно-коричневий або брудно-сірий колір;
- б) при виведенні дрібних чорних деталей без використання чорного кольору (змішуванням) вони виходять розмитими;
- в) змішування великої кількості пурпурного, блакитного і жовтого чорнил в одній точці призводить до розмокання і деформації паперу, збільшує час висихання;
- г) чорний пігмент (зазвичай використовується сажа) дешевший від кольорових.

Таким чином, код кольору в СМУК не може сам по собі описати колір. Це набір апаратних даних, що використовуються в друкарському процесі для формування зображення. На практиці отриманий при друку колір буде обумовлений особливостями конкретного друкарського процесу.

Колірна модель RGB має у багатьох тонах ширше колірне охоплення (може подати більш насичені кольори), ніж типове охоплення кольорів СМУК, тому іноді якісні зображення в RGB тьмяніють при поданні в моделі СМУК.

Для отримання уявлення про колір, заданий у колірній моделі СМУК, застосовують колірні профілі, які пов'язують значення апаратних даних з реальним кольором, вираженим, як правило, у колірних моделях XYZ або LAB.

3) **Модель HSB/HSV** – основні компоненти: відтінок (H), насиченість (S) і яскравість (B або V). Колірне охоплення даної моделі можна подати у вигляді перевернутої піраміди або конуса, як показано на рис.2.3. Вершини основи піраміди відповідають чистим основним

кольорам (червоному, жовтому, зеленому, кольору морської хвилі, синьому і фіолетовому). При їх змішуванні один з одним у різних пропорціях (у просторі це будуть лінійні комбінації відповідних векторів) точка, яка відповідає кольору, переміщується по основі піраміди. Змішуючи протилежні кольори (наприклад, жовтий і синій), можна отримати білий.

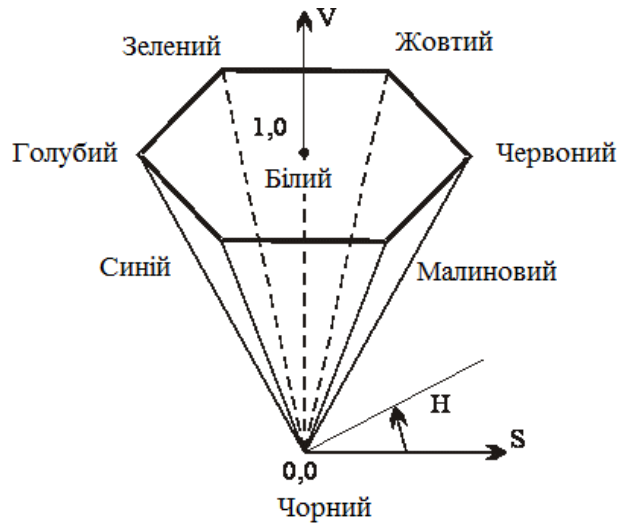


Рисунок 2.3 – Колірне охоплення моделі HSB (V)

Приклад, зручний для запам'ятовування послідовності кольорів у вершинах піраміди (проти годинникової стрілки): "Чимало Охочих Жадібно Знати Бажають Де Сидить Фазан" (виключаємо Помаранчевий колір).

2.3 Апаратно-незалежне подання кольору

1) **Кольороворізнисна модель Yuv** має більше колірне охоплення, ніж модель RGB. Особливістю цієї моделі є те, що яскравість відокремлена від колірних компонент і може оброблятися незалежно. Колірні компоненти u і v – це різновиди основних кольорів моделі RGB.

Під аббревіатурою Yuv ховається кілька подібних моделей, які відрізняються сферою застосування і методом перетворення з моделі RGB. Моделі групи Yuv застосовуються при стисненні відео та фотозображень і в телевізійних системах PAL, SECAM, NTSC. Кольорові моделі $YCbCr$ і $YPbPr$ є варіаціями Yuv з іншими вагами для u і v . $YPbPr$ застосовується для опису аналогових сигналів (переважно в телебаченні), а $YCbCr$ – для цифрових. У телевізійній системі SECAM використовується модель $YDbDr$, а для NTSC – модель YIQ .

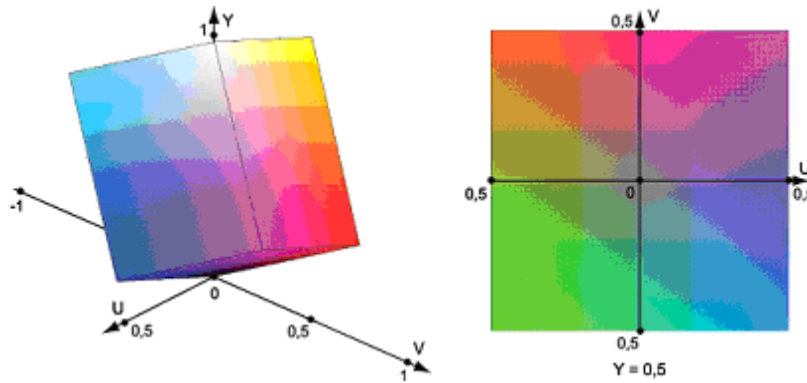


Рисунок 2.4 – Колірне охоплення моделі Yuv

Як видно з рисунку 2.4, колірний простір Yuv виходить шляхом поєднання діагоналі куба RGB з віссю Y.

2) **Модель XYZ** – це еталонна колірна модель, задана математично. Модель була введена організацією CIE (International Commission on Illumination – Міжнародна комісія з освітлення) в 1931 р. Модель XYZ є основною для всіх інших колірних моделей, що використовуються в технічній галузі.

Модель основана на реакції людського ока на спектр RGB і влаштована так, щоб точно визначати сприйняття кольору людиною. Ця модель використовується для того, щоб визначати апаратно-незалежні кольори, які підлягають правильному наведенню пристроями будь-якого типу: сканерами, моніторами та принтерами.

Система часто подається у вигляді двомірного графіка, який за формою схожий на вітрило. Червоні компоненти кольору витягнуті уздовж осі X координатної площини (горизонтально), а зелені – витягнуті уздовж осі Y (вертикально). При такому способі подання кожному кольору відповідає певна точка на координатній площині. Спектральна чистота кольорів зменшується в міру того, як точка переміщується координатною площиною вліво. Особливістю цієї моделі є те, що в ній не враховується яскравість.

Для усунення впливу освітлення на процес корекції кольору зображень модель XYZ перетвориться в модель xyz, в якій один параметр визначає інтенсивність освітлення, а два інші параметри визначають колір (параметри хроматичності). Ця модель найчастіше подається у вигляді $x = X / (X + Y + Z)$, $y = Y / (X + Y + Z)$, $z = Z / (X + Y + Z)$.

Параметр x являє собою хроматичність червоний/зелений (red/green), а параметр y – хроматичність жовтий/синій (yellow/blue). Недоліком моделі XYZ є неоднорідність похибки спостереження кольору в різних ділянках колірної діапазону.

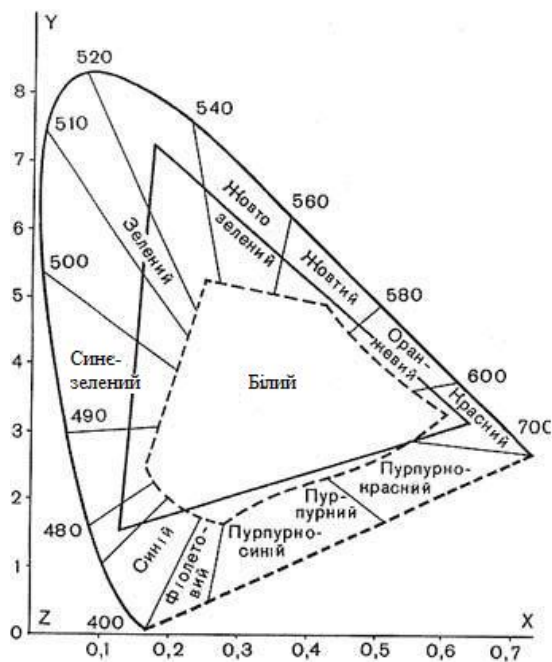


Рисунок 2.5 – Колірне охоплення моделі XYZ

3) **Модель Lab** – універсальна модель опису кольору (удосконалення моделі XYZ). Ця модель є апаратно-незалежною. З її допомогою одні й ті ж кольори відображаються незалежно від особливостей пристрою відображення, який використовується для виведення зображень. Колірний діапазон цієї моделі перекриває колірні діапазони RGB і CMYK. У межах моделі Lab будь-який колір визначається набором з трьох параметрів: L – світність (luminosity), а – кольоровість (хроматична) від зеленого до червоного кольору (green/magenta) і b – кольоровість від синього до жовтого кольору (blue/yellow).

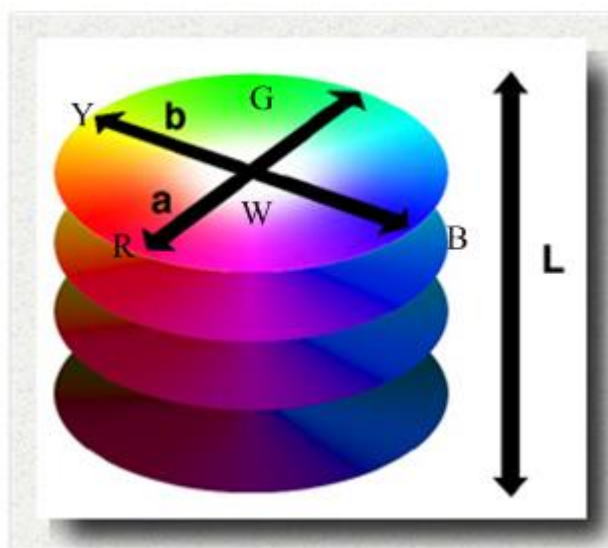


Рисунок 2.6 – Колірне охоплення моделі Lab

Модель Lab отримана нелінійним перетворенням моделі XYZ. Параметри моделі Lab розраховуються за допомогою нелінійних перетворень параметрів моделі XYZ таким чином, щоб забезпечити однорідність помилки спостереження колірних параметрів. У порівнянні з колірною моделлю XYZ кольори Lab більш сумісні з кольорами, які сприймаються людським оком.

2.4 Перетворення кольору між різними колірними моделями

У комп'ютерній графіці колір подається у вигляді вектора, кожен з елементів якого показує координату по відповідній осі системи кодування кольору. Найбільш часто використовується система RGB, для якої кожен елемент вектора показує рівень випромінювання за основними кольорами.

Вектор можна подати у вигляді набору чисел, зазвичай з плаваючою комою, при цьому значення 0 відповідає відсутності випромінювання, а 1 – максимальному випромінюванню. Також часто зустрічається запис одним числом, коли в групах біт числа розташовані рівні випромінювання. У такому випадку число записується в 16-річній системі, кожний колірний канал представлений 8 бітами (2-ма цифрами).

Для перекладу з такого подання у векторний вигляд необхідно виділити колірні канали та розділити їх значення на максимально можливе (255 для 8 біт на канал).

Індивідуальні завдання

Згідно з варіантом з табл. 2.2 отримати код кольору в системі RGB за його назвою. Далі перетворити колір з RGB в YUV, HSV, CMY, CMYK, а потім назад з YUV, HSV, CMY, CMYK в систему RGB (табл. 2.3).

Таблиця 2.2 – Кольори для кодування

№ п/п	Колір
1	Білий
2	Морквяний
3	Кукурудзяний
4	Хакі
5	Аквамариновий
6	Темно-синій
7	Червоний
8	Світло-зелений
9	Сірий
10	Чорний
11	Коричневий
12	Жовтий
13	Ціан
14	Пурпурний
15	Помаранчевий
16	Фіолетовий
17	Рожевий
18	Темно-червоний
19	Бурштиновий
20	Бронзовий

Компоненти кольору необхідно ставити в діапазоні від 0 до 1, для цього код кожної компоненти необхідно розділити на 255.

Таблиця 2.3 – Формули для перетворення

RGB -> YUV	YUV -> RGB
$Y = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B$ $U = 0,492 \cdot (B - Y) =$ $= -0,147 \cdot R - 0,289 \cdot G + 0,436 \cdot B$ $V = 0,877 \cdot (R - Y) =$ $= 0,615 \cdot R - 0,515 \cdot G + 0,100 \cdot B$	$R = Y + 1,140 \cdot V$ $G = Y - 0,395 \cdot U - 0,581 \cdot V$ $B = Y + 2,032 \cdot U$
RGB -> HSV	HSV -> RGB
$V = \max (R, G, B)$ $M = \min (R, G, B)$ $d = V - M$ $h = (d \neq 0) ? \infty :$ $(R \neq V) ? (G - B) / d :$ $(G \neq V) ? 2 + (B - R) / d :$ $4 + (R - G) / d$ $S = (d \neq 0) ? 0 : V / d$ $H = 360 \cdot (h < 0) + 60 \cdot h$	<p>Знаходимося на осі симетрії ($S \equiv 0$) $R = G = B = V$, інакше</p> $\text{sector} = \text{floor} (H / 60)$ $\text{frac} = H / 60 - \text{sector}$ $T = V \cdot (1 - S)$ $P = V \cdot (1 - S \cdot \text{frac})$ $Q = V \cdot (1 - S \cdot (1 - \text{frac}))$ <p>в залежності від значення sector:</p> <ol style="list-style-type: none"> 0) $R = V; G = Q; B = T$ 1) $R = P; G = V; B = T$ 2) $R = T; G = V; B = Q$ 3) $R = T; G = P; B = V$ 4) $R = Q; G = T; B = V$ 5) $R = V; G = T; B = P$
RGB -> CMY	CMY -> RGB
$C = 1 - R$ $M = 1 - G$ $Y = 1 - B$	$R = 1 - C$ $G = 1 - M$ $B = 1 - Y$
CMY -> CMYK	
$K = \min (C, M, Y)$ $C = C - K$ $M = M - K$ $Y = Y - K$	

Контрольні питання

1. Що таке світло?
2. Що таке колір?
3. Визначення колірного простору.
4. Що таке колірна температура?
5. Для чого використовується колірна температура?
6. Дані зображення в колірній моделі.
7. У чому різниця між адитивною і субтрактивною моделями відображення кольору?
8. Системи RGB і CMY, призначення та відмінності.
9. Графічне відображення колірного простору моделі RGB.
10. Як перейти від адитивної системи до субтрактивної?
11. Для чого в системі CMYK додана 4-та компонента?
12. Система HSB, основні відмінності від RGB.
13. Графічне відображення колірного простору моделі HSB.
14. Апаратно-незалежні моделі кодування кольору.
15. Кольороворізнисна модель кодування кольору.
16. Графічне відображення колірного простору моделі YUV.
17. Система Lab, сфера застосування.
18. Графічне відображення колірного простору моделі Lab.
19. Особливості перетворення кольору між різними системами.
20. Перетворення з RGB в YUV, HSV, CMY, CMYK.
21. Перетворення з YUV, HSV, CMY, CMYK в RGB.

3. ТРИВИМІРНА ГРАФІКА

Розташування будь-якого об'єкта можна визначити за допомогою системи координат. У переважній більшості комп'ютерних програм для створення тривимірної графіки використовується Декартова система координат. У більшості додатків, які використовують тривимірну графіку (3D графіку), об'єкти складаються з безлічі багатокутників, розміщених таким чином, що створюється їх реалістичний образ. Сотні або тисячі багатокутників, необхідних для єдиного 3D об'єкта, утворюють величезний масив даних, які треба створити і якими необхідно управляти.

Уся кількість 3D об'єктів, джерел освітлення та інших допоміжних об'єктів утворює віртуальний простір моделювання – сцену.

Тривимірне зображення відрізняється від плоского (двовимірного) побудовою геометричної проекції сцени, зазвичай на екрані комп'ютера, за допомогою спеціалізованих програмних засобів. При цьому сцена може як відповідати об'єктам з реального світу, так і бути повністю абстрактною.

Для отримання зображення тривимірних об'єктів потрібно зробити наступні кроки:

- 1) моделювання – створення математичної моделі сцени і об'єктів у ній;
- 2) рендеринг (візуалізація) – проекція й отримання растра відповідно до обраної моделі освітлення;
- 3) постобробка – усе, що відбувається після основних дій з побудови зображення (ефекти постобробки накладаються на растрове зображення).

3.1 Відображення моделей об'єктів у 3D графіці

У всій множині об'єктів, що утворюють сцену, можна виділити такі основні види:

- а) геометричний опис об'єктів і зв'язків між ними;
- б) матеріали – набір параметрів, які описують взаємодію поверхонь об'єктів з джерелами світла;
- в) джерела світла;
- г) віртуальні камери, які визначають вибір точки і кута побудови проекції;
- д) сили і вплив, які використовуються для фізичного моделювання взаємодії об'єктів та їх частин;
- е) додаткові ефекти (атмосферні явища, різні процеси: дим, полум'я, пил).

Завданням моделювання є створення математичної моделі об'єктів

і розміщення їх на сцені за допомогою геометричних перетворень.

Кожна модель створюється з використанням графічних примітивів:

1) **Точки і відрізки** (рис. 3.1) – базові абстрактні примітиви, що використовуються для опису полігонів (трикутників), з яких складаються об'єкти.

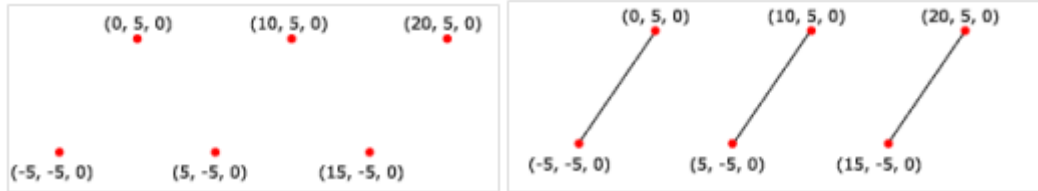


Рисунок 3.1 – Абстрактні примітиви

Точки і відрізки практично ніколи не візуалізуються, а використовуються як вершини і ребра фігур.

2) **Грані (трикутники) і полігони** (рис. 3.2) – основні примітиви, з яких створюються 3D-об'єкти. Використовуються в геометричних перетвореннях, накладаннях матеріалів, фізичному моделюванні.

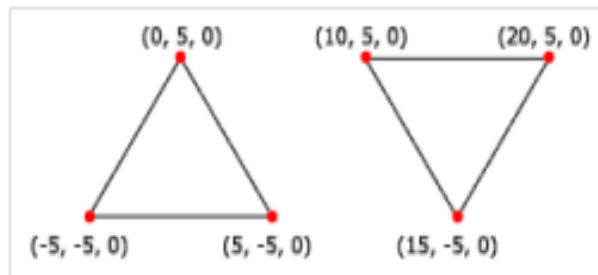


Рисунок 3.2 – Основні примітиви

Важливою характеристикою кожного трикутника є нормаль. Нормаль розраховується як векторний добуток двох ребер p та q трикутника за формулою (3.1).

$$pxq = \begin{pmatrix} i & j & k \\ p_x & p_y & p_z \\ q_x & q_y & q_z \end{pmatrix} = (p_y q_z - p_z q_y) \cdot i + (p_z q_x - p_x q_z) \cdot j + (p_x q_y - p_y q_x) \cdot k \quad (3.1)$$

3) **3D-об'єкти** (рис. 3.3) конструюються з полігонів (трикутників). Являють собою апроксимацію (спрощену модель) реальних або абстрактних об'єктів.

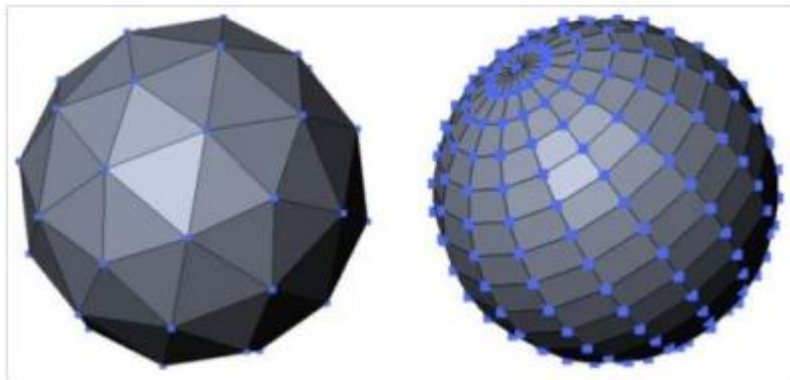


Рисунок 3.3 – Конструювання 3D-об'єктів за допомогою примітивів

По суті 3D-об'єкт являє собою полігональну сітку – сукупність вершин, ребер і граней, які визначають його форму. Гранями зазвичай є прості опуклі багатокутники (полігони). В основному це трикутники (рідко чотирикутники), оскільки це спрощує процес рендерингу і дозволяє легко реалізувати його апаратно.

3.2 Створення моделі об'єкта

Для створення моделі об'єкта необхідно задати положення його вершин, а також указати, як вони з'єднуються в межі (полігони).

Вершина – це позиція разом з іншою (необов'язковою) інформацією, такою як колір, нормаль до поверхні, текстурні координати.

Ребро – це з'єднання між двома вершинами.

Грань – це замкнутий набір ребер. Більшість апаратних засобів рендерингу підтримує межі з трьома сторонами.

Полігон – це набір з декількох граней.

Полігональна сітка як правило подається набором вершин і списком граней, кожен елемент якого являє собою трійку номерів (індексів) у наборі вершин. Таким чином, узявши вершини з номерами з трійки, можна отримати координати трикутника, який утворює грань.

Для кожного трикутника розрізняють передню (видиму) і задню сторони. Видимість межі визначається порядком обходу вершин – за годинниковою стрілкою або проти. Грані з одним з порядків обходу вважаються видимими. Таким чином, трійка індексів може починатися з будь-якої вершини, що входить до неї, але повинна бути задана в певному порядку.

Приклад задання трійок індексів для фігури показаний на рис. 3.4.

Список граней

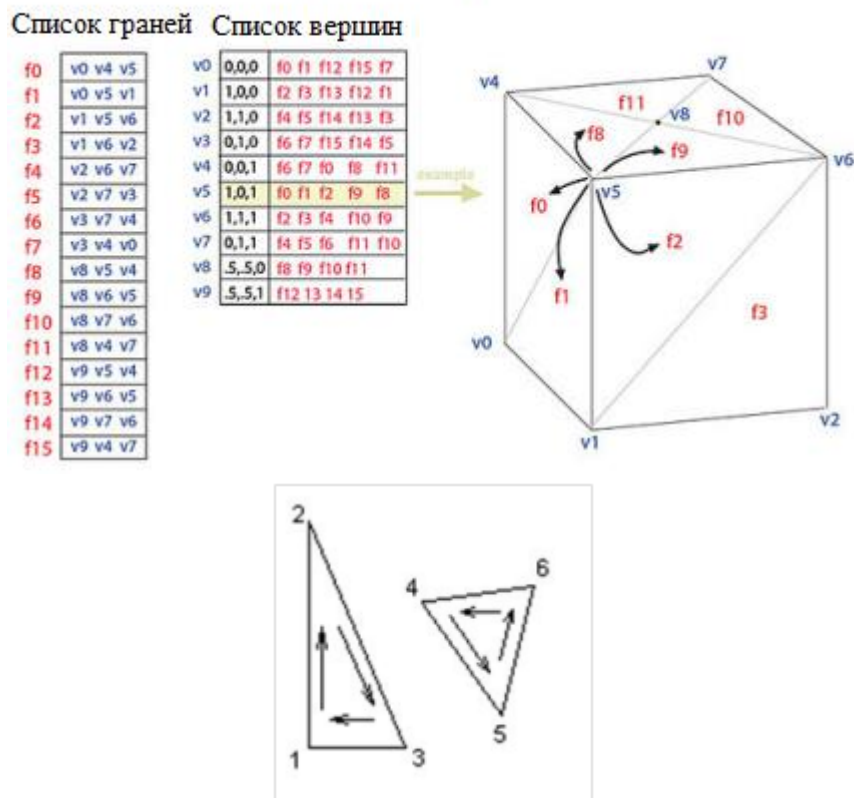


Рисунок 3.4 – Приклад задання списку граней для 3D об'єкта

Важливим завданням при описі фігури є розрахунок нормалі до апроксимованої поверхні, який необхідно провести для кожної вершини. Така нормаль використовується в подальшому при розрахунку освітленості поверхні об'єкту. Для розрахунку нормалі необхідно:

- розрахувати нормалі до поверхні кожного трикутника;
- підсумувати нормалі всіх трикутників, яким належить вершина;
- нормалізувати отримані нормалі в кожній вершині.

Якщо апроксимована поверхня має розрив (або злам) в якомусь місці, то необхідно продублювати вершини, що знаходяться в місці розриву, зберігши їхні координати, але рознести їх номери по різних гранях (трійках індексів).

Для визначення області безперервності поверхні в пакетах 3D моделювання використовуються групи згладжування.

Групи згладжування (Smoothing Groups) – це групи, які утворюють трикутні грані (рис. 3.5). Для граней, що входять до однієї групи згладжування, нормаль у точках контакту (загальних вершинах) "усереднюється". Як наслідок, межі 3D об'єкту відтіняються так, що переходи між ними здаються гладкими, і поверхня, яку вони апроксимують, не виглядає гранованою.

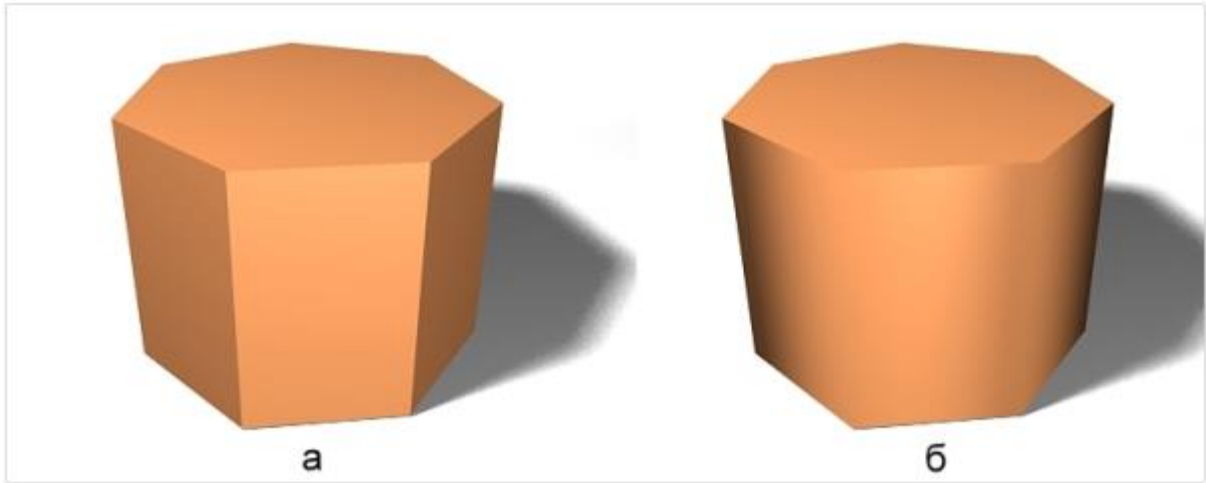


Рисунок 3.5 – Задання зламу і плавної поверхні при однаковій геометрії об'єкта

Насправді це лише ілюзія, якщо подивитися на силует, він буде зламаним (рис. 3.6). Але таке згладжування грає важливу роль у доданні поверхням об'єкта більш реалістичного зовнішнього вигляду.

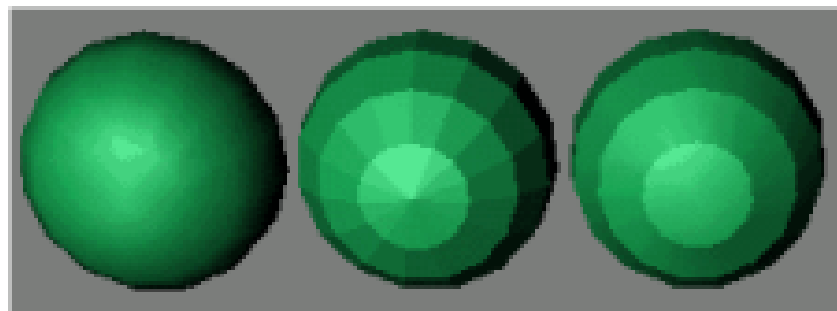


Рисунок 3.6 – Різні групи згладжування при однаковій геометрії моделі

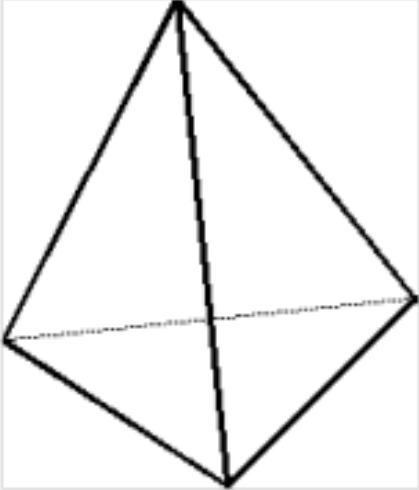
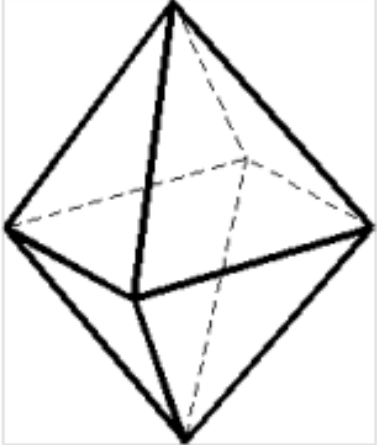
Наприклад, у сфери всі грані сітки належать до однієї групи згладжування. У циліндра грані бічної поверхні належать до однієї групи згладжування, а грані основи – до іншої.

Індивідуальні завдання

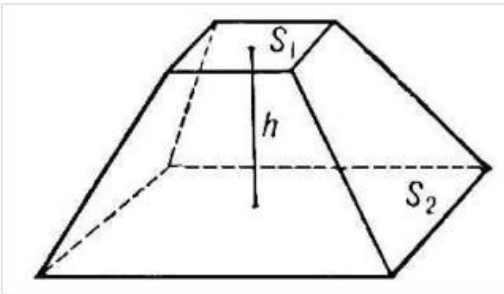
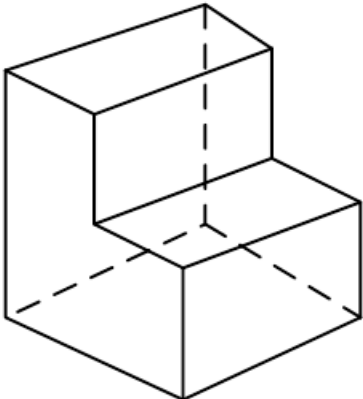
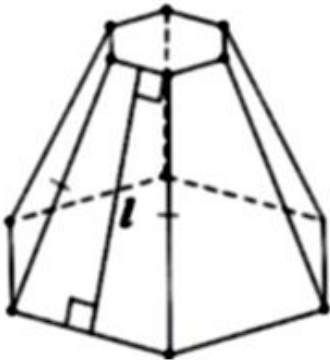
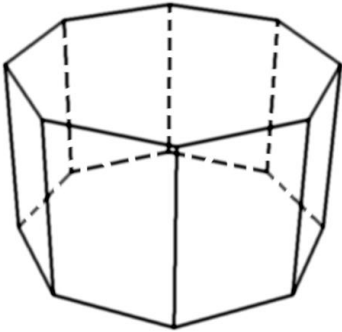
1. Необхідно створити модель тривимірної фігури у вигляді сітки трикутників. Для чого потрібно:

- а) задати координати вершин;
 - б) розрахувати кількість трикутників, якими буде подана фігура;
 - в) накреслити 3 проєкції фігури (вид зверху, праворуч і спереду);
 - г) скласти список трійок індексів вершин, які описують трикутники.
- Список фігур наведено в табл. 3.1.

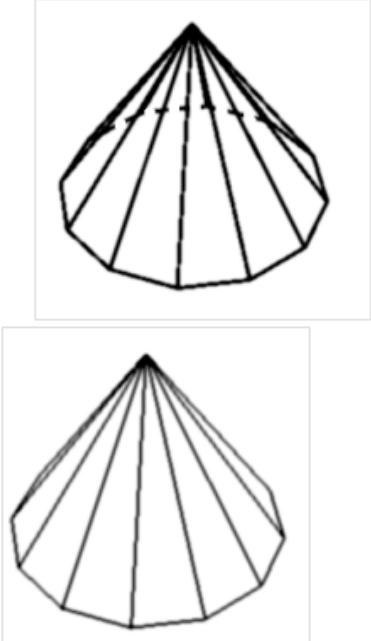
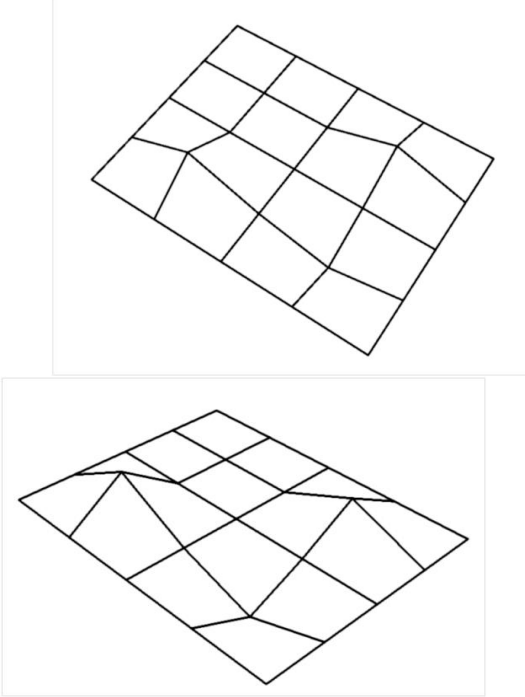
Таблиця 3.1 – Варіанти фігур

№п/п	Зображення фігури	Характеристики фігури
1		<p>Тетраедр. 4 грані, 4 вершини і 6 ребер. Грань – трикутник.</p>
2		<p>Октаедр. 8 граней, 12 ребер, 6 вершин (на кожній сходяться 4 ребра). Грань – трикутник.</p>

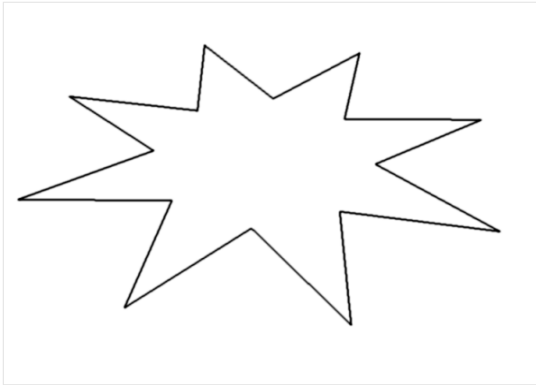
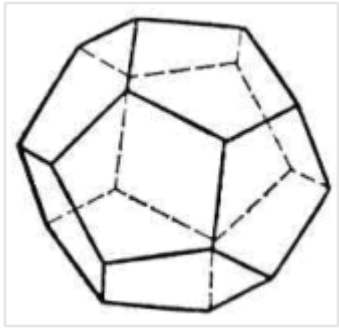
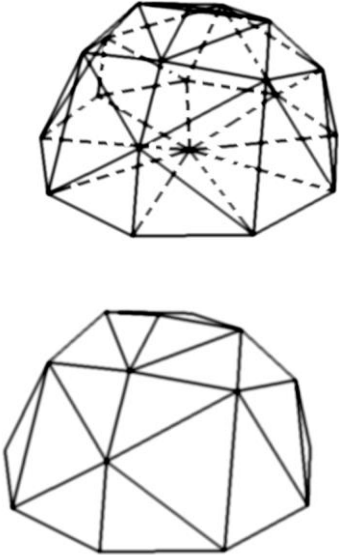
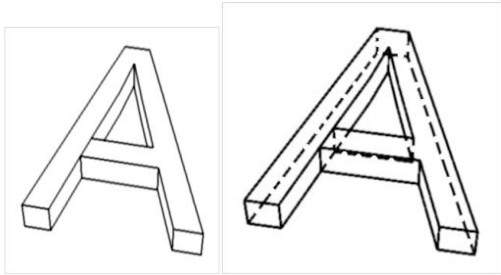
Продовження таблиці 3.1

№п/п	Зображення фігури	Характеристики фігури
3		<p>Усічена піраміда. 6 граней, 12 ребер, 8 вершин. <i>Грань –</i> <i>чотирикутник.</i> На кожній вершині сходяться 3 ребра.</p>
4		<p>Сходинка. 8 граней, 16 ребер, 12 вершин.</p>
5		<p>Усічений конус. 8 граней, 18 ребер, 12 вершин. В основі – шестикутник.</p>
6		<p>Циліндр. 10 граней, 24 ребра, 16 вершин. В основі – восьмикутник.</p>

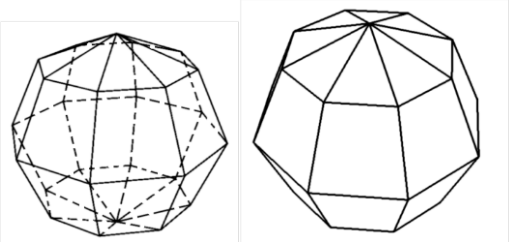
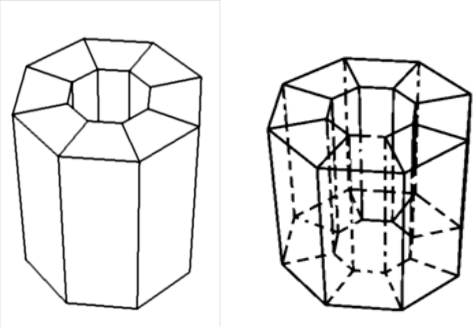
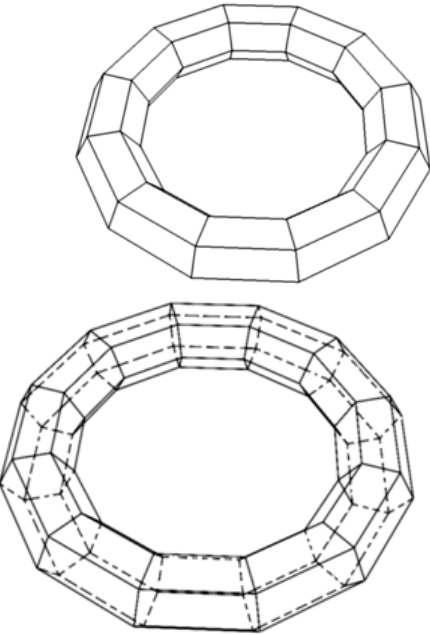
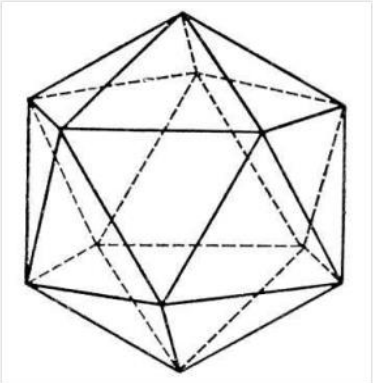
Продовження таблиці 3.1

№п/п	Зображення фігури	Характеристики фігури
7		<p>Конус. 13 граней, 24 ребра, 13 вершин. В основі – дванадцятикутник.</p>
8		<p>Поверхня. 16 граней, 25 вершин, 40 ребер. Вершини, що знаходяться вище або нижче площини, віддалені від неї (по нормалі) на довжину ребра грані.</p>

Продовження таблиці 3.1

№п/п	Зображення фігури	Характеристики фігури
9		<p>Восьмикутна зірка. 14 граней, 29 ребер, 16 вершин. Усі вершини знаходяться в одній площині.</p>
10		<p>Додекаедр. 12 граней, 30 ребер, 20 вершин. <i>Грань –</i> <i>п'ятикутник.</i> На кожній вершині сходяться 3 ребра.</p>
11		<p>Півсфера. 36 граней, 20 вершин. В основі – дев'ятикутник.</p>
12		<p>Буква А. 13 граней, 39 ребер, 26 вершин.</p>

Продовження таблиці 3.1

№п/п	Зображення фігури	Характеристики фігури
13		<p>Сфера. 48 граней, 26 вершин, 72 ребра. Бічна грань – чотирикутник.</p>
14		<p>Труба. 64 грані, 96 ребер, 32 вершини. В основі – восьмикутник.</p>
15		<p>Тор. 144 грані, 216 ребер, 72 вершини. <i>Переріз – шестикутник.</i> Зовнішній і внутрішній радіуси – дванадцятикутники.</p>
16		<p>Ікосаедр. 20 граней, 30 ребер, 12 вершин. <i>Грань – трикутник.</i> На кожній вершині сходяться 5 ребер.</p>

Контрольні питання

1. Які системи координат використовуються в тривимірній графіці?
2. Що таке тривимірна сцена?
3. Чим тривимірне зображення відрізняється від двовимірного?
4. Що необхідно зробити для отримання зображення тривимірних об'єктів?
5. Які основні види об'єктів у сцені можна виділити?
6. Задача тривимірного моделювання?
7. Які примітиви використовуються для тривимірного моделювання?
8. Як розрахувати нормаль трикутника?
9. Що таке вершина?
10. Що таке ребро?
11. Як пов'язані вершина і ребро?
12. Як описати грань тривимірного об'єкта?
13. Як визначається видимість грані?
14. Що таке полігональна сітка?
15. Чому в КГ грані подають у вигляді трикутників або чотирикутників?
16. Що таке рендеринг?
17. Що необхідно зробити, якщо апроксимована поверхня має розрив в якомусь місці об'єкта?
18. Чим визначається видимість грані тривимірного об'єкта?
19. Як розрахувати нормаль до поверхні тривимірного об'єкта?
20. Що таке групи згладжування і для чого вони використовуються?
21. Що відбувається з нормаллю в точках контакту для граней, які входять в одну групу згладжування?

4. ПРОЦЕС ВІЗУАЛІЗАЦІЇ (РЕНДЕРИНГУ)

Процес перетворення математичної моделі сцени (просторової моделі) в плоске (растрове) зображення називається візуалізацією або рендерингом. Часто також необхідно створити відеопотік: набір послідовних кадрів із зображеннями сцени. Процес візуалізації триває безперервно, синтезуючи кадри з певною частотою. Усе це вимагає взаємодії з операційною системою. Приклад взаємодії для ОС Windows показаний в додатку А.

Візуалізація перетворює тривимірну векторну структуру даних в плоску матрицю пікселів. Цей крок вимагає зробити низку складних обчислень, особливо для реалістичних зображень складних об'єктів.

Найпростіший вид рендерингу – отримання контурів моделей на екрані комп'ютера за допомогою проєкції. У більшості випадків цього недостатньо, коли необхідно розрахувати взаємодію матеріалів поверхні об'єктів з падаючим на них світлом.

Спрощена послідовність дій, яку необхідно провести для отримання зображення 3D сцени на моніторі:

1. Отримати локальний простір кожного об'єкту.
2. Перехід до світового простору.
3. Перехід до простору спостерігача.
4. Відсікання невидимих поверхонь (трикутників або частин об'єктів).
5. Розрахунок освітлення.
6. Відсікання невидимих фрагментів зображення.
7. Проєкція (перехід до плоского векторного зображення).
8. Перехід до простіру порту перегляду.
9. Растеризація.

Перші кроки, які необхідно виконати, це перетворення з однієї системи координат в іншу. Зазвичай ці перетворення виконуються за допомогою матричних перетворень на графічному прискорювачі або центральному процесорі.

Локальний простір (local space), або простір моделювання (modeling space) – це та система координат, в якій описується об'єкт у вигляді списку трикутних граней. Локальний простір спрощує процес моделювання, оскільки створювати модель в її власній, локальній системі координат простіше, ніж вбудувати її безпосередньо в загальну сцену. Локальний простір дозволяє створювати моделі, не піклуючись про взаємне розташування, розмір або орієнтації щодо інших об'єктів сцени (рис. 4.1). Центр такого простіру обирається у центрі симетрії або мас

об'єкту.

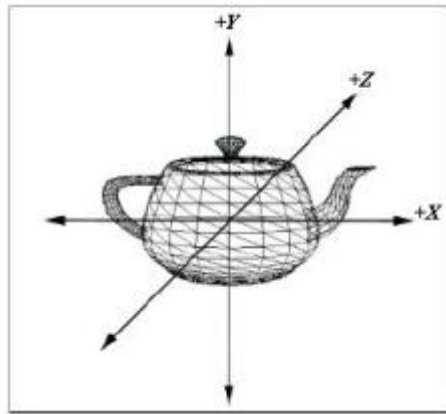


Рисунок 4.1 – Модель чайника, описана в локальній системі координат

4.1 Системи координат в 3D графіці

Для задання координат об'єктів і вершин примітивів використовується тривимірна *Декартова система координат*. Координати точок дозволяють описувати статичне положення точок у просторі. У комп'ютерній графіці застосовуються лівостороння і правостороння системи координат (рис. 4.2).

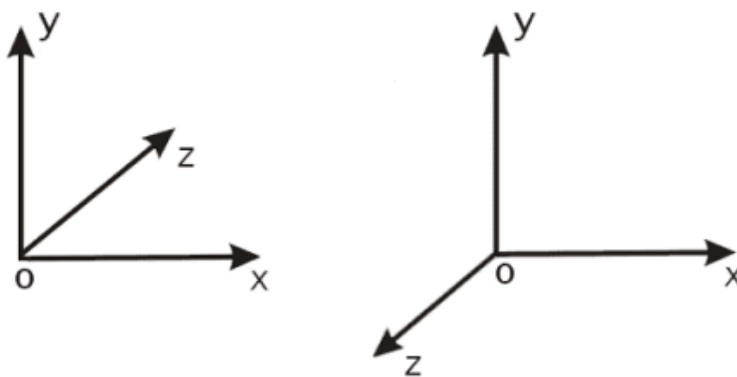


Рисунок 4.2 – Лівостороння і правостороння системи координат

Перетворення, пов'язані з системами координат, які можна виконувати в 3D графіці, зазвичай описуються квадратними матрицями розміром 4×4 .

Матрицею $m \times n$ називається прямокутний масив чисел, що складається з m рядків і n стовпців. Кількість рядків і стовпців визначає розмір матриці.

4.2 Перетворення координат

Для позначення координат вершин або напрямків використовуються матриці, які складаються з єдиного рядка або єдиного стовпця. Такі матриці називаються векторами.

Множення матриць – це найважливіша операція, яка постійно використовується в тривимірній комп'ютерній графіці. Саме множення матриць дозволяє здійснювати перетворення векторів і комбінувати кілька перетворень в одне.

Одиничною матрицею (identity matrix) називається квадратна матриця, усі елементи якої дорівнюють нулю, за винятком тих, що розташовані на головній діагоналі – ці елементи дорівнюють одиниці. Операція множення на одиничну матрицю не змінює вихідну матрицю - означає *відсутність перетворення*.

Процес використання матриць для проведення перетворень полягає в заданні елементів матриці розміром 4×4 таким чином, щоб вони описували необхідне перетворення або послідовність перетворень. Координати точки поміщаються у вектори розміром 1×4 , для останнього елемента w береться значення, рівне 1, що дозволяє коректно виконувати переміщення точки (якщо переміщення не потрібне, $w = 0$).

Доповнений *чотиривимірний вектор* називається однорідним вектором (homogenous vector) – такий вектор може описувати і точки, і власне вектори. Коли точка (x, y, z) записана у вигляді $(x, y, z, 1)$, вона фактично описує тривимірний простір як площину в чотиривимірному просторі (площина в чотиривимірному просторі є тривимірним простором, як і площина в тривимірному просторі є двовимірним простором). Таким чином, присвоюючи w яке-небудь інше значення, можна переміститися з площини $w = 1$. Щоб повернутися на цю площину, яка відповідає використовуваному тривимірному простору, потрібно виконати зворотну проєкцію шляхом ділення кожної компоненти на w .

Матриці розміром 4×4 дозволяють здійснити всі необхідні перетворення: переміщення, перспективну проєкцію і відображення. Результатом перемноження матриці на вектор буде новий вектор.

Для того щоб виконати зворотне перетворення, необхідно взяти інверсну (зворотну) матрицю перетворення і помножити на неї вектор.

4.2.1 Масштабування

Операція масштабування полягає у множенні координати на певну величину, $x' = x * s$, якщо $|s|$ більше 1, то розмір об'єктів збільшується, якщо $1 > |s| > 0$, то зменшується (рис.4.3). Якщо $s < 0$, то об'єкт інвертується. При $s = 0$ розміри об'єкта стають рівними 0.

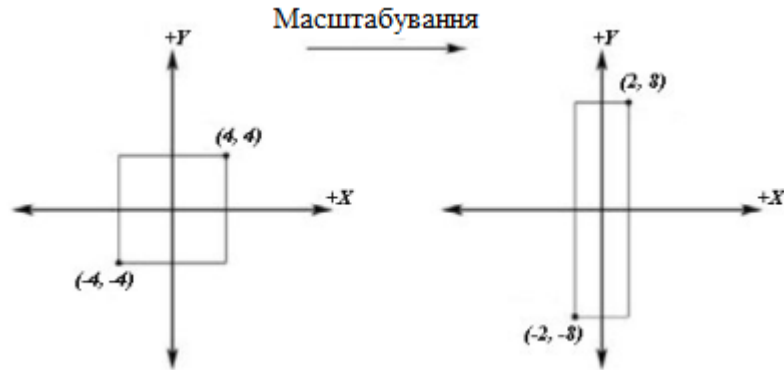


Рисунок 4.3 – Масштабування з коефіцієнтом 1/2 по осі X і коефіцієнтом 2 по осі Y.

Для масштабування вектора з коефіцієнтом s_x по осі X, коефіцієнтом s_y по осі Y і коефіцієнтом s_z по осі Z необхідно помножити його на матрицю (4.1):

$$S(q) = \begin{pmatrix} q_x & 0 & 0 & 0 \\ 0 & q_y & 0 & 0 \\ 0 & 0 & q_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.1)$$

Якщо значення s_x, s_y, s_z збігаються, таке масштабування називається *рівномірним*.

Щоб інвертувати матрицю масштабування, треба взяти зворотний дріб для кожного коефіцієнта масштабування за формулою (4.2).

$$S^{-1} = S\left(\frac{1}{q_x}, \frac{1}{q_y}, \frac{1}{q_z}\right) = \begin{pmatrix} \frac{1}{q_x} & 0 & 0 & 0 \\ 0 & \frac{1}{q_y} & 0 & 0 \\ 0 & 0 & \frac{1}{q_z} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.2)$$

4.2.2 Переміщення

Операція переміщення полягає в додаванні до координати певної величини $x' = x + t$. Якщо t позитивне, точка переміщується в напрямку осі, якщо негативне – проти (рис. 4.4). Якщо $t = 0$, переміщення не відбувається.

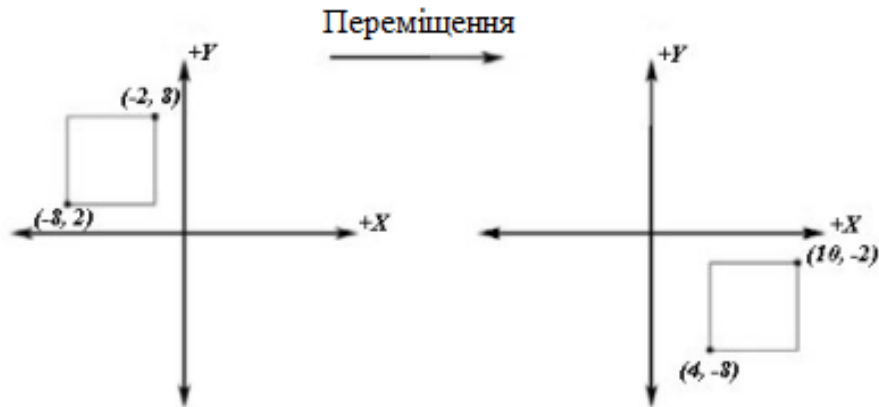


Рисунок 4.4 – Операція переміщення на 12 одиниць по осі X і на -10 одиниць по осі Y

Таким чином, для того щоб перемістити вектор $(x, y, z, 1)$ на t_x одиниць по осі X , t_y одиниць по осі Y і t_z одиниць по осі Z , необхідно помножити його на матрицю (4.3).

$$T(q) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p_x & p_y & p_z & 1 \end{pmatrix} \quad (4.3)$$

Інверсією матриці зсуву є матриця, в якій три значення в останньому рядку помножені на -1.

Таким чином, якщо збудувати матрицю в якій нема переміщення ($t = 0$) і нема масштабування ($s = 0$), буде отримана одинична матриця, яка задає відсутність перетворення.

4.2.3 Обертання

Операція обертання складається з 3-х незалежних операцій обертання по осях x , y , z , які здійснюються шляхом зв'язування двох координат за допомогою функцій \sin та \cos (рис. 4.5).

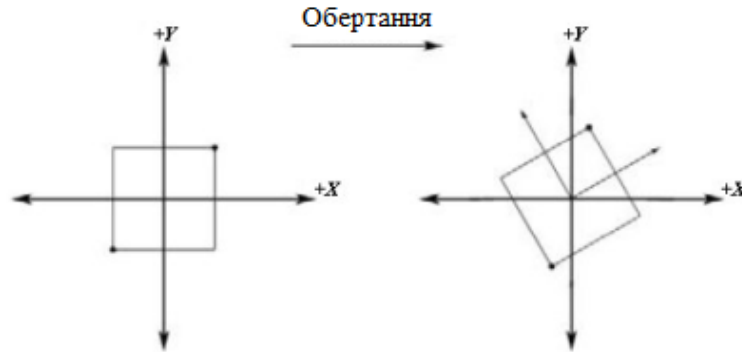


Рисунок 4.5 – Поворот на 30 градусів проти годинникової стрілки навколо осі Z

Для того щоб повернути вектор на θ радіан навколо осей X , Y або Z використовуються матриці (4.4), (4.5), (4.6).

$$X(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.4)$$

$$Y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.5)$$

$$Z(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.6)$$

Інверсією матриці обертання є результат транспонування підматриці 3×3 цієї матриці.

Для того щоб застосувати послідовні перетворення, необхідно скомбінувати всі матриці перетворень у потрібному порядку. Наприклад, масштабувати, повернути, потім перемістити об'єкт у потрібну позицію.

Ключовою перевагою використання матриць є можливість

використання множення матриць для комбінування декількох перетворень в одній матриці. Таким чином, замість того, щоб застосовувати до кожного вектора послідовність перетворень, можна скомбінувати всі перетворення в одній матриці шляхом перемноження їх матриць. Після цього помножити кожний вектор на єдину матрицю, що містить комбінацію всіх трьох перетворень. Завдяки цьому кількість виконуваних операцій множення вектора на матрицю значно скорочується, і так само скорочуються потрібні обчислювальні ресурси.

Порядок, в якому множаться матриці, повинен суворо відповідати порядку застосування окремих перетворень.

4.3 Світова система координат

Після того як усі моделі створені в своїй власній локальній системі координат, їх треба розташувати на сцені в єдиній, *глобальній або світовій системі координат* (world space). Об'єкти перетворюються з локального простору в світовий за допомогою процесу, названого світовим перетворенням (world transform).

Такий процес складається з операцій переміщення, обертання і масштабування, у результаті яких модель об'єкта розташовується так, як це необхідно. Світове перетворення задає взаємозв'язок між усіма об'єктами світу в частині їх розташування, розміру й орієнтації (рис. 4.6).

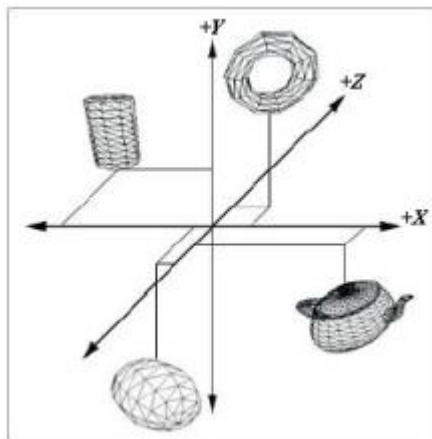


Рисунок 4.6 – Тривимірні об'єкти, описані у світовій системі координат

4.4 Система координат спостерігача і камера

У світовому просторі геометрія об'єктів і розташування спостерігача прив'язані до однієї системи координат. Однак проєкція та інші необхідні для візуалізації операції будуть досить складними, якщо спостерігач (камера) не займає певне місце розташування і належним чином не орієнтована.

Щоб спростити обчислення проєкції, камера переміщується в початок координат і повертається таким чином, щоб вона була спрямована вздовж напрямку осі Z (рис. 4.7).

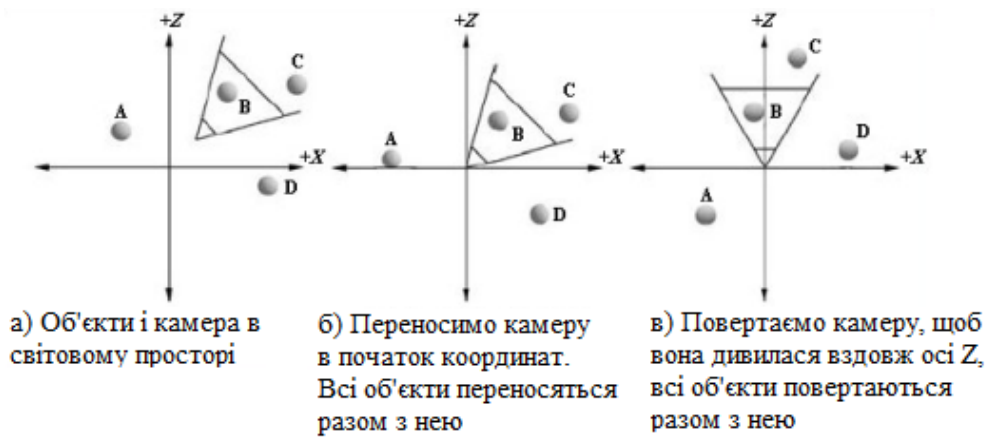


Рисунок 4.7 – Перетворення з світового у простір спостерігача

Таке перетворення створює *систему координат спостерігача* і називається перетворенням простору виду (view space transformation). Схема моделі камери, яка використовується в 3D графіці, показана на рис. 4.8.

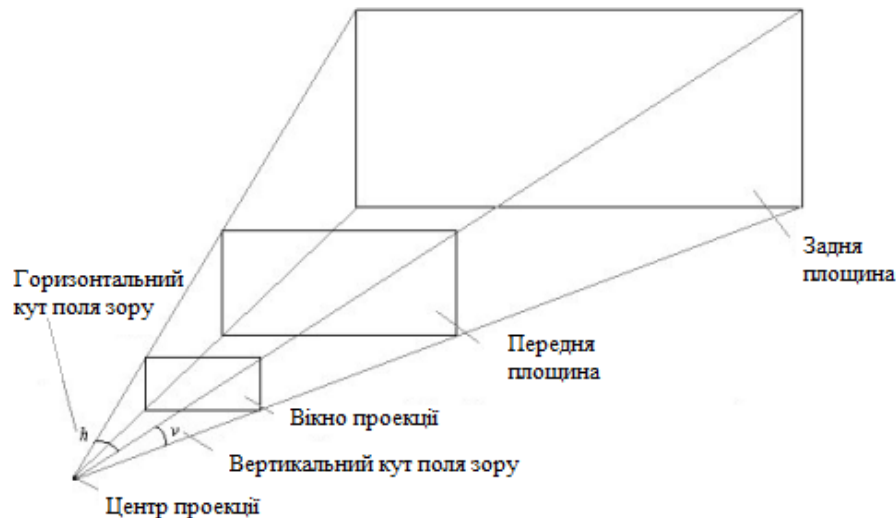


Рисунок 4.8 – Усічена піраміда, яка визначає область простору об'єкти в якому "бачить" спостерігач

Разом з камерою переміщуються і повертаються всі об'єкти, що утворюють сцену, тому відносний вид сцени залишається незмінним. Після цього перетворення об'єкти розташовані в просторі спостерігача (view space).

Камера визначає, яку частину світу може бачити спостерігач і, отже, для якої частини сцени необхідно створювати її двомірне зображення.

4.4.1 Відсікання

Область видимого простору являє собою усічену піраміду (frustum), яка складається з кута огляду по вертикалі (fov), зв'язаного з ним співвідношенням пропорцій екрана кута огляду по горизонталі (fov_x), двох площин, розташованих перпендикулярно осі Z на відстані Z_n і Z_f . Дані площини утворюють ближній і дальні плани відсікання (рис. 4.9).

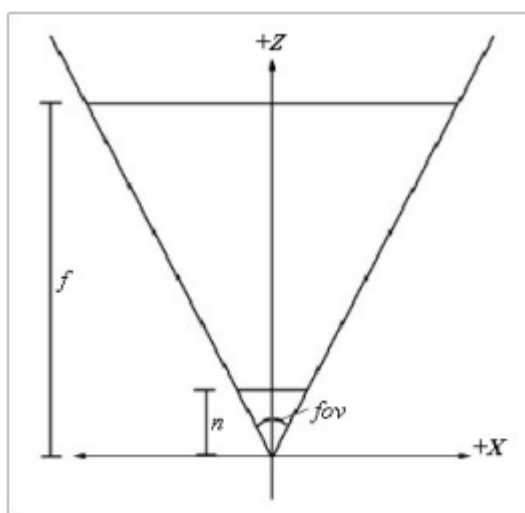


Рисунок 4.9 – Компоненти піраміди (frustum) видимого простору

Усі об'єкти та їх частини, які розташовані за межами піраміди, не буде видно на екрані, тобто їх не потрібно обробляти. Після перетворення в систему координат спостерігача (простір виду) проводиться відсікання зворотних граней і відсікання пірамідою видимості. На рис. 4.10 показаний об'єкт у просторі виду, лицьова сторона кожної грані якого позначена стрілкою.



Рисунок 4.10 – Об'єкт з фронтальними і зворотними полігонами

Фронтальні полігони приховують зворотні полігони, які знаходяться за ними і які можна виключити з подальшої обробки. Цей процес називається *видаленням невидимих граней* (backface culling). На рис. 4.11 показано той же об'єкт, але вже після видалення невидимих граней. Камера буде все одно показувати таку ж саму сцену, оскільки зворотні грані приховані і в будь-якому випадку їх не можна побачити з цього ракурсу.

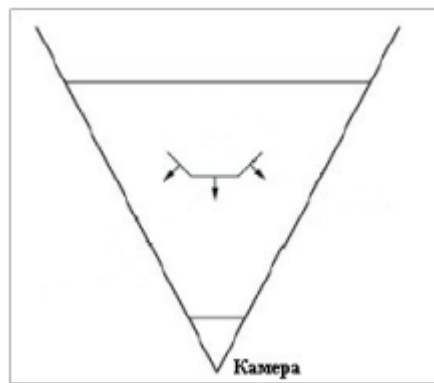


Рисунок 4.11 – Сцена після відсікання зворотних граней

Далі необхідно відсікти геометрію, яка знаходиться поза видимим простором. Такий процес називається *відсіканням* (clipping). Існує три варіанти розміщення трикутної грані щодо усіченої піраміди видимого простору:

1. Повністю всередині - якщо трикутник повністю знаходиться в середині видимого простору, він переходить на наступний етап.
2. Повністю зовні - якщо трикутник знаходиться повністю поза пірамідою видимого простору, він виключається з процесу подальшої обробки.
3. Частково всередині (частково зовні) - якщо всередині піраміди видимого простору знаходиться тільки частина трикутника, то він розбивається на дві частини. Частина, яка знаходиться всередині піраміди

видимого простору, залишається, а частина, яка знаходиться зовні – відкидається.

Усі три розглянуті варіанти зображені на рис. 4.12.

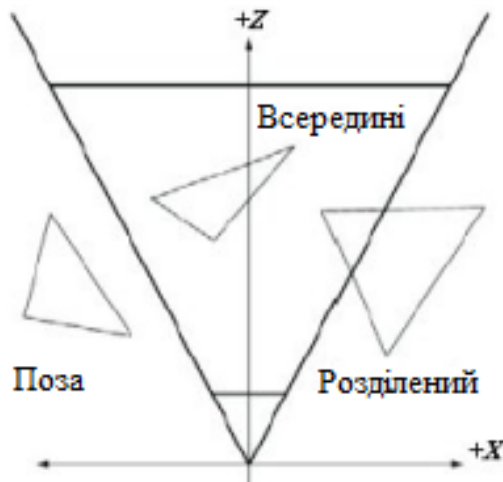


Рисунок 4.12 – Відсікання геометрії, яка перебуває поза видимим простором

4.4.2 Проеціювання

Після відсікання для простору виду залишається завдання отримання двомірного уявлення тривимірної сцени. Процес переходу від n -мірного простору до $(n-1)$ -мірного називається *проекцією* (projection).

Плоскі геометричні проекції діляться на два види: центральні та паралельні. Якщо центр проекції перебуває на кінцевій відстані від проекційної площини, то проекція центральна. Якщо центр проекції віддалений на нескінченність, то проекція паралельна (рис. 4.13).



Рис. 4.13 – Види проекцій:
а) – центральна, б) – паралельна

Для спрощення відображення, площина проекції (площина, в якій розташоване вікно проекції) і передня площина відсікання збігаються.

Центральна перспективна проекція призводить до візуального ефекту, подібного до того, який дає зорова система людини. При цьому спостерігається ефект *перспективного укорочення*, коли розмір проекції об'єкта змінюється обернено пропорційно відстані від центру проекції до об'єкта. У паралельних проекціях відсутнє перспективне укорочення, за рахунок чого зображення виходить менш реалістичним, і паралельні прямі завжди залишаються паралельними.

Цей тип проекції дозволяє подати тривимірну сцену як двовимірне зображення і розміщує отримані об'єкти у вікні проекції.

Вікно проекції (projection window) – це двовимірна область, на яку проектуються тривимірні об'єкти, що знаходяться всередині видимого простору, для створення двовимірного зображення, яке являє собою відображення тривимірної сцени з обраної позиції спостерігача. Вікно проекції визначено таким чином, що координати його правого верхнього кута $(1, 1)$, а координати лівого нижнього кута $(-1, -1)$.

4.5 Растеризація

Після перетворення вершин в екранні координати і їх розміщення на вікні проекції утворюється список двовимірних плоских трикутників. Фінальним етапом візуалізації є растеризація, яка відповідає за вирахування кольорів окремих пікселів, що утворюють трикутник (рис. 4.14).

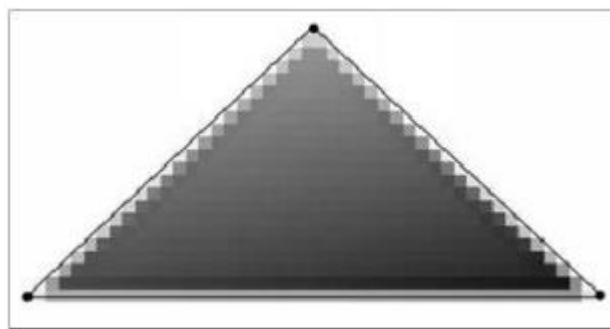


Рисунок 4.14 – Растеризація трикутника на екрані

На етапі растеризації також проводиться відсікання граней, які пройшли всі перевірки відсікання, але закриті іншими об'єктами або частинами об'єктів. Така перевірка проводиться попіксельно (пофрагментно) з використанням *буфера глибини (Z-буфера)*.

Буфер глибини (depth buffer) – це поверхня, яка містить не зображення, а інформацію про глибину (відстані по осі Z від спостерігача) окремих пікселів або їх фрагментів. Кожному пікселю в остаточному

зображенні відповідає елемент буфера глибини.

Так, якщо розмір растеризованого зображення 1920x1080 фрагментів, в буфері глибини має бути також 1920x1080 елементів.

На рис. 4.15 показано просту сцену, де одні об'єкти частково приховують інші, що знаходяться позаду них.

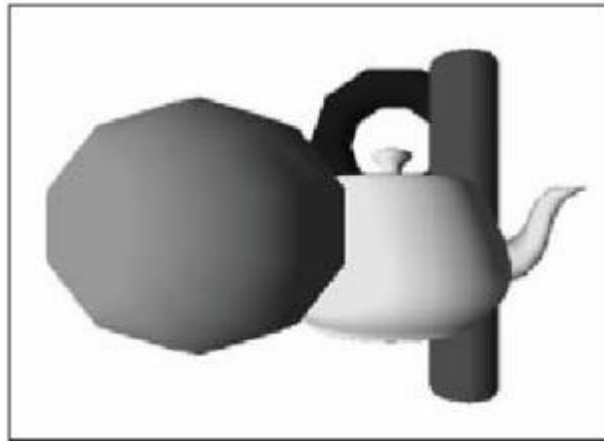


Рисунок 4.15 – Група об'єктів, які частково закривають один одного

Перед початком растеризації зображення Z -буфер заповнюється значенням максимального $Z = Z_f$, що відповідає дальній площині відсікання піраміди видності.

Щоб визначити, пікселі якого об'єкта знаходяться поверх усіх інших, використовується операція порівняння Z координати нового фрагмента з відповідним фрагментом, що знаходиться в Z -буфері. Якщо нове значення більше – фрагмент відкидається і значення в буфері кадру і Z -буфері не змінюються. У тому ж випадку, коли нове значення менше – фрагмент записується в буфер кадру і в Z -буфер. Таким чином, найближчий до камери піксель приховує всі інші пікселі, що знаходяться позаду нього.

Поведінка цієї операції може бути змінена за допомогою графічного АРІ, наприклад порівняння може бути обратним, або зовсім відключено.

Індивідуальні завдання

У таблиці 4.1 наведено вихідні дані для проведення матричного перетворення точки P .

Необхідно скласти матриці і провести перетворення, для чого потрібно виконати наступні кроки:

- 1) Скласти матрицю масштабування S .
- 2) Скласти матрицю зміщення T .
- 3) Скласти матриці обертання навколо кожної з осей: R_x , R_y , R_z , а потім розрахувати загальну матрицю обертання $R = R_x \cdot R_y \cdot R_z$.
- 4) Обчислити загальну матрицю W перетворення відповідно до варіанту (по дві колонки таблиці).
- 5) Перетворити точку P послідовно матрицями S , R , T відповідно до варіанту (по дві колонки таблиці).
- 6) Перетворити точку P матрицею W .

Результати перетворення необхідно записати в числовому вигляді рисунка, на якому відобразити вихідну точку і результати всіх перетворень (S , R , T , W).

Таблиця 4.1 – Вихідні дані матричного перетворення

№ п/п	Послідовність перетворень W	Координати точки P	Масштаб (S_x, S_y, S_z) S	Обертання (R_x, R_y, R_z) R	Зсув (T_x, T_y, T_z) T
1	SRT	1, 1, 1	1, 1, 0.5	30, 0, 0	1, 0, 0
2	TRS	0, 0, 1	2, 1, 3	90, 0, 45	2, 2, 0
3	RST	1, 0, 1	0.2, 2, 1	120, 0, 30	-2, 3, 1
4	SRT	-1, -1, 1	0.5, 0.5, 0.5	30, 0, 0	0, -1, 2
5	RST	-1, 1, 0	0.1, 0.1, 2	90, 0, 45	-1, 1, 0.5
6	TSR	0, 0, 1	1, 1, 0.5	0, -45, 0	-2, 3, 1
7	SRT	-1, 1, 0	0.2, 2, 1	45, 45, 45	0, -1, 2
8	TSR	1, 0, 1	0.1, 0.1, 2	120, 0, 30	3, -0.25, 1
9	TRS	1, 0, 0	2, 1, 3	30, 0, 0	2, 2, 0
10	SRT	-1, 1, 0	0.1, 0.1, 2	45, 45, 45	-2, 3, 1
11	RST	0, 0, 1	0.2, 2, 1	0, 75, 0	1, 0, 0
12	TSR	-1, -1, 1	0.5, 0.5, 0.5	0, -45, 0	-1, 1, 0.5
13	SRT	-1, 1, 0	0.5, 4, 0.1	0, 75, 0	2, 2, 0
14	TSR	-1, -1, 1	2, 1, 3	45, 45, 45	-2, 3, 1
15	TRS	1, 0, 0	0.1, 0.1, 2	90, 0, 45	0, -1, 2
16	RST	1, 1, 1	0.5, 0.5, 0.5	120, 0, 30	3, -0.25, 1
17	SRT	0, 0, 1	0.2, 2, 1	30, 0, 0	1, 0, 0
18	TRS	1, 1, 0	1, 1, 0.5	120, 0, 30	-1, 1, 0.5
19	RST	1, 0, 0	0.5, 4, 0.1	0, -45, 0	2, 2, 0
20	SRT	1, 1, 1	2, 1, 3	90, 0, 45	0, -1, 2

Контрольні питання

1. Які системи координат використовуються в тривимірній графіці?
2. В якому вигляді представляються перетворення в тривимірній графіці?
3. Яке перетворення подається одиничною матрицею?
4. Що таке однорідний вектор?
5. Операція масштабування.
6. Подання масштабування в матричному вигляді.
7. Операція переміщення.
8. Подання переміщення в матричному вигляді.
9. Операція обертання.
10. Подання обертання в матричному вигляді.
11. Як комбінуються послідовні перетворення?
12. У чому суть процесу рендерингу?
13. Для чого використовується локальний простір?
14. Що таке світова система координат?
15. Що таке простір виду?
16. Як подається видимий спостерігачем простір?
17. Що визначає камера?
18. У чому полягає процес видалення невидимих граней?
19. Які існують варіанти розміщення трикутників щодо піраміди видимості?
20. Для чого використовується відсікання?
21. Якими площинами визначається піраміда видимості?
22. Що таке проекція?
23. Для чого використовується проектування?
24. Які основні види проекцій використовуються в тривимірній графіці?
25. Що таке вікно проекції?
26. У чому суть процесу растеризації?
27. Що таке Z-буфер і для чого він використовується?

Список літератури

1. Гилой В. Интерактивная машинная графика : структуры данных, алгоритмы, языки / В. Гилой. — М. : Мир, 1981. — 380 с.
2. Дуда Р. Распознавание образов и анализ сцен : пер. с англ. / Р. Дуда, П. Харт. — М. : Мир, 1976. — Гл. 7, 9.
3. Ньюмен П. Основы интерактивной машинной графики / П. Ньюмен, Р. Спрулл ; пер. с англ. В. М. Грина, О. Н. Родинко. — М. : Мир, 1976. — 573 с.
4. Павлидис Т. Алгоритмы машинной графики и обработки изображений / Т. Павлидис. — М. : Радио и связь, 1986. — 198 с.
5. Порев В.Н. Компьютерная графика : учеб. пособие / Виктор Порев. — СПб. : ВHV-Санкт-Петербург, 2002. — 428 с.
6. Прэтт У. Цифровая обработка изображений : в 2 кн. / У. Прэтт ; пер. с англ. под ред. Д. С. Лебедева. — М. : Мир, 1982. — Гл. 2, 3, 12, 13, 17, 18, 20.
7. Роджерс Д. Алгоритмические основы машинной графики : пер. с англ. / Д. Роджерс. — М. : Мир, 1989. — 503 с.
8. Роджерс Д. Математические основы машинной графики / Д. Роджерс, Дж. Адамс ; пер. со 2-го англ. изд. П. А. Монахова [и др.]. — М. : Мир, 2001. — 604 с.
9. Тихомиров Ю. Программирование трехмерной графики : OpenGL, создание реалистических образов / Ю. Тихомиров. СПб. : ВHV-Санкт-Петербург, 1998. — 245 с.
10. Фоли Дж. Основы интерактивной машинной графики : в 2 кн. / Дж. Фоли. — М. : Мир, 1987. — Кн. 1. — 367 с. ; Кн. 2. — 365 с.
11. Шикин Е. В. Компьютерная графика. Динамика, реалистические изображения / Е. В. Шикин, А. В. Боресков. — М. : ДИАЛОГ-МИФИ, 1995. — 286 с.
12. Шикин Е. В. Компьютерная графика. Полигональные модели / Е. В. Шикин, А. В. Боресков. — М. : Диалог-МИФИ, 2000. — 461 с.
13. Эйнджел Э. Интерактивная компьютерная графика : ввод. курс на базе OpenGL / Эдвард Эйнджел ; пер. с англ. и ред. В. Т. Тертышного. — 2-е изд. — М. : Вильямс, 2001. — 590 с.

Навчальне видання

ЗУЄВ Андрій Олександрович
ЄВСЕЄНКО Олег Миколайович
КРИЛОВА Вікторія Анатоліївна

ОСНОВИ КОМП'ЮТЕРНОЇ ГРАФІКИ.
ЧАСТИНА 1

Методичні вказівки

до виконання практичних та лабораторних робіт
для студентів спеціальностей
151 «Автоматизація та комп'ютерно-інтегровані технології»
172 «Телекомунікація та радіотехніка»

Відповідальний за випуск Качанов П.О.
Роботу до друку рекомендував О.В. Дудник

Редактор Самініна О.С

План 2020 р., Поз. 56

Підписано до друку . Формат 60'84 1/16. Папір друк. № 2.
Друк - ризографія. Гарнітура Times New Roman. Умов. друк. арк. 3,2.
Обл.-вид. арк. 2,7. Наклад 100 прим. Зам. № . Ціна договірна.
