

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається  
т.в.о. завідувача кафедри  
\_\_\_\_\_ Сафонова С.О.  
« \_\_\_\_ » \_\_\_\_\_ 2020 р.

**МАГІСТЕРСЬКА РОБОТА**

НА ТЕМУ:

Методи удосконалення реалістичності відтворення кольорів у засобах комп'ютерної графіки

Освітньо-кваліфікаційний рівень “Магістр”  
Спеціальність 123 – “Комп’ютерна інженерія”

Науковий керівник роботи:

\_\_\_\_\_

(підпис)

Щербакова М.Є.

(ініціали, прізвище)

Консультант з охорони праці:

\_\_\_\_\_

(підпис)

Критська Я.О.

(ініціали, прізвище)

Студент:

\_\_\_\_\_

(підпис)

Букша К. С.

(ініціали, прізвище)

Група:

КІ-18дм

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки  
Кафедра Комп'ютерних наук та інженерії  
Освітньо-кваліфікаційний  
рівень Магістр  
Напрямок підготовки \_\_\_\_\_  
(шифр і назва)  
Спеціальність 123 – “Комп'ютерна інженерія”  
(шифр і назва)

**ЗАТВЕРДЖУЮ:**

Т.в.о. завідувача кафедри

\_\_\_\_\_ С.О. Сафонова  
« \_\_\_\_\_ » \_\_\_\_\_ 2020 р.

**З А В Д А Н Н Я  
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Букші Катерині Сергіївні

(прізвище, ім'я, по батькові)

1. Тема роботи Методи удосконалення реалістичності відтворення кольорів  
у засобах комп'ютерної графіки

керівник проекту (роботи) к.т.н., доц. Щербакова  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від " 11 " 10 2019 р. № 35/15.15

2. Строк подання студентом роботи \_\_\_\_\_

3. Вихідні дані до роботи Матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно  
розробити) 1. Розробка нових моделей відбивної здатності поверхні

2. Урахування перспективи в процедурах зафарбування

3. Розробка програмного модулю для моделювання та тестування методів  
зафарбовування

4. Охорона праці в галузі \_\_\_\_\_

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
презентація

## 6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	ст. викл. Критська Я.О.		

7. Дата видачі завдання \_\_\_\_\_

Керівник \_\_\_\_\_

(підпис)

Завдання прийняв до виконання \_\_\_\_\_

(підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту ( роботи )	Примітка
1	Отримання завдання, збір матеріалів	18.10.19- 24.10.19	
2	Огляд літератури й обґрунтування необхідності дослідження	25.10.19 –23.10.19	
3	Дослідження методів та засобів підвищення реалістичності відтворення кольорів у засобах комп'ютерної графіки	24.10.19 – 24.11.19	
4	Розробка нових моделей відбивної здатності поверхні	11.11.19 –20.11.19	
5	Розробка програмного модулю для моделювання та тестування методів зафарбовування	03.12.20 – 27.12.20	
6	Оформлення пояснювальної записки	28.12.20 – 08.01.20	
7	Підготовка та подання магістерської роботи до захисту	09.01.20 – 10.01.20	

Студент \_\_\_\_\_

( підпис )

Букша К.С.

(прізвище та ініціали)

Науковий керівник \_\_\_\_\_

( підпис )

Щербакова М.Є.

(прізвище та ініціали)

## АНОТАЦІЯ

Букша К.С. Методи удосконалення реалістичності відтворення кольорів у засобах комп'ютерної графіки.

Розглянуто методи та засоби підвищення реалістичності відтворення кольорів у засобах комп'ютерної графіки, виявлено недоліки. Розроблено нові моделі відбивної здатності поверхні, складність обчислення яких менша, порівняно з існуючими аналогами, розроблено метод перспективно-коректного зафарбування, а також модифікацію ДФВЗ, що є енергетично-коректною, тобто забезпечує виконання закону збереження енергії при відбитті світла від поверхні об'єкта.

**Ключові слова:** двопротенева функція відбивної здатності поверхні, модель, рендеринг, комп'ютерна графіка, графічне зображення, програма, сцена.

## АННОТАЦИЯ

Букша Е.С. Методы усовершенствования реалистичности передачи цвета в средствах компьютерной графики.

Рассмотрены методы и средства повышения реалистичности цветопередачи в средствах компьютерной графики. Разработаны новые модели отражательной способности поверхности, сложность вычисления которых меньше по сравнению с существующими аналогами, разработан метод перспективно-корректного окрашивания, а также модификация ДФВЗ, которая является энергетически-корректной, то есть обеспечивает выполнение закона сохранения энергии при отражении света от поверхности объекта.

**Ключевые слова:** двухлучевая функция отражательной способности поверхности, модель, рендеринг, компьютерная графика, графическое изображение, программа, сцена.

## ABSTRACT

Buksha K.S. Methods for improving the realism of color reproduction in computer graphics.

Methods and means to improve the realism of color reproduction in computer graphics was considered. New models of the reflectivity of the surface were developed, which complexity is less in comparison with existing analogs, a method of perspective-correct brushing and modification BRDF which is energetically correct was developed, that is, ensures that the law of conservation of energy when light is reflected from the surface of the object is complied.

**Keywords:** bidirectional reflectance distributive function, model, rendering, computer graphics, graphic image, program, scene.

## ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 .....	10
АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ОСВІТЛЕННЯ ТА ОБРАННЯ НАПРЯМКУ	
ДОСЛІДЖЕННЯ.....	10
1.1    Етапи формування зображень у комп'ютерній графіці .....	10
1.2    Дослідження моделей відбивної здатності поверхні.....	13
1.3    Методи зафарбовування тривимірних об'єктів .....	19
1.4    Моделі відтворення кольору.....	24
1.4.1    Модель RGB.....	25
1.4.2    Субтрактивна кольорова модель СМΥΚ.....	27
1.4.3    HSV .....	28
1.5    Постановка мети і завдань магістерської роботи .....	29
1.6    Висновки до першого розділу .....	30
РОЗДІЛ 2 .....	32
УДОСКОНАЛЕННЯ ТА РОЗРОБКА НОВИХ МОДЕЛЕЙ ВІДБИВНОЇ ЗДАТНОСТІ	
ПОВЕРХНІ. УРАХУВАННЯ ПЕРСПЕКТИВИ В ПРОЦЕДУРАХ ЗАФАРБУВАННЯ .....	
2.1    Апроксимація ДФВЗ Шліка.....	32
2.2    Апроксимація ДФВЗ квадратичною функцією.....	34
2.3    Модель з використанням косинус – квадратичної функції .....	37
2.4    Тонування за методом Гуро .....	39
2.5    Створення енергетично-коректної двопроменевої функції відбивної здатності поверхні .....	44
2.6    Апроксимований метод з'єднання зображення на основі алгоритму зворотного трасування променів.....	46
2.7    Висновки до другого розділу.....	49
РОЗДІЛ 3 .....	51
РЕАЛІЗАЦІЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ У ГРАФІЧНИХ ЗАСОБАХ	
КОМП'ЮТЕРНОЇ ГРАФІКИ. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ.....	
3.1    Апаратна реалізація ДФВЗ.....	51
3.2    Розробка шейдерних програм та плагінів на базі отриманих результатів .....	53
3.3    Розробка програмних засобів для відтворення кольорів у комп'ютерній графіці .....	58
3.4    Експериментальні дослідження.....	61
3.5    Висновки до третього розділу .....	65
РОЗДІЛ 4 .....	66

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ .....	66
4.1 Аналіз стану умов праці .....	66
4.2 Аналіз потенційних небезпечних і шкідливих виробничих факторів при роботі з персональним комп'ютером .....	67
4.3 Заходи з охорони праці .....	68
4.3.1 Загальні заходи безпеки .....	68
4.3.2 Електробезпека .....	69
4.3.3 Розрахунок захисного заземлення .....	70
4.4 Заходи, що забезпечують виробничу санітарію та гігієну праці .....	72
4.4.1 Мікроклімат .....	72
4.4.2 Освітлення .....	73
4.5 Рекомендації щодо пожежної безпеки .....	74
4.6 Екологія .....	76
4.7 Висновки до четвертого розділу .....	77
ВИСНОВКИ .....	78
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	80
Додаток А Лістинг програмної реалізації .....	84
Додаток Б Презентація .....	102

## ВСТУП

**Актуальність теми.** Комп'ютерна графіка – галузь людської діяльності, пов'язана з використанням комп'ютерів для створення зображень і обробки візуальної інформації, отриманої з реального світу. Вона забезпечує організацію більш ефективний обмін інформаційного каналу між користувачем і ПК, у зв'язку з чим значення графічного зображення результатів обчислень безупинно збільшується.

Формування просторових зображень є складним обчислювальним процесом. Це обумовлено багатоетапністю та високою трудомісткістю геометричних перетворень. При формуванні тривимірних зображень важливо правильно передати градації кольорів, що є визначальним при створенні ілюзії об'ємності тривимірного об'єкта.

У реальному світі поширення світла описується набором складних рівнянь, розв'язок яких для стаціонарної сцени вимагає значних обчислювальних витрат, а візуалізація динамічних сцен в режимі реального часу за таких умов взагалі неможлива. Поширення світла, відбиття його від поверхні описується складними математичними залежностями. Для практичної реалізації візуалізації були створені моделі освітлення Фонга, Бліна, Ламберта, Шліка, які використовують лише певні закони для імітації освітлення тривимірних сцен.

Однією з найбільш трудомістких процедур при зафарбовування поверхонь є розрахунок двопроменевої функції відбивної здатності поверхні (ДФВЗ), яка відповідає за оптичні властивості матеріалу. Вона є моделлю освітлення та визначає, яку частку випромінювання, що надійшло в точку з напрямку джерела світла, буде відбито в напрямку спостерігача. У системах комп'ютерної графіки у переважній більшості випадків двопроменеву функцію розраховують програмним шляхом, що, безумовно, відображається на швидкодії формування тривимірних зображень.

Окрім продуктивності візуалізації, важливою вимогою є фізична адекватність растеризованих моделей. Відомі на даний час моделі освітлення та їх апроксимації мають ряд недоліків: не враховують закон збереження енергії, спектральні особливості розсіювання світла.

У зв'язку з цим актуальною задачею є підвищення реалістичності відтворення кольорів у засобах комп'ютерної графіки за рахунок розробки нових методів, які б забезпечили спрощення процедур візуалізації як на програмному, так і на апаратному рівнях.

**Зв'язок роботи з науковими програмами, планами, темами.** Робота виконувалася відповідно до плану науково-дослідних робіт кафедри комп'ютерних наук та інженерії.

**Мета і задачі дослідження.** Мета роботи – збільшення реалістичності відображення кольорів у системах комп'ютерної графіки. Для досягнення поставленої мети необхідно розв'язати ряд наступних задач:

- провести аналіз методів зафарбовування та вже існуючих моделей освітлення у комп'ютерній графіці;
- створити нові методи та моделі освітлення на основі двопроменевої функції відбивної здатності поверхні;
- розробити засоби підвищення реалістичності відтворення моделей освітлення завдяки підвищенню фізичної адекватності;
- розробити методи перспективно-коректного зафарбування об'єктів;
- розробити програмне забезпечення для моделювання отриманих результатів;
- дослідити ефективність розроблених моделей.

**Об'єкт дослідження** – процес зафарбовування тривимірних графічних об'єктів у системах комп'ютерної графіки.

**Предмет дослідження** – високопродуктивні засоби зафарбовування тривимірних об'єктів.

**Методи дослідження.** У процесі дослідження застосовувалися: теорія чисел і чисельних методів; теорія алгоритмів; методи аналітичної геометрії, лінійної алгебри, диференціального й інтегрального числення для розробки моделей освітлення, а також для розробки методів та засобів підвищення реалістичності відтворення кольорів у засобах комп'ютерної графіки; комп'ютерне моделювання для аналізу та перевірки достовірності отриманих теоретичних положень.

**Наукова новизна одержаних результатів** полягає у наступному:

1) подальшого розвитку дістали моделі двопроменевої функції відбивної здатності поверхні, що на відміну від існуючих, мають більш високу точність і меншу обчислювальну складність, що дозволяє підвищити їх продуктивність при формуванні графічних сцен, зокрема:

- подальшого розвитку отримала модифікація функції Шліка, яка значно точніше відтворює як зону відблиску, так і зону блюмінгу;
- розроблено апроксимацію ДФЗ квадратичною функцією, що дозволяє відносно рівномірно розподілити абсолютну похибку на всьому інтервалі визначення функції.



– розроблено модель з використанням косинус-квадратичної функції, що дозволяє достатньо точно апроксимувати не лише епіцентр відблиску, а також зону затухання.

2) розроблено методи підвищення реалістичності зображень при зафарбовуванні тривимірних об'єктів шляхом встановлення відповідності кольору точок поверхонь в об'єктній та екранній системах координат, що забезпечує підвищення точності визначення інтенсивностей складових кольору точок поверхонь об'єктів.

3) вперше розроблено енергетично-коректну ДФВЗ, що забезпечує виконання закону збереження енергії, що забезпечує підвищення реалістичності.

**Практичне значення одержаних результатів** полягає в тому, що на основі проведених теоретичних досліджень:

- розроблено шейдерні програми, що інтегровані в професійний графічний конвеєр;
- отримано експериментальні оцінки для розроблених моделей і методів;
- запропоновано нову структуру пристрою для визначення інтенсивності спекулярної складової кольору;
- розроблено програму для відтворення спекулярної складової кольору.

**Особистий внесок здобувача.** Усі результати отримано автором самостійно.

**Апробація результатів.** Основні результати магістерської атестаційної роботи докладалися на ІХ Всеукраїнській науково-практичній конференції «ЕЛЕКТРОНІКА ТА ТЕЛЕКОМУНІКАЦІЇ» та на ХХІІ Міжнародній науково-технічній конференції «ТЕХНОЛОГІЯ-2019» (м. Сєвєродонецьк).

**Публікації.** Основні результати магістерської атестаційної роботи опубліковано в 3 наукових працях, з яких 1 наукова стаття у фахових виданнях та 2 тез доповідей.

**Структура та обсяг роботи:** магістерська робота складається зі вступу, чотирьох розділів та висновків на 73 сторінках, містить посилання на список із 50 використаних літературних джерел на 4 сторінках, додатків на 26 сторінках. Загальний обсяг роботи складає 110 сторінок. В магістерській роботі міститься 51 рисунок, 38 формул, 4 таблиці, 2 додатки.

# РОЗДІЛ 1

## АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ОСВІТЛЕННЯ ТА ОБРАННЯ НАПРЯМКУ ДОСЛІДЖЕННЯ

### 1.1 Етапи формування зображень у комп'ютерній графіці

Комп'ютерна графіка стрімко розвивається. За останні кілька десятиліть вона пройшла шлях від креслення простих ліній і відрізків, до побудови віртуальної реальності та створення повнометражних кінофільмів.

На сьогодні за допомогою комп'ютерної графіки вирішується велике коло завдань в усіх областях людської діяльності [1].

Видимий колір об'єкта в реальному світі залежить від властивостей матеріалу, з якого складається його поверхня, від характеристик світла, що падає на нього, і стану навколишнього середовища. Світло має хвильову природу і фізика процесу освітлення дуже складна. Реалізація в графічній системі моделі, наближеної до реальності, призвела б до величезних обчислювальних витрат. На практиці для зафарбовування зображень використовується ряд спрощених моделей і підходів. Спрощена модель освітлення розглядає окремо дифузно відбите або розсіяне світло, а також дзеркально відбите світло. Ключові параметри, що впливають на процес, є інтенсивності джерел світла і геометричні характеристики, такі як кут падіння світла і кут між напрямком на спостерігача і відбитим променем світла.

Логічна сукупність обчислень, що виконуються послідовно і дають на виході синтезовану сцену називається графічним конвеєром. Обчислення в конвеєрі розділені на кілька етапів, у кожному з яких апаратно або програмно виконується визначена функція.

Основна більшість програм для моделювання тривимірних об'єктів використовують стандартну схему побудови 3D сцен. При формуванні тривимірного зображення виділяють декілька етапів опису тривимірного зображення, геометричних перетворень і візуалізації [2] (рис. 1.1).

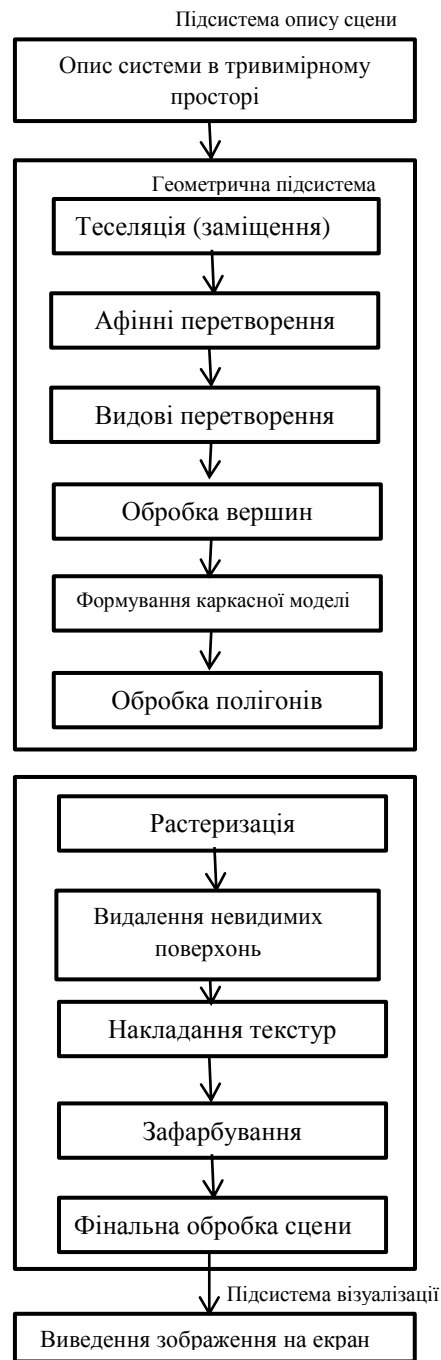


Рисунок 1.1 – Графічний конвеєр

На стадії опису сцени визначаються стани складових об'єктів, їх взаємне розташування та визначається алгоритм наступних дій над об'єктами. На стадії геометричних перетворень виконують декомпозицію графічної сцени та афінні перетворення над отриманими об'єктами. Після перетворень з глобального простору в простір спостерігача виконують відсікання, вилучення невидимих граней та конвертують отримані результати в екранний простір. Надалі визначаються параметри вершин тривимірної сцени, включаючи їх розташування в екранній системі координат, вектори

нормалей, освітленість, текстурні координати. На цьому етапі найважливішими процедурами є процедура перетворень і освітлення, тому весь етап оброблення вершин часто називають T&L (Transformation and Lighting). У подальшому здійснюється «збір» трикутників і формування каркасної моделі. На етапі обробки полігонів оброблюються вхідні примітиви як цілісні об'єкти та формування нових при необхідності. Цей етап введено в графічний конвеєр після появи стандартів DirectX 10 [3] і Open GL 2.1 [4].

Комп'ютерна візуалізація – це процес отримання зображення за моделлю за допомогою комп'ютерної програми, що дає можливість користувачу спостерігати результати комп'ютерного моделювання явищ і процесів. При візуалізації враховують специфікацію об'єктів, їх атрибути, взаємне розташування, динаміку і способи взаємодії [5].

Віртуальний простір моделювання, тобто сцена, включає в себе кілька категорій об'єктів:

- матеріали (інформація про візуальні властивості моделі);
- геометрія (побудована за допомогою різних технічних моделей);
- віртуальні камери (вибір точки та кута побудови проекції);
- джерела світла (налаштування напрямку, потужності, спектра освітлення);
- сили та дії (налаштування динамічних спотворень об'єктів, застосовується в основному в анімації);
- додаткові ефекти (об'єкти, що імітують атмосферні явища, такі як дощ, туман).

Завдання тривимірного моделювання – опис об'єктів і розміщення їх у сцені з допомогою геометричних перетворень відповідно вимогам готового зображення.

Етап візуалізації – це етап кінцевого рендерингу, на якому згідно з даними, отриманими на етапі геометричних перетворень, формуються пікселі зображення, для яких визначаються екранні координати та інтенсивності кольору.

На цьому етапі векторна просторова модель перетворюється на плоску (растрову) картинку. Як структура даних, зображення на екрані представлено матрицею точок, де кожна точка визначена мінімально трьома числами: інтенсивністю червоного, синього і зеленого кольору. Таким чином рендеринг перетворює тривимірну векторну структуру даних у плоску матрицю пікселів. Цей крок вимагає дуже складних обчислень, особливо при створенні ілюзії реальності. У процесі візуалізації потрібно створити ілюзію матеріалів, з яких виготовлені об'єкти, а також розрахувати спотворення цих об'єктів за рахунок прозорих середовищ [6].

Існує декілька технологій візуалізації, часто комбінованих разом. Наприклад:

- Z-буфер (використовується в OpenGL і DirectX 10);

- трасування променів (рейтрейсінг, англ. Raytracing) – колір пікселя уточнюється за рахунок побудови додаткових променів (відображених, заломлених і т. д.) від точки перетину променя погляду. Застосовується тільки для зворотнього трасування променів (від спостерігача до джерела світла), пряма вкрай неефективна і споживає занадто багато ресурсів для отримання якісної картинки;

- сканлайн (scanline) – він же Ray casting («кидання променя», спрощений алгоритм зворотного трасування променів) – розрахунок кольору кожної точки картини побудовою променя з точки зору спостерігача через уявне отвір в екрані на місці цього пікселя «в сцену» до перетину з перший поверхнею.

- глобальне освітлення (англ. global illumination, radiosity) – розрахунок взаємодії поверхонь і середовищ у видимому спектрі випромінювання за допомогою інтегральних рівнянь.

Існує тонка грань між алгоритмами трасування променів, яка в даний час практично стерлася. Так, стандартний візуалізатор в Autodesk 3D Studio Max називається Default scanline renderer, але він помічає не лише внесок дифузного, відбитого та власного світла, але і згладжені тіні. Зважаючи на цю особливість, частіше поняття Raycasting відноситься до зворотного трасування променів, а Raytracing - до прямої [7, 8].

## 1.2 Дослідження моделей відбивної здатності поверхні

Існує поняття двопроменевої функції відбивної здатності поверхні (ДФВЗ), що обчислює оптичні властивості поверхні. На рис. 1.2 зображено класифікацію локальних моделей освітлення, що містить найпоширеніші на даний час двопроменеві функції відбивної здатності поверхні. Наведена класифікація може доповнюватися новими моделями.

Локальні моделі освітлення розділені на дві групи. Перша включає ДФВЗ поверхні, що були отримані експериментальним шляхом. Це досить велика група найбільш поширених моделей, оскільки вони прості та забезпечують задовільну реалістичність відтворення об'єктів, для яких не вимагається точна фізична інформація про освітлення. До другої групи відносяться більш точні моделі, які, як правило, ураховують як корпускулярну, так і хвильову природу світлового потоку. В таких моделях поверхня розглядається у вигляді мікроскопічних ділянок. При цьому враховується шорсткість поверхні. У деяких моделях, враховується затінення поверхні можливими мікрориступами або їх перешкоджання проходженню світлового потоку.

ДФВЗ можна розділити на ізотропні та анізотропні. Ізотропні ДФВЗ незмінні відносно повороту навколо вектору нормалі до поверхні. В анізотропних двопробневих функціях, навпаки, відбивні властивості матеріалу змінюються при повороті навколо вектора нормалі.

Залежно від методу зафарбовування, при використанні експериментальних ДФВЗ використовуються різні дані. У більшості випадків двопробнева функція визначається через скалярний добуток відповідних нормованих векторів, що дорівнює косинусу кута між цими векторами. Також можна зустріти, але значно рідше, оперування зі значенням кута між векторами, оскільки це передбачає виконання обернених тригонометричних функцій [9].

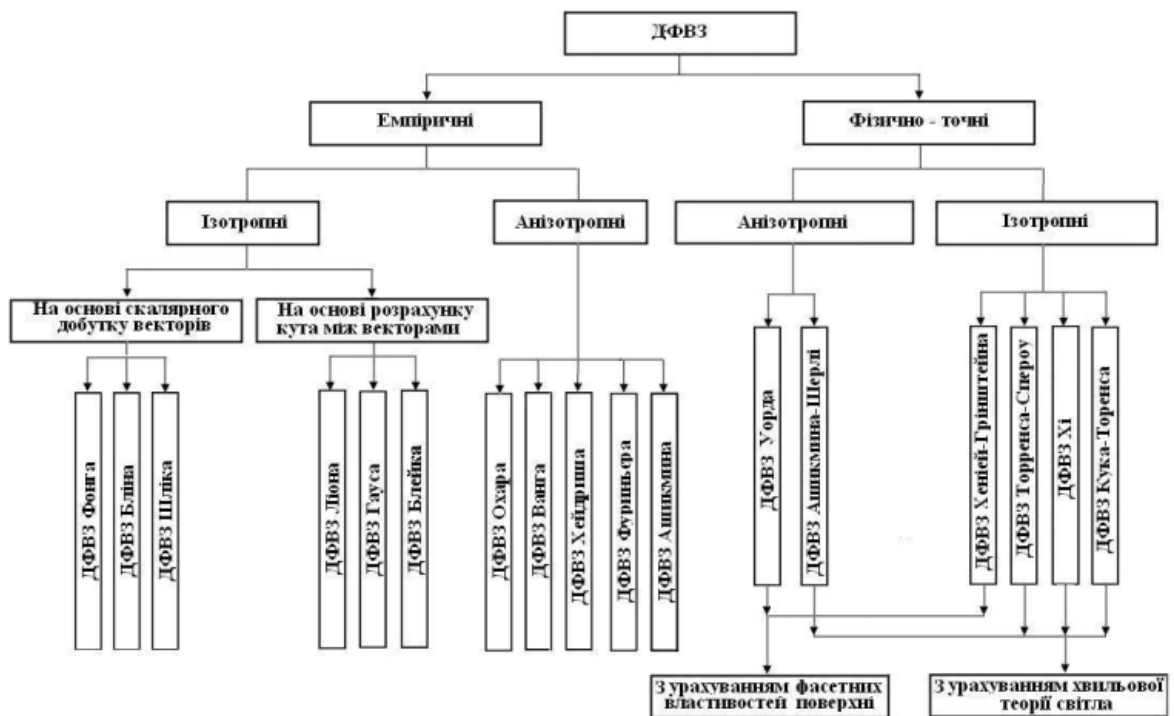


Рисунок 1.2 – Класифікація моделей освітлення

Параметри функції ДФВЗ – напрямок вхідного світла  $L$  і напрямок вихідного, відбитого світла  $V$ , які визначені відносно нормалі  $N$  до поверхні, що чітко демонструє модель освітлення Бліна (рис. 1.3) [10].

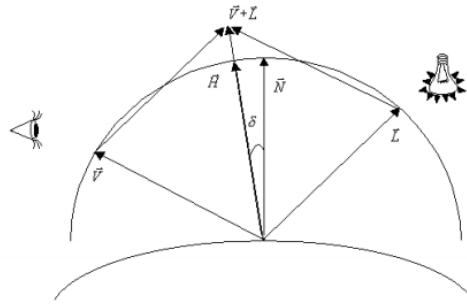


Рисунок 1.3 – Прохід променів відбитого світла у моделі освітлення Бліна

ДФВЗ можуть бути побудовані на основі реальних об'єктів з використанням відкаліброваних камер і джерел світла, але такий метод є надзвичайно ресурсомістким і нерентабельним в умовах широкого розвитку комп'ютерних технологій, саме тому було запропоновано багато аналітичних моделей, які різняться за методами апроксимації (спрощення) розрахунку ДФВЗ. ДФВЗ є функцією кількох змінних:

$$\text{ДФВЗ} = f_{\lambda}(\alpha, \mu_1, \varphi, \mu_2, x) = f(\vec{L}, \vec{V}, \vec{N}), \quad (1.1)$$

де  $\lambda$  – довжина хвилі,

$(\alpha, \mu_1)$  – параметри, що визначають напрямок падаючого світла,

$(\varphi, \mu_2)$  – параметри, що визначають напрямок спостереження.

Інтенсивність випромінювання  $I$  визначається як величина променевого потоку через ділянку  $ds \perp$  у заданому напрямку (рис. 1.4), перпендикулярно променю, в межах диференціального тілесного кута  $d\omega$ , який дорівнює площі нескінченно малого елемента на поверхні одиничної сфери.

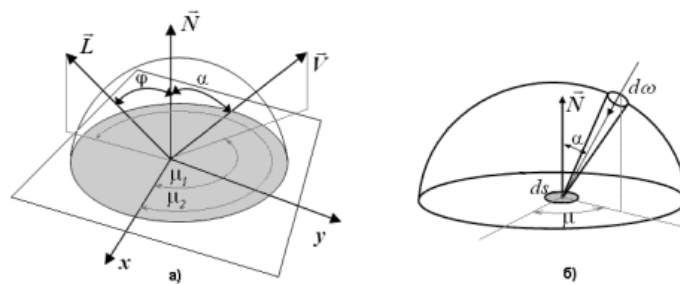


Рисунок 1.4 – Початкові показники для розрахунку ДФВЗ

Модель освітлення Ламберта, зображена на рис. 1.5, що визначає освітленість об'єкта, яка залежить тільки від параметрів сцени і не залежить від положення спостерігача. Модель припускає, що об'єкт розсіює світло однаково в усіх напрямках, а інтенсивність падаючого світла пропорційна косинусу кута падіння (відносно нормалі в точці падіння). Формула, за якою у моделі Ламберта обраховують інтенсивність світла у кожній точці [11]:

$$I = I_a k_a + I_l^{bx} k_d (\vec{N} \times \vec{L}), \quad (1.2)$$

де  $I_l^{bx}$  - інтенсивність зовнішнього джерела світла,

$\vec{N}$  - вектор нормалі до поверхні,

$\vec{L}$  - вектор напрямку світла,

$I_a$  - інтенсивність розсіяної складової кольору,

$k_a$  - коефіцієнт розсіяного відбиття,

$k_d$  - коефіцієнт дифузного відбиття.

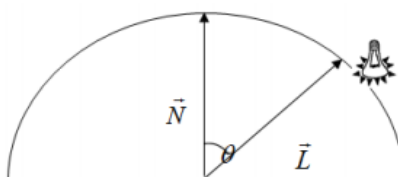


Рисунок 1.5 – Модель освітлення Ламберта

Згідно моделі Ламберта поверхня виглядає рівномірно яскравою на всіх напрямках. У природньому середовищі такого освітлення майже не існує, але є досить наближені моделі. Модель Ламберта враховує лише ідеальне розсіювання світла (рис. 1.6) [4].



Рисунок 1.6 – Ілюстрація освітлення об'єкта на основі моделі Ламберта

Також існує не менш популярна і відома модель освітлення Фонга – емпірична модель, яка базується на фізичних спостереженнях. Для блискучих поверхонь спекулярна складова світла досить велика, а інтенсивність кольору падає дуже швидко, а для шороховатих поверхонь дифузна складова більша і інтенсивність спадає досить повільно. Фонг удосконалив модель Ламберта, яка містила лише вектор спостереження  $V$  і вектор



напрямку світла  $L$ , методом вводу третьої складової – вектора відбитого світла  $R$  (рис. 1.7, рис. 1.8) [12].

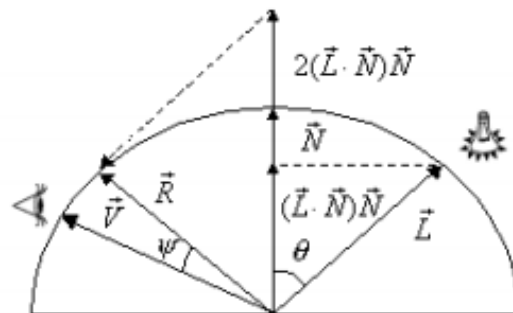


Рисунок 1.7 – Модель освітлення Фонга

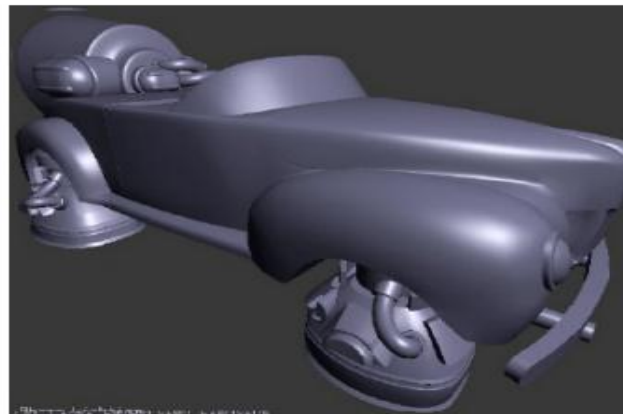


Рисунок 1.8 – Ілюстрація освітлення об'єкта на основі моделі Фонга

Модель Торранса-Спарроу (рис. 1.9) – мікрофасетна модель (фр. *facete* "грань відшліфованого каменя"), при якій поверхня, відбиваюча світло, розглядається як сукупність блискучих мікрограней, орієнтованих в різноманітних напрямках, так звана фасетна поверхня [13, 14]. Мікрофасетна модель передбачає існування між вектором падаючого випромінювання  $L$  і вектором спостереження  $V$  деякого вектора  $H$ , який утворює однаковий кут  $\theta$  з векторами  $L$  і  $V$ . Цей «половинний» вектор називається фасетною нормаллю і визначається як:  $H = \frac{L+V}{|L+V|}$ .

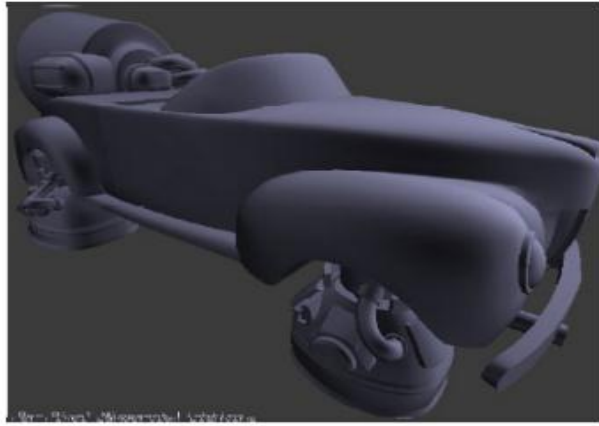


Рисунок 1.9 – Ілюстрація освітлення об'єкта на основі моделі Торранса-Спарроу

Модель Кука-Торренса базується на мікрофасетній моделі за якою поверхня, яка відбиває світло, розглядається як сукупність блискучих мікрограней [14]. Кожна мікрогрань відображає падаюче світло, і тільки ті грані, які орієнтовані певним чином, відбивають світло в бік спостерігача і впливають на освітленість.

Загальна освітленість  $R$  розглядається як:

$$R = sR_s + dR_d, \quad (1.3)$$

де  $S$  – коефіцієнт дзеркального відображення;

$d$  – коефіцієнт дифузного відображення, причому:  $s + d = 1$ ;

$R_s$  - дзеркальна складова освітленості;

$R_d$  - дифузна складова освітленості.

По закону Ламберта розраховується дифузна складова, а дзеркальна складова визначається за наступною формулою:

$$R_s = \frac{FDG}{(N, V)}, \quad (1.4)$$

де  $F$  – коефіцієнт Френеля (визначає співвідношення відбитої і заломленої енергії на плоскій границі двох середовищ з різними показниками заломлення),

$D$  – частка граней, націлених за напрямком до спостерігача,

$G$  – коефіцієнт затемнення мікрограней,

$V$  – одиничний вектор, націлений в бік спостерігача,

$N$  – нормаль до поверхні.

Частка граней, націлених за напрямком до спостерігача може бути обчислена за однією із функцій розподілу, наприклад за функцією розподілу Гауса.

Модель освітлення Кука-Торренса є досить трудомісткою з обчислювальної точки зору. Проте, дана модель найбільш точно описує поведінку світла для широкого кола матеріалів (рис. 1.10).

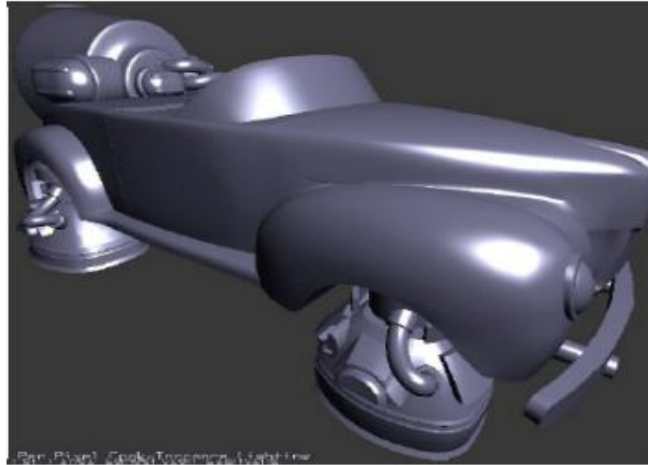


Рисунок 1.10 – Демонстрація освітлення об'єкта на основі моделі Кука-Торренса

Модель Шліка – це найбільш широко поширена і використовувана модель, яка базується на мікрофасетній моделі, рівнянні Френеля та наборі параметрів, що дозволяють використовувати модель в анізотропній (відмінність властивостей у різних напрямках) і ізотропній (однорідній) формі. Шлік запропонував апроксимувати класичну модель освітлення Фонга у такій формі [15]:

$$S_n(\cos x) = \frac{\cos x}{n - n \cos x + \cos x}, \quad (1.5)$$

де  $n \in \infty [1; )$  параметр контролю інтенсивності освітлення;

$\cos x$  – косинус кута між вектором спостереження  $V$  і вектором відображення  $R$ .

Порівнюючи з розглянутими моделями, модель Шліка є менш затратною з економічної точки зору і кількості складних математичних операцій, а також не потребує пам'яті (лише параметр  $n$  необхідно зберігати у постійній пам'яті).

### 1.3 Методи зафарбовування тривимірних об'єктів

Конвертація 3D-зображень є складною обчислювальною задачею, з цієї причини виконують її декомпозицію. В основному у графічних мережах складові поверхні об'єктів апроксимують системою трикутників, що растеризуються в екранній системі координат після трансформації з глобальної системи. Найпершим методом зафарбовування був метод однотонного зафарбовування, який визначається правилом, що для кожного плоского трикутника визначався вектор нормалі, а на його основі – колір. Складові трикутники об'єкта заповнюються одним кольором без градації, що призвело до стрімкої зміни інтенсивності кольору на границі трикутників. Однотонне зафарбовування потребує

мінімальних обчислювальних затрат. Воно внесено до функцій Direct3D і підтримується тривимірними регуляторами, незважаючи на низьку якість.

Найпоширеніші методи зафарбування тривимірних об'єктів – методи Гуро і Фонга [16]. Зафарбовування по методу Гуро пом'якшує тоновий перехід між ребрами ділянки, обчислюючи інтенсивність кольору для кожного пікселя, за допомогою засобів лінійної інтерполяції. Результат – отримана поверхня реалістично і більш природньо візуалізує процес відбиття світла. Процес зафарбовування складається з таких стадій:

- а) розрахунок векторів нормалей до кожної грані;
- б) визначення інтенсивності кольору у вершинних точках трикутника, за допомогою значення нормалей;
- в) усереднення нормалей усіх граней вершини;
- г) зафарбовування ділянки, обмеженої трикутником, за допомогою лінійної інтерполяції пікселей відображення інтенсивності кольору.

Зазвичай метод Гуро використовують як додатковий етап з інтерактивного формування 3D-відображення. На наступному етапі фінального рендерингу будується повноцінна сцена.

Розглянемо детальніше процес фарбування області, обмеженої трикутником. Спочатку задаються координати вершин трикутника ( $A$ ,  $B$  і  $C$ ), за значенням яких визначають зріст координат для кожної сторони. За методом Гуро задаються значення інтенсивності у вершинах -  $I_a$ ,  $I_b$ ,  $I_c$  за допомогою інтерполяції трикутника (рис. 1.11). Інтерполяція — спосіб знаходження проміжних значень величини за наявним дискретним набором відомих значень. Необхідно обчислити інтенсивність всіх інших точок в середині даного трикутника і на ребрах. Для цього знайдемо інтенсивність випадково обраної точки  $D$  у середині ділянки.

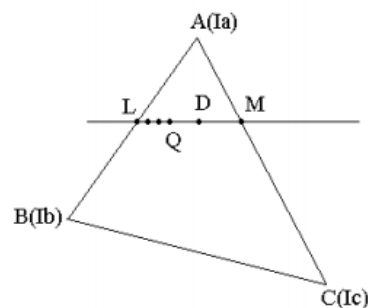


Рисунок 1.11 - Інтерполяція трикутника

Визначимо збільшення інтенсивності світла на піксель уздовж ребер трикутника  $AB$ ,  $AC$ ,  $BC$ :

$$\Delta I_{AB} = (I_A - I_B) / \text{БП}_{AB},$$

$$\Delta I_{AC} = (I_A - I_C) / \text{БП}_{AC},$$

$$\Delta I_{BC} = (I_B - I_C) / \text{БП}_{BC},$$

де БП - більший з приростів координат  $\Delta x$ ,  $\Delta y$  обраного відрізка прямої. Проведемо через точку D відрізок сканування, паралельний горизонтальній осі. Він перетне ребра трикутника в точках L і M. По збільшеннях інтенсивності уздовж ребер визначимо інтенсивність точок, що належать цим ребрам. Розрахувати інтенсивність точок M та L можна за формулами:

$$I_M = I_A + \Delta I_{AB} * \text{БП}_{AM},$$

$$I_L = I_A + \Delta I_{AC} * \text{БП}_{AL},$$

Наступний крок за алгоритмом визначимо приріст інтенсивності вздовж відрізка  $LM = \text{БП}_{LM}$  за формулою:

$$\Delta I_{LM} = (I_L - I_M) / LM$$

Інтенсивність світла в точці D знаходиться за формулою:

$$I_D = I_L + \Delta I_{LM} * LD$$

Схожим способом визначається інтенсивність будь-якої іншої випадково обраної точки, обмеженої заданою ділянкою. Найменша похибка досягається при розбивці поверхні на прямокутні трикутники з катетами паралельними координатним осям, оскільки в цьому випадку більший приріст (БП) катета збігається з його довжиною. Зафарбування Гуро найбільш доцільне при використанні простої моделі освітлення з дифузійним відбитком, тому що форма відблисків при дзеркальному відбитку сильно залежить від вибору багатокутників, що задають об'єкт або поверхню. На рис. 1.12 наведено приклад зображення, сформованого з використанням методу Гуро [17].

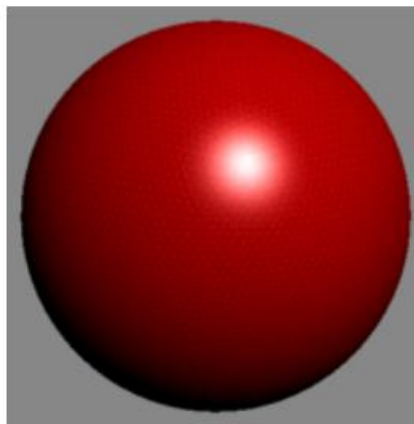


Рисунок 1.12 – Зафарбування за Гуро

До недоліків методу Гуро можна віднести:

- а) метод Гуро використовує лінійну інтерполяцію для визначення інтенсивностей кольору, на відміну від дифузної та спекулярної складової кольору, у яких нелінійний змінний характер;
- б) відблиски відтворюються тільки в разі, якщо вершини трикутників знаходяться в їх зоні, при інших обставинах відблиск не буде сформовано;
- в) вектори нормалей визначаються тільки для вершин трикутника, тому локальна кривизна поверхні не враховується;
- г) присутня зміна інтенсивності кольору зображення кадр за кадром, що виражається у відображенні відблисків, їх мерехтінні;
- д) відблиск, який повинен мати форму еліпса, має форму зірки. Просте пояснення, тому що ділянки відблиску формуються в різних трикутниках;
- є) у методі Гуро не враховується перспектива об'єкта.

Метод Гуро зазвичай використовують у програмах, де важлива швидкість, адже він швидше методу Фонга. У методі Фонга спочатку апроксимуються нормалі до поверхонь в вершинах трикутників, так як і в методі Гуро. Далі переходять до інтерполяції значень вектора нормалі до поверхні всередині багатокутників. Отримані значення вектора нормалі використовуються для визначення інтенсивності кожного пікселя. Помітні поліпшення в порівнянні з інтерполяцією інтенсивності спостерігаються у випадку використання моделі дзеркального відбитка, тому що при цьому більш точно відтворюються світлові відблиски. Проте навіть якщо дзеркальний відбиток не використовується, інтерполяція векторів нормалі дає більш якісні результати, ніж інтерполяція інтенсивності, оскільки апроксимація нормалі в цьому випадку здійснюється в кожній точці. Отже, утворюється більш реалістичне зображення.

На відміну від зафарбування Гуро, де проходить інтерполяція кольорів через полігони, в зафарбуванні Фонга вектор нормалі до поверхні лінійно інтерполюється на багатокутник по нормалям в вершинах багатокутника. Нормаль поверхні інтерполюється і нормалізується в кожному пікселі, а потім використовується в моделі відбиття Фонга, для отримання кінцевого кольору пікселя. Зафарбування по методу Фонгу потребує більше обчислювальних ресурсів, ніж у зафарбування Гуро, оскільки модель відбиття має бути обчисленою в кожному пікселі, а не лише в вершинах.

Процедури інтерполяції для векторів нормалі у методі Фонга ідентичні до пошуку значень інтенсивності у методі Гуро.

При зафарбовуванні за методом Фонга інтенсивність кольору пікселей визначають за формулою:

$$I = I_a k_a + I_l (k_d \cos \lambda + k_s \cos^n \lambda), \quad (1.6)$$

де  $I_a$  – інтенсивність розсіяного джерела світла;

$I_l$  – інтенсивність направленої джерела світла;

$k_a, k_s, k_d$  – коефіцієнти розсіяного, дзеркального і дифузного світла;

$\lambda$  – кут між спостерігачем і відбитим напрямком світла;

$n$  – коефіцієнт дзеркальності поверхні;

$\cos^n \lambda$  – двопротенева функція відбивної здатності поверхні, що характеризує оптичні властивості поверхні.

Як і в методі Гуро є недоліки, метод Фонга також не ідеальний. Головний недолік – чималі обчислювальні витрати, бо для кожної точки обчислюється значення інтенсивності. Шлях вирішення цієї проблеми базується на використанні карти відбитків. Вона представляє із себе набір попередньо підрахованих значень інтенсивностей для певної кількості значень вектора нормалі. Алгоритм роботи цієї схеми полягає в обчисленні один раз карти відбитків, а далі залишається лише зчитувати з неї значення інтенсивності, без обов'язкової процедури перераховування. Як відомо, зафарбування за методом Фонга забезпечує високоякісне зображення, але потребує істотних економічно-обчислювальних витрат, тому для візуалізації складних сцен доцільніше застосовувати алгоритм Гуро, який на даний час використовуються практично в усіх комерційних робочих станціях [16]. На рис. 1.13 показані об'єкти, які зафарбовуються із застосуванням методу Фонга.

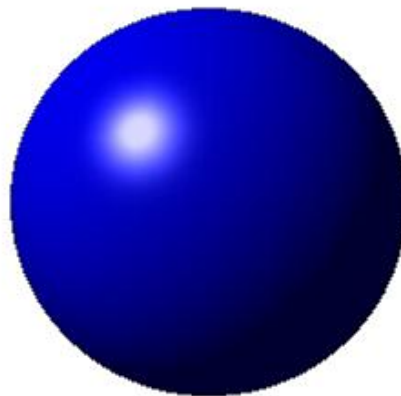


Рисунок 1.13 – Сфера із зафарбуванням методом Фонга

Для наглядного порівняння методів Фонга і Гуро на рис. 1.14 представлено дві фігури, зафарбовані різними методами.

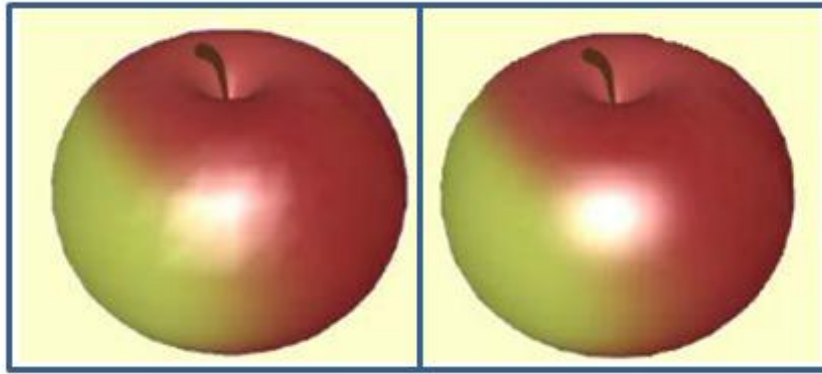


Рисунок 1.14 – Зліва знаходиться фігура, яка зафарбована методом Гуро, праворуч – методом Фонга

#### 1.4 Моделі відтворення кольору

Використання кольору при створенні й редагуванні зображень один з важливих розділів комп'ютерної графіки. У сучасних комп'ютерних програмах маніпуляції з кольором здійснюються за допомогою кольорових моделей і режимів. Кольорові моделі (або кольорові простори) надають засоби для концептуального та кількісного опису кольору.

Будь-який колір у графічних редакторах задається в кольоровій моделі, що визначає аналітичні вирази для обчислення кольорової складової пікселя в різних кольорових просторах (базисах) і для переходу від одного базису до іншого. За допомогою кольорової моделі можна моделювати колір на екрані чи принтері. Діапазон кольорів, що може сприйматися чи відтворюватися називається кольоровим охопленням чи кольоровою гамою пристрою.

Призначення кольорової моделі полягає в можливості зручним чином описувати кольори в межах деякого кольорового діапазону.

Більшість комп'ютерних колірних моделей засновано на використанні трьох основних кольорів, що відповідає сприйняттю кольору людським оком. Кожному основному кольору привласнюється визначене значення цифрового коду, після чого всі інші кольори визначаються як комбінації основних кольорів.

Базові кольори – це кольори, за допомогою яких можна одержати практично весь видимий спектр.

Колір може бути отриманий як у процесі випромінювання світла активним джерелом так і в результаті відображення світла від будь-якого предмету. Цим процесам у комп'ютерній графіці відповідають дві системи представлення кольору: адитивна й



субтрактивна. Кольорові моделі, що використовують у графічних програмах засновані на додаванні (адитивні кольорові моделі) чи на відніманні основних кольорів (субтрактивні кольорові моделі). Моделі обох типів містять кольори, що розраховуються за математичними формулами. Ці формули складають основу вимірів кольору відповідно до кольорових стандартів.

За принципом дії моделі умовно розбивають на наступні класи:

- адитивні (RGB);
- субтрактивні (CMYK);
- перцепційні (HSB).

Адитивні моделі побудовані на складанні кольорів, основу субтрактивних моделей становить операція віднімання кольорів, перцепційні базуються на сприйнятті кольору.

В табл. 1.1 описано детально відмінності моделей RGB і CMYK, а також галузі їх застосування. Відмінність цих моделей обумовлена тим, що вони розроблялися різними компаніями і кожний з розроблювачів використовував свій набір формул.

Таблиця 1.1 — Моделі відтворення кольору й галузі їх застосування у картографічному дизайні

<b>Кольорова модель</b>	<b>Базові кольори моделі</b>	<b>Галузь застосування</b>
<b>CMYK</b>	Блакитний, пурпурний, жовтий, чорний	Чотирикольоровий Багатошаровий друк
<b>RGB</b>	Червоний, зелений, синій	Електронні карти, екранні презентації, Web-графіка і слайди

#### **1.4.1 Модель RGB**

Модель RGB заснована на змішуванні променів червоного (Red), зеленого (Green) і синього (Blue) кольорів і передає практично всі кольори, що сприймаються людським оком. Чим більше насиченість кольору, тим ближче він до повного спектрального кольору, а чим менше, тим ближче він до чорного кольору. Якщо скласти всі три кольори з максимальною насиченістю, то вийде білий колір [18].

Кольоровий простір моделі зручно представити у вигляді кольорового куба (рис. 1.15). За осями координат відкладені значення 3 основних кольорів. Кожний з них може приймати значення від 0 (нульова яскравість) до максимальної рівної 1. Внутрішня частина утвореного куба містить усі можливі сполучення RGB, що утворює кольоровий

простір. На початку координат значення відповідних кольорів рівне 0 – випромінювання відсутнє, отже початок координат – це точка чорного кольору. У протилежній точці по головній діагоналі куба розмішуються максимальні значення кольорів (рівні 1), утворюють білий колір.

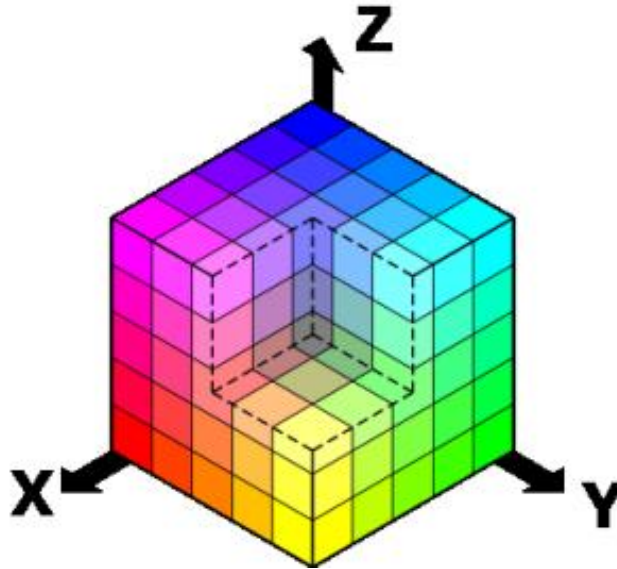


Рисунок 1.15 – RGB колірна модель у вигляді куба

На лінії, що з'єднає ці точки, розташовуються змішування рівних значень основних складових колірних променів щоб утворити сірі відтінки – від чорного до білого. Цей діапазон інакше називається сірою шкалою. Три вершини куба відповідають вихідним кольорам – червоному, зеленому і синьому, інші три відповідають парному змішуванню вихідних кольорів і відповідають – жовтому, пурпурному та блакитному. У звичайному RGB-зображенні кожний із трьох основних кольорів і сіра шкала мають 256 градацій (відтінків).

RGB-модель ближче усього до фізичної сутності монітора. Модель RGB ідеально підходить для представлення кольору в системі, де світло випромінюється екраном монітора, але не підходить для пристрою друку, де передавання кольору побудовано на відбитому світлі. Рис. 1.16 демонструє наглядно змішування кольорів у моделі RGB.



Рисунок 1.16 – Приклад змішування кольорів у моделі RGB

#### 1.4.2 Субтрактивна кольорова модель СМУК

Субтрактивні кольори виходять відніманням вторинних кольорів із загального променя. У цій системі білий колір виходить як результат відсутності всіх кольорів, а їхня присутність дає чорний колір.

Коли необхідно роздрукувати створене на екрані комп'ютера зображення на допомогу приходять субтрактивна система кольору. На практиці використовують чотирьохкольоровий друк з чорним кольором. Прикладом такої колірної моделі є модель СМУК, назва якої відповідає першим буквам основних кольорів: блакитного (Cyan), пурпурного (Magenta), жовтого (Yellow) і black – чорного (у назві К за останньою літерою). (рис. 1.17).

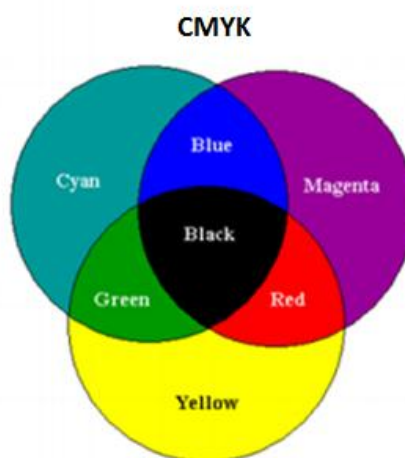


Рисунок 1.17 – Приклад змішування кольорів у моделі СМУК

Простір моделі CMYK аналогічно простору моделі RGB, але система координат перевернута: 0,0,0 – буде відповідати білому кольору, а 1,1,1 – чорному (рис. 1.18). У результаті змішування трьох рівних компонентів отримуємо чорний колір [19].

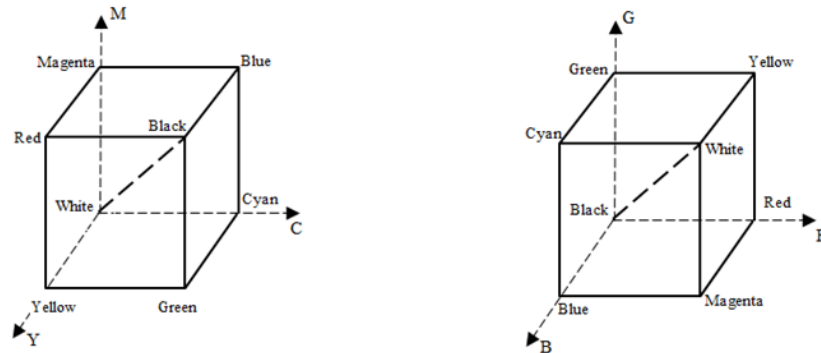


Рис. 1.18 – Колірний куб а) RGB б) CMYK.

### 1.4.3 HSV

Існує метод, що вирішує недоліки своїх попередників. Метод складання кольорової характеристики на основі HSV моделі враховує ситуації зі зміною тону кольору під час затінення об'єкта. HSV відображає кольори у вигляді трьох складових: колірний тон (Hue), насиченість (Saturation) та значення кольору (Value) [20]. Колірний тон - це значення “чистого” кольору, без накладання чорного або білого тонів. Значення колірного тону коливається в межах  $0 - 360^{\circ}$ , або просто від 0 до 100 (рис. 1.19).

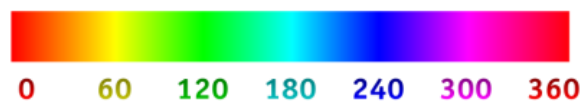


Рисунок 1.19 - Шкала колірного тону (hue)

Насиченість у кольорі відповідає за відтінок білого і лежить у межах від 0 до 100, або від 0 до 1. При показнику 1 колір “чистий”, при 0 - білий. Цей параметр може змінюватись при засвіченні об'єкта білим світлом. Тобто при попаданні на об'єкт променів білого відтінку значення Hue, в ідеальному випадку, не змінюється. Змінює своє значення параметр Saturation. Отже можна зазначити, що обираючи колірну модель HSV, покращується сприйняття алгоритмом впливу світла на об'єкт розпізнавання.

Останнім параметром є - значення кольору. Дана складова відповідає за яскравість кольору та задається в аналогічних межах від 0 до 1, або від 0 до 100. При наближенні показника до 0 колір приймає чорний відтінок. Таку особливість колірної моделі дуже зручно використовувати для аналізу падаючої на об'єкт тіні. Візуалізація HSV простору приведена на рис. 1.20 у вигляді циліндричної системи координат.

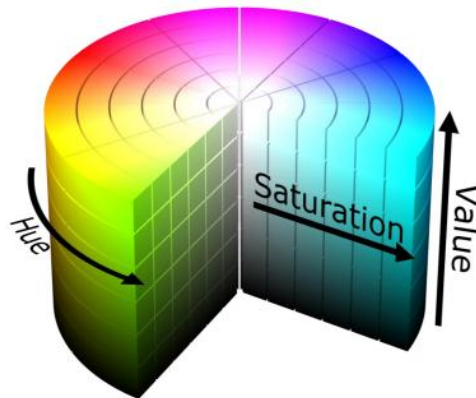


Рисунок 1.20 - Візуалізація HSV моделі

Колірні суміші в RGB просторі можуть відтворювати широкий спектр кольорів, однак співвідношення між складовими червоного, зеленого та синього світла та отриманого кольору не є інтуїтивним, особливо для користувачів, які знайомі з субтрактивним змішуванням кольорів фарб або моделей традиційних художників на основі відтінків. Крім того, ні адитивні, ні субтрактивні колірні моделі не визначають співвідношення кольорів так само, як робить людське око.

Така колірна модель в більшості випадків дає кращі результати ніж RGB, проте вони не є ідеальними. Проблеми полягають у сприйнятті кольорів та світла. Іноді при засвіченні або затіненні об'єкта він приймає відтінок іншого кольору чим спотворює дані алгоритму обробки.

### **1.5 Постановка мети і завдань магістерської роботи**

На основі розглянутої вище інформації можна сформулювати мету роботи, що полягає у збільшенні реалістичності відображення кольорів у системах комп'ютерної графіки. Для досягнення мети необхідно розв'язати ряд наступних задач:

- провести аналіз методів зафарбовування та вже існуючих моделей освітлення у комп'ютерній графіці;

- створити нові методи та моделі освітлення на основі двопроменевої функції відбивної здатності поверхні;
- розробити засоби підвищення реалістичності моделей освітлення за рахунок підвищення їх фізичної адекватності;
- розробити методи перспективно-коректного зафарбування об'єктів;
- розробити програмне забезпечення для моделювання отриманих результатів;
- дослідити ефективність розроблених моделей.

## 1.6 Висновки до першого розділу

У першому розділі було проведено аналіз предметної галузі, виділено основні тенденції розвитку комп'ютерної графіки на даний час, визначено основні етапи графічного механізму. Проаналізувавши та врахувавши вище наведені факти, найбільш поширеними у комп'ютерній графіці є моделі Бліна, Ламберта, Шліка та Фонга. Це пояснюється їх простотою, залежністю від косинуса кута між векторами, а не від кута між ними, а також достатньою точністю формування графічних зображень і мають відносно малу обчислювальну складність. Фізично-точні моделі, які ураховують фасетну структуру поверхні та хвильову теорію світла, доцільно розраховувати програмним шляхом (піксельними шейдерами), оскільки вони достатньо трудомісткі. Моделі Шліка, Кука-Торренса, Торренса-Спарроу – більш реалістичні, але вони використовуються тільки в окремих випадках, де треба сформувати фотореалістичні зображення. Їх обчислювальна складність обмежує їх застосування. Тому в магістерській атестаційній роботі для аналізу було вибрано моделі відбиття Бліна, Ламберта, Фонга, Шліка. А також проведено аналіз сучасних методів і засобів зафарбовування зображень тривимірних сцен. Виявлено, що деталізація об'єктів тривимірного простору в графічних системах з кожним роком зростає в експоненціальній залежності. Показано, що на сучасному етапі розвитку комп'ютерної графіки у зв'язку зі збільшенням геометричної складності графічних сцен необхідна розробка нових методів та моделей.

Аналіз показав, що двопроменеві функції відбивної здатності поверхні мало пристосовані до програмно-апаратної реалізації, оскільки є достатньо трудомісткими і включають «довгі» операції. Розрахунок ДФВЗ передбачає нормалізацію векторів, що вимагає достатньо великих обчислювальних витрат. Ураховуючи, що двопроменева функція й нормалізовані вектори обчислюються для кожної точки поверхні, то продуктивність формування графічних сцен у значній мірі визначається саме реалізацією зазначених процедур.

Спираючись на наукові джерела і проаналізувавши вище розглянуту інформацію, є необхідність розробки нових моделей, а також удосконалення вже існуючих моделей відбивної здатності поверхні. Проведено постановку задачі та визначено методи її досягнення.

## РОЗДІЛ 2

### УДОСКОНАЛЕННЯ ТА РОЗРОБКА НОВИХ МОДЕЛЕЙ ВІДБИВНОЇ ЗДАТНОСТІ ПОВЕРХНІ. УРАХУВАННЯ ПЕРСПЕКТИВИ В ПРОЦЕДУРАХ ЗАФАРБУВАННЯ

#### 2.1 Апроксимація ДФВЗ Шліка

Шлік запропонував спростити функцію  $\cos^n \gamma$ , що використовується в моделі освітлення Бліна для розрахунку інтенсивності дзеркальності складової кольору, функцією

$$H(\gamma) = \frac{\cos \gamma}{n - n \cos \gamma + \cos \gamma}, \quad (2.1)$$

Характерна ознака такого підходу полягає в тому, що вона характеризується через значення коефіцієнта спекулярності  $n$  та  $\cos \gamma$ , а не через кут  $\gamma$ , що є властиво майже усім відомим двопроменевим функціям. Максимальна відносна похибка апроксимації в даному випадку не більша 3,2%. Функція Шліка – монотонно спадна, що задовольняє характер зміни спекулярної складової кольору. На рис. 2.1 показано приклад сформування відблиску на об'єктній поверхні з використанням моделі освітлення Шліка. Також з рисунка видно, що функція Шліка характеризується досить повільним затуханням функції у зоні блюмінгу [21].

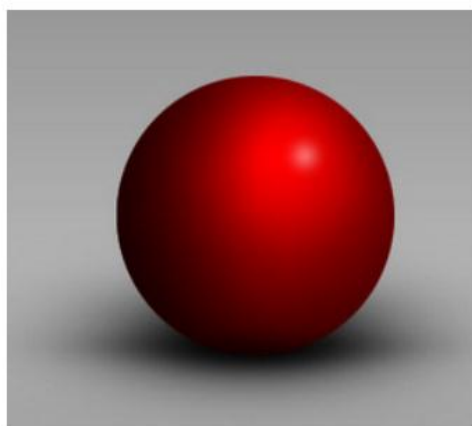


Рисунок 2.1 – Утворення відблиску за формулою Шліка

При апроксимації функції слід враховувати, що стійкі вимоги до точності відображаються при зображенні лише епіцентру відблиску [22]. Для конкретних областей, що характеризують затухання концентрованого світла до мінімального значення, по-



перше є необхідність забезпечення монотонності зміни інтенсивності кольору, що виключає зміну фіксованого зображення [21]. На рис 2.2 зображено класичну модель суцільною лінією та спрощений варіант функції  $\cos^n \gamma$  функцією Шліка пунктирною лінією. З графіку видно, що даний підхід забезпечує достатньо якісне відображення лише для епіцентру відблиску. За розглянутою областю спостерігається значне розходження з результатами, що були отримані згідно з класичною моделлю освітлення. На рис. 2.3 зображено залежність найвищої відносної похибки апроксимації функції  $\cos^n \gamma$  функцією Шліка, для демонстрації епіцентру відблиску в діапазоні зміни функції від 1 до 0,6.

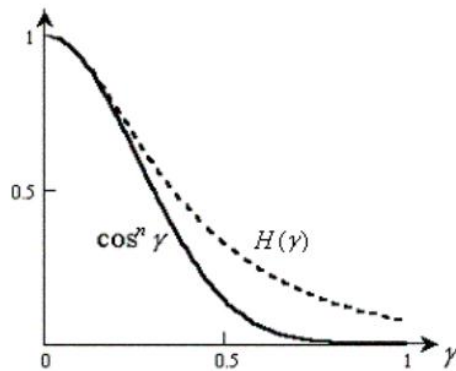


Рисунок 2.2 – Класична модель Бліна та модель Шліка

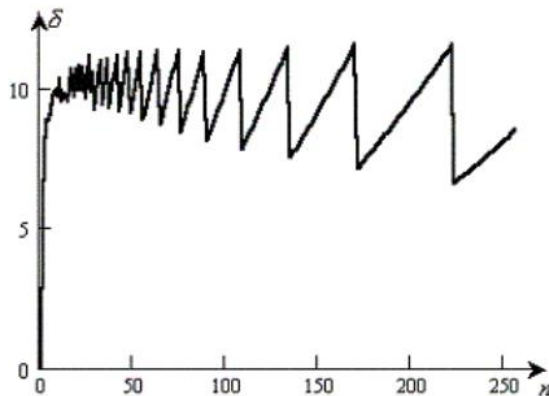


Рисунок 2.3 – Максимальна відносна похибка апроксимації

Основний недолік цієї функції – в зоні затухання відблиску вона спадає до нульового значення дуже повільно, що зумовлює штучне освітлення графічного об'єкта та зайві обчислення за рахунок приросту інтервалу зміни аргументу. Погана точність відтворення зони блюмінгу за моделлю Шліка зумовила певну мету, що полягає в покращенні формули Шліка для більш реалістичного зображення відблисків на поверхні

графічних тривимірних об'єктів. Наступна модифікація функції Шліка дозволить зменшити ефект повільного затухання функції:

$$H(\gamma) = \frac{\cos \gamma}{0.5n(1 - \cos \gamma^2) + (n - n \cos \gamma + \cos \gamma)}, \quad (2.2)$$

Від традиційної функції Шліка дана модифікація відрізняється додаванням в чисельнику виразу  $0.5n(1 - \cos \gamma^2)$ , що забезпечує покращення формування зони затухання функції. На рис. 2.4 зображено графіки трьох функцій: Шліка позначено цифрою 1, Бліна 2, а також модифікованої розробленої функції 3.

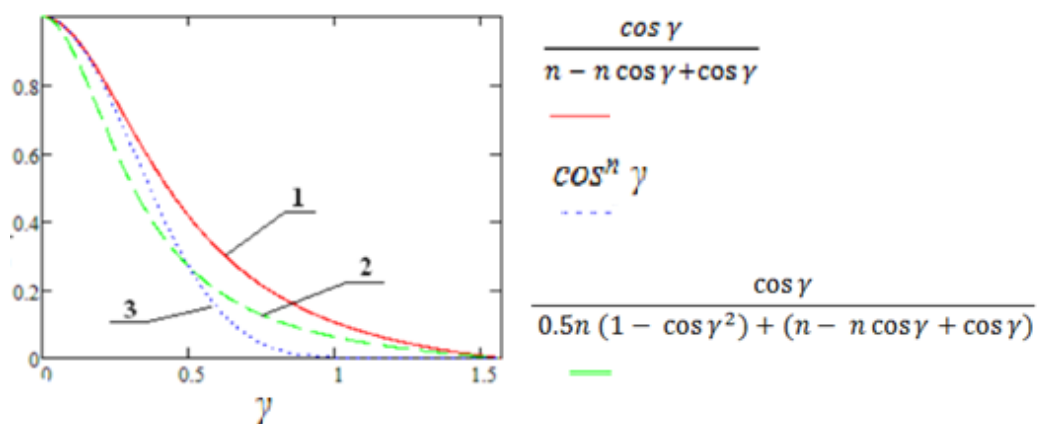


Рис. 2.4 – Порівняння моделей ДФВЗ

Можна зробити висновок, що апроксимована функція забезпечує більш реалістичне відтворення зони блюмінгу. Максимальна відносна похибка апроксимації в цьому випадку відрізняється від оригінальної формули Шліка на 6,8%.

## 2.2 Апроксимація ДФВЗ квадратичною функцією

Проводиться спрощення ДФВЗ квадратичною функцією стосовно  $\cos x$ , тобто

$$\cos^n x = a \times \cos^2 x + b \times \cos x + c, \quad (2.3)$$

У формулі (2.1) невідомими є три коефіцієнти, для визначення яких необхідно розв'язати систему трьох рівнянь. Для знаходження невідомих коефіцієнтів  $a, b, c$  скористаємося точками, значення ДФВЗ в яких можна легко визначити [23].

Розглянемо 2 варіанта значень  $\cos^n x$ . Перше значення  $x = 0$ ,  $\cos^n x = 1$ , тому можна записати, що  $a + b + c = 1$ . Друге,  $\cos x = 0$ , то  $\cos^n x = 0$ , звідси  $a \times 0 + b \times 0 + c = 0$ . Як висновок з цього слідує, що  $c = 0$ ,  $b = 1 - a$ .

Для визначення коефіцієнту  $a$  прирівнюємо значення квадратичної функції та ДФВЗ у точці перегину функції. Двічі продиференціюємо  $\cos^n x$  по  $x$  і прирівнявши до 0 отриману другу похідну визначимо, те що абсциса точки перегину дорівнює:

$$\gamma = \arctg\left(\frac{1}{\sqrt{n-1}}\right), \quad (2.4)$$

Враховуючи формули (2.1) та  $c = 0$ ,  $b = (1 - a)$ , можна записати, що

$$a \times \cos^2 \gamma + (1 - a) \times \cos \gamma = \cos^n \gamma, \quad (2.5)$$

Далі введемо наступні позначення:  $t = \cos x$ ,  $t^n = \cos^n x$ . Формула на основі заміни з урахуванням формули (2.3) має наступний вигляд:

$$a \times t^2 + (1 - a) \times t = t^n, \quad (2.6)$$

Спростивши формулу 2.4 отримаємо

$$a = \frac{t^{n-1} - 1}{t - 1}, \quad (2.7)$$

З вище наведених розрахунків сформуємо загальну формулу для пошуку коефіцієнта  $a$ :

$$a = \frac{\cos^{n-1}\left(\arctg\left(\frac{1}{\sqrt{n-1}}\right)\right) - 1}{\cos\left(\arctg\left(\frac{1}{\sqrt{n-1}}\right)\right) - 1}, \quad (2.8)$$

Таким способом можна знайти невідомі  $a, b, c$  квадратичної функції за допомогою формул (2.3) та (2.8) для даного коефіцієнта спекулярності  $n$ . Розрахування коефіцієнта  $a$  за формулою (2.8) є важко обчислюваною процедурою. Відповідно кожному значенню  $n$  належить одне значення  $a$ , то для більш ефективного обчислення необхідно попередньо визначити значення  $a$  для  $n \in [1; 1000]$  та зберегти їх у створеній таблиці, що значно полегшить процес обчислення ДФВЗ. На рис. 2.5 зображен графік квадратичної апроксимації функції  $f(x) = \cos^n x$  та на рис. 2.6 відносна похибка апроксимації функції  $q(x) = a \times \cos^2 x + b \times \cos x + c$ .

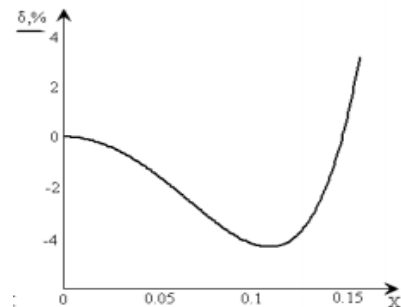
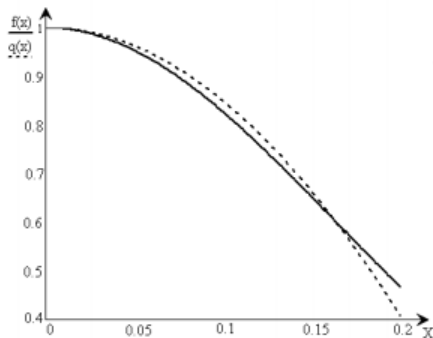


Рисунок 2.5 – Квадратична апроксимація

Рисунок 2.6 – Відносна похибка апроксимації

Установимо граничний рівень 0,56 і вище, зобразимо графік відносної максимальної похибки апроксимації ДФВЗ (рис. 2.7).

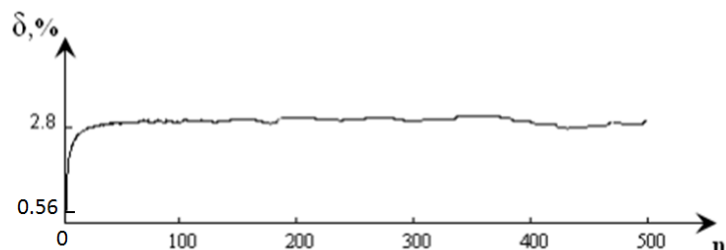


Рисунок 2.7 – Максимальна відносна похибка для  $n \in [1;512]$

На рис. 2.8 наведені приклади формування зображень з використанням квадратичної апроксимації ДФВЗ.



Рисунок 2.8 – Зафарбовування об'єктів з використанням квадратичної апроксимації ДФВЗ

Ця функція має просту апаратну реалізацію та набагато точніше відображає епіцентр відблиску порівняно з функцією Шліка. Недоліком такої функції є те, що вона занадто різко спадає в області формування блюмінгу [4], що обумовлює її використання переважно для моделювання освітлення поверхонь із високим значенням коефіцієнта спекулярності.

Запропонований метод дозволяє достатньо точно (з відносною похибкою не більше 3 %) апроксимувати не лише епіцентр відблиску, а також зону затухання. Порівняно із

існуючими підходами, функція, що пропонується оперує з косинусом кута і має низьку обчислювальну складність.

### 2.3 Модель з використанням косинус – квадратичної функції

У моделі освітлення Фонга інтенсивність спекулярного світла пропорційна до функції  $\cos^n \alpha$ , де значення  $n$  залежить від властивостей поверхні і лежить у діапазоні від 1 до 1000 [24]. Блискучі поверхні мають великі значення  $n$ . Відомо, що  $\cos \alpha = \vec{R} \times \vec{V}$ . Інтенсивність кольору для кожної точки обраховується за формулою:

$$I = K_S I_S (\vec{R} \times \vec{V})^n, \quad (2.9)$$

де  $K_S$  – інтенсивність освітлення;  $I_S$  – вихідний колір об'єкта;  $n$  – коефіцієнт яскравості світіння,  $\vec{V}$  – вектор спостерігача.

Значення  $\vec{R}$  обчислюється за формулою:

$$\vec{R} = 2(\vec{L} \times \vec{N})\vec{N} - \vec{L} \quad (2.10)$$

де  $\vec{L}$  – вектор розташування джерела світла,  $\vec{N}$  – вектор нормалі.

Чим більше значення кута  $\alpha$  між вектором  $V$  і вектором відображення  $R$ , тим яскравіше бікове освітлення (рис. 2.9).

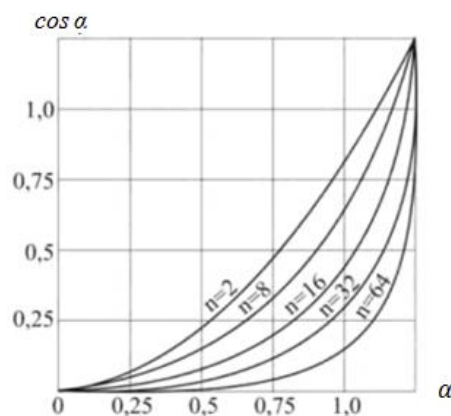


Рисунок 2.9 – Залежність яскравості світіння від кута  $\alpha$

Для зафарбування об'єктів у комп'ютерній графіці найчастіше використовується модель освітлення Бліна, згідно з якою інтенсивність спекулярної складової кольору розраховується по формулі:

$$I = I_l K_S \cos^n \gamma, \quad (2.11)$$

де  $I_l$  – інтенсивність джерела світла,  $K_S$  – коефіцієнт дзеркального відбиття,  $n$  – коефіцієнт спекулярності поверхні, який знаходиться в діапазоні 1–1000,  $\gamma$  – кут між вектором нормалі  $\vec{N}$  до поверхні в заданій точці і вектором півшляху,  $\cos^n \gamma$  – двопротенева функція відбивної здатності поверхні.

Для апроксимації двопротеневої функції  $\cos^n \gamma$  використаємо косинус-квадратичну функцію

$$W(\gamma, n) = (\sigma \times (\cos \gamma - 1) + 1)^2, \quad (2.12)$$

Із рівняння  $(\sigma \times (\cos \gamma - 1) + 1)^2 = 0$  знаходимо, що  $\gamma = \arccos((\sigma - 1)/\sigma)$ . Для запропонованої розроблюваної моделі відбивної здатності поверхні можна записати, що:  $\cos^n \gamma = (\sigma \times (\cos \gamma - 1) + 1)^2$ . Помножимо ліву та праву частини виразу на  $\sin \gamma$  і візьмемо визначений інтеграл окремо від лівої та правої частин отриманого рівняння за умови, що межі інтегрування визначаються відрізком від нульового значення аргументу до точки, де функції приймають нульове значення. Знайдемо визначений інтеграл

$$\int_0^{\arccos(\frac{\sigma-1}{\sigma})} [\sigma \times (\cos \gamma - 1) + 1]^2 \times \sin \gamma d\gamma = \frac{1}{3 \times \sigma}, \quad (2.13)$$

Виконаємо такі ж дії й для

$$\cos^n \gamma \int_0^{\pi/2} \cos^n \gamma \times \sin \gamma \times d\gamma = \frac{\cos^{n+1} \gamma}{n+1} \Big|_0^{\pi/2} = \frac{1}{n+1}, \quad (2.14)$$

Прирівнявши праві частини отриманих виразів, знаходимо, що  $\sigma = (n + 1)/3$ .

Нова формула має такий вигляд

$$\left( \frac{(n+1)}{3} \times (\cos \gamma - 1) + 1 \right)^2, \quad (2.15)$$

Таким чином,  $\cos^n \gamma = \left( \frac{(n+1)}{3} \times (\cos \gamma - 1) + 1 \right)^2$ . Характерною особливістю отриманої формули є те, що згідно з нею, відносно рівномірно розподілено абсолютну похибку на всьому інтервалі визначення.

Для знаходження максимальної відносної похибки апроксимації було знайдено похідну від виразу  $\left[ \left( \frac{(n+1)}{3} \times (\cos \gamma - 1) + 1 \right)^2 - \cos^n \gamma \right] / \cos^n \gamma$  і прирівняно її до нуля. Дослідження показали, що максимальна відносна похибка має місце при  $\gamma = \arctg \left( \frac{(1+2 \times n)^{\frac{1}{2}}}{n} \right)$ . Цю точку розміщено після точки перегину (на рівні 0,2 - 04), в нижній частині зони затухання. Що стосується зони блюмінгу, то максимальна абсолютна похибка апроксимації має місце в точці  $\gamma = \arctg \left( \frac{(6 \times n - 3)^{\frac{1}{2}}}{n-2} \right)$  і не перевищує 0,1 [4, 16]. На рис. 2.10 наглядно продемонстровано залежність максимальної відносної похибки апроксимації від коефіцієнта  $n$ .

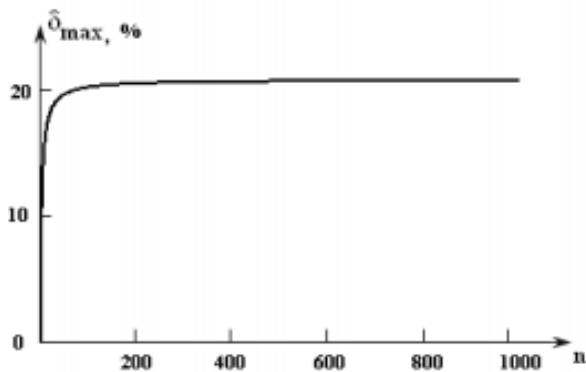


Рисунок 2.10 – Графік залежності максимальної відносної похибки апроксимації від коефіцієнта  $n$

У розділі 3 представлено апаратну реалізацію блоку для обчислення спекулярної складової кольору, який використовує запропоновану апроксимацію.

## 2.4 Тонування за методом Гуро

Процес тонування (зафарбовування) полігонів має найбільшу обчислювальну складність, що відбувається на етапі рендерингу в графічному конвеєрі [25].

Метод тонування Гуро - метод зафарбовування в тривимірній комп'ютерній графіці, призначений для створення ілюзії гладкої криволінійної поверхні, описаної у вигляді полігональної сітки з плоскими гранями, шляхом інтерполяції кольорів сусідніх граней. Метод вперше запропонований Анрі Гуро в 1971 році.

Принцип методу полягає в послідовному обчисленні нормалей до кожної з граней тривимірної моделі, подальшого визначення нормалей вершин шляхом усереднення нормалей всіх прилеглих до вершини граней. Далі на підставі значень нормалей по обраної моделі відображення обчислюється освітленість кожної вершини, яка представляється інтенсивністю кольору в вершині. Розрахунок освітлення, використаний Гуро, був заснований на моделі дифузного віддзеркалення Ламберта.

Багатокутники, що утворюють межі, нарешті закрашуються уздовж ребер шляхом білінійної інтерполяції значень інтенсивності кольору в вершинах.

У методі Гуро розраховуються значення інтенсивності кольору для полігональних вершин, які потім у процесі растеризації лінійно інтерполюються уздовж ребер і рядків сканування [26]. Алгоритмічна та схематична простота методу Гуро, можливість виконання паралельних незалежних обчислень спонукає продовжувати роботи щодо удосконалювання цього підходу. Згідно з методом Гуро кожен полігон зафарбовується

шляхом лінійної інтерполяції значень інтенсивності кольору у вершинах, спочатку уздовж кожного ребра, а потім і між ребрами уздовж кожного скануючого рядка (рис. 2.11).

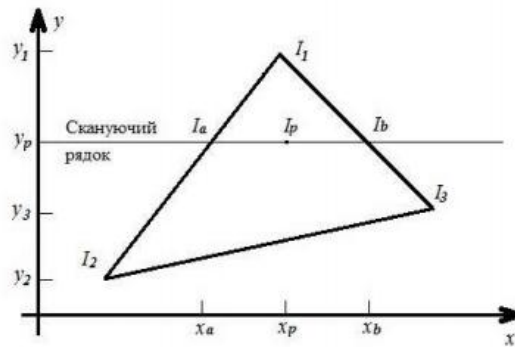


Рисунок 2.11 – Інтерполяція інтенсивності кольору

Для всіх вершин запам'ятовується початкова інтенсивність, а також зміна інтенсивності при кожному одиничному кроці по координаті  $x$ . Заповнення видимого інтервалу на скануючому рядку проводиться шляхом інтерполяції між значеннями інтенсивності на двох ребрах, що обмежують інтервал. Для кольорових об'єктів окремо інтерполюється кожен із компонентів кольору.

При зафарбовуванні за методом Гуро прирости інтенсивності вздовж рядка растеризації трикутника (РРТ) визначаються за формулою

$$\Delta I = \frac{I_b - I_a}{\Delta x}, \quad (2.16)$$

де  $I_b, I_a$  – значення складових інтенсивності кольору відповідно в початковій та кінцевій точках РРТ, який включає всі піксели між лівою та правою точками ребер трикутника в одному із ортогональних напрямків;  $\Delta x$  – довжина рядка растеризації. Аналогічно, можна записати цю формулу у такому вигляді

$$A = \frac{a_2 - a_1}{x_2 - x_1}, \quad (2.17)$$

Остання формула включає трудомістку операцію ділення, яка, безумовно, суттєво впливає на час зафарбовування. Це пояснюється тим, що операція ділення виконується для всіх ребер трикутника і рядків растеризації. Для середньостатистичного трикутника, який включає 100 точок, кількість пікселів у рядку растеризації трикутника, як правило, не перевищує 16 [27]. Однією з властивостей двійкової системи числення є те, що операція ділення на  $2^n$  – це зсув операнда на  $n$  розрядів у сторону молодших. Використовуючи цю властивість, можна замінити операцію ділення на певну кількість зсувів і нагромаджувальних додавань. Зрозуміло, що точність такої процедури залежить від



кількості проведених операцій зсуву. Для реалізації операції ділення можна запропонувати таку апроксимаційну формулу

$$\frac{1}{k} = i_1 \frac{1}{2} + i_2 \frac{1}{2^2} + \dots + i_j \frac{1}{2^j} + \dots + i_n \frac{1}{2^n}, \quad (2.18)$$

де  $k$  – довжина рядка растеризації,  $n$  – кількість операцій зсуву,  $i_j$  – коефіцієнт, що приймає значення «0» або «1» залежно від наявності степеневого доданку в апроксимаційній формулі. Розглянемо використання формули (2.18) для конкретного випадку. Нехай  $k = 7$  та  $n = 6$ :

$$\frac{1}{7} \approx \frac{1}{8} + \frac{1}{64}, \quad (2.19)$$

При інтерполяції інтенсивності кольору, згідно формули (2.16), максимальне значення приросту інтенсивності кольору для режиму TrueColor буде мати при  $I_a = 0$ ,  $I_b = 255$ . Тобто

$$\Delta I = \frac{255 - 0}{7} = 36,4286$$

При використанні апроксимаційної формули (2.19)

$$\Delta I = (255 - 0) \times \left( \frac{1}{8} + \frac{1}{64} \right) = 35,8594$$

Абсолютна похибка обчислень, для випадку, що розглядається, становить

$$\Delta_{\max} = 36,4286 - 35,8594 = 0,5692$$

Відносна похибка обчислень дорівнює

$$\delta_{\%} = \frac{0,5693}{36,4286} \times 100\% = 1,56\%$$

Даний результат показує, що при використанні для операції ділення апроксимаційної формули, якість сформованого зображення на екрані комп'ютера візуально не погіршиться.

При застосуванні апроксимаційних формул, що враховують 8 операцій зсуву, максимальна абсолютна похибка обчислень становить  $\Delta_{\max} = 0,4427$ , а середня відносна похибка  $\delta_{\%} = 0,81\%$ . На рисунку 2.12 наведено графік відносних похибок для  $n = 6$ ,  $n = 7$  та  $n = 8$ .

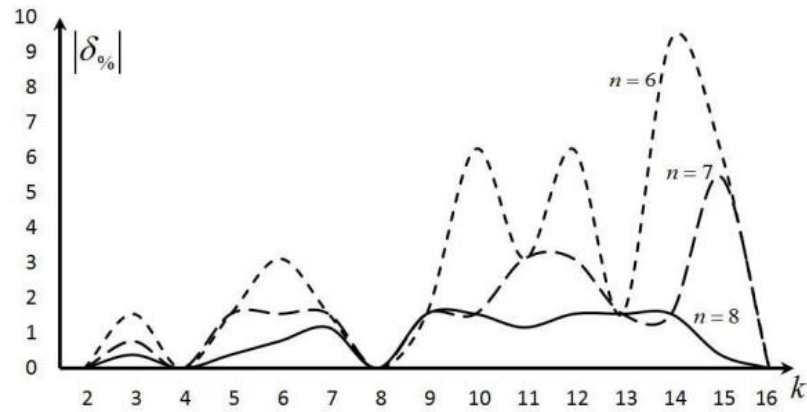


Рисунок 2.12 — Графік відносних похибок формул апроксимації

Проаналізуємо трудомісткість процедури зафарбовування за методом Гуро для середньостатистичного трикутника, який включає 100 точок [28]. Використаємо для розрахунків трикутник ABC з  $\Delta x_{AB} = 7$ ,  $\Delta y_{AB} = 8$ ,  $\Delta x_{AC} = 9$ ,  $\Delta y_{AC} = 12$ ,  $\Delta x_{BC} = 16$ ,  $\Delta y_{BC} = 4$ . Для оцінки трудомісткості виконання процедури зафарбовування трикутника за методом Гуро використано кількість тактів процесора Pentium M виконання складових інструкцій. Інструкція задає одну окрему операцію процесора, визначену системою його команд. У табл. 2.1 наведено кількість необхідних для розрахунку операцій відповідно до кожного етапу зафарбовування.

Таблиця 2.1 — Процеси та кількість інструкцій для зафарбовування трикутника за методом Гуро

Процедура	Кількість інструкцій	
	«+» або «-»	«÷»
Розрахунок приростів складових інтенсивності кольору вздовж ребер	18	9
Розрахунок інтенсивності складових кольору точок ребер	99	—
Розрахунок приростів інтенсивності складових кольору рядків пастеризації	66	33
Розрахунок інтенсивності складових кольору точок рядків пастеризації	225	—
Разом:	408	42

Структурну схему пристрою для виконання операції ділення за формулою (2.18), зображено на рис. 2.13

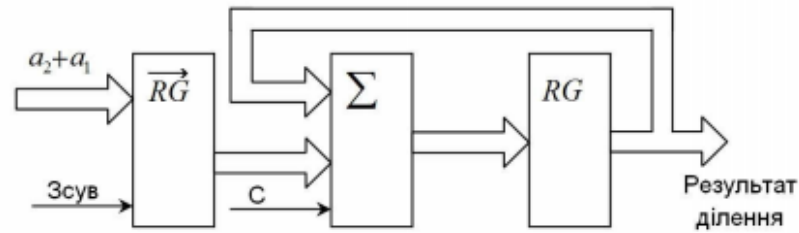


Рисунок 2.13 — Пристрій, що виконує операцію ділення за апроксимаційними формулами

Пристрій складається із зсувного регістра  $\overline{RG}$  і нагромаджувального суматора, утвореного регістром RG і комбінаційним суматором  $\Sigma$ . Значення  $a_1 + a_2$  поступає на вхід зсувного регістра, який використовується для знаходження доданків, які поступають на вхід нагромаджувального суматора. Додавання у нагромаджувальному суматорі виконується тільки при умові, що  $i_j = 1$ . Якщо поточне значення  $i_j = 0$ , то виконується тільки зсув.

Виконання наведених у таблиці операцій при використанні процесора Pentium M потребує 2946 тактів. При виконанні наведених операцій за запропонованим підходом, реалізованим програмно, необхідно 2064 такти, що свідчить про підвищення швидкодії зафарбовування.

Зображені на рис. 2.14 об'єкти, сформовані з урахування перспективи графічного об'єкта

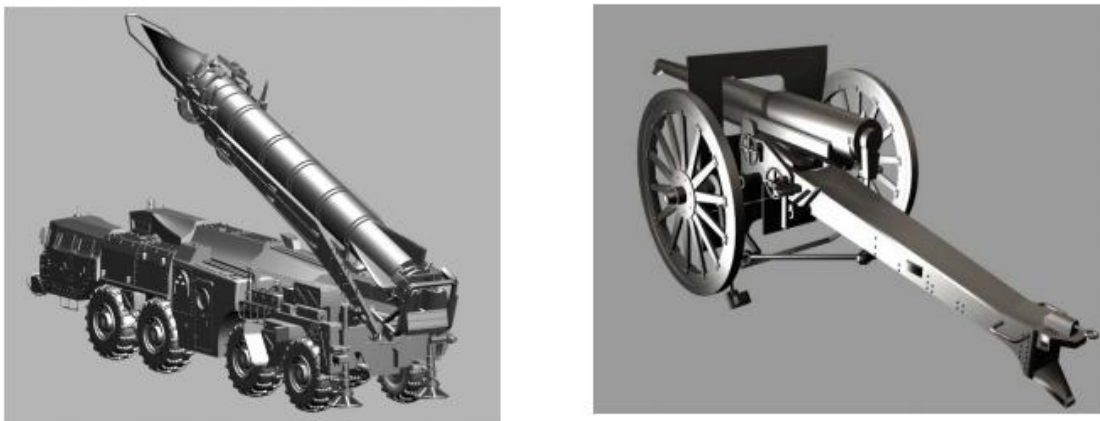


Рисунок 2.14 – Зображення тривимірних фігур із урахування їх перспективи

Даний підхід направлений на підвищення швидкодії зафарбовування методом заміни складної операції ділення спрощеною формулою, що передбачає певну кількість накопичувальних додавань та зсувів [29]. Досягається підвищення швидкодії в 1,43 рази при реалізації даного підходу у методі Гуро.

## 2.5 Створення енергетично-коректної двопроменевої функції відбивної здатності поверхні

Фізично точні ДФВЗ мають властивість оберненості та збереження енергії [1, 30]. Відношення диференційного випромінювання поверхні, враховуючи принцип оберненості, до диференційної енергетичної освітленості, що задається ДФВЗ, для усіх пар значень та має залишитись вільним при їх взаємній заміні, формула має вигляд:

$$f_r(\omega_i, \omega_o) = f_r(\omega_o, \omega_i), \quad (2.20)$$

де  $\omega_i$  - напрямок падіння променя на поверхню,  $\omega_o$  - напрямок відбиття променя.

Відповідно закону збереження енергії сукупна енергія світлового потоку, що падає на поверхню об'єкта, повинна бути рівною або меншою енергії відбитого світлового потоку. Якщо ДФВЗ не є енергетично точною, об'єкт набуває властивість випромінювача, що негативно впливає на загальний реалістичний вигляд тривимірної сцени. Виконання функцією закону збереження енергії є дуже важливим при використанні алгоритмів глобального освітлення. Введемо поняття направленої напівсферичної інтегральної відбивної здатності поверхні (hemispherical-directional reflectance) для проведення досліджень енергетичної коректності ДФВЗ. Ця величина визначається функцією, що характеризує загальну відбивну здатність поверхні при освітленні одним променем, що падає у напрямку  $\vec{L}$  [31] та задається таким чином:

$$R(\vec{L}) = \int f_r(\vec{L}, \vec{V}) \cos \theta_0 d\omega_0, \quad (2.21)$$

де  $\theta_0$  – кут між нормаллю  $\vec{N}$  та поточним напрямком до спостерігача  $\vec{V}$ .

Щоб ДФВЗ стала енергетично-коректною, для цього необхідно аби величина напівсферичної інтегральної відбивної здатності поверхні була меншою або рівною одиниці:

$$R(\vec{L}) = \int f_r(\vec{L}, \vec{V}) \cos \theta_0 d\omega_0 \leq 1, \quad (2.22)$$

Значення  $R(\vec{L}) = 0$  характеризує поверхню із повним поглинанням всього падаючого світлового потоку, а  $R(\vec{L}) = 1$  абсолютно дзеркальну поверхню. Для запропонованої ДФВЗ (3.7) умова відповідності закону збереження енергії має вигляд:

$$\int \left( \frac{n+1}{3} \times (\cos \theta_h - 1) + 1 \right)^2 \cos \theta_0 d\omega_0 \leq 1, \quad (2.23)$$

де  $\theta_h$  – кут між векторами  $\vec{H}$  і  $\vec{N}$ .

Згідно до закону косинусів Ламберта, найбільш освітленість поверхні променем відбувається при умові нормального падіння променя, тобто коли  $\vec{N} = \vec{L}$ . При такій умові

кут між векторами  $\vec{H}$  і  $\vec{N}$  дорівнює половині кута між нормаллю  $\vec{N}$  та поточним напрямком до спостерігача  $\vec{V}$ . За таких умов вираз (3.8) виглядає наступним чином:

$$\int \left( \frac{n+1}{3} \times \left( \frac{\cos \theta_0}{2} - 1 \right) + 1 \right)^2 \cos \theta_0 d\omega_0 \leq 1, \quad (2.24)$$

Для урахування лише першого періоду спрощеної функції (3.9) необхідно проводити інтегрування в межах  $\theta \in \left[ 0; \frac{5.8}{\sqrt{n}} \right]$ :

$$2\pi \int_0^{\frac{5.8}{\sqrt{n}}} \left( \frac{n+1}{3} \times \left( \cos \frac{\theta_0}{2} - 1 \right) + 1 \right)^2 \cos \theta_0 \sin \theta_0 d\omega_0 \leq 1, \quad (2.25)$$

На рис. 2.15 наведено графік результатів інтегрування та визначення направленої напівсферичної інтегральної відбивальної здатності поверхні для запропонованої ДФВЗ.

На рис. 2.15 видно, що ДФВЗ не є енергетично-коректною для значень  $n \in [1; 17]$ . Для усунення цієї помилки необхідно ввести нормалізаційний коефіцієнт, що дозволить умові (2.25) виконуватись для  $n \in [1; 1000]$ . Проаналізувавши форму кривої на рис. 2.15 та врахувавши вигляд нормалізаційного коефіцієнту в ДФВЗ Бліна, дійдемо до висновку, що коефіцієнт для запропонованої ДФВЗ має вигляд  $\frac{n+A}{B}$ , де А і В – деякі сталі.

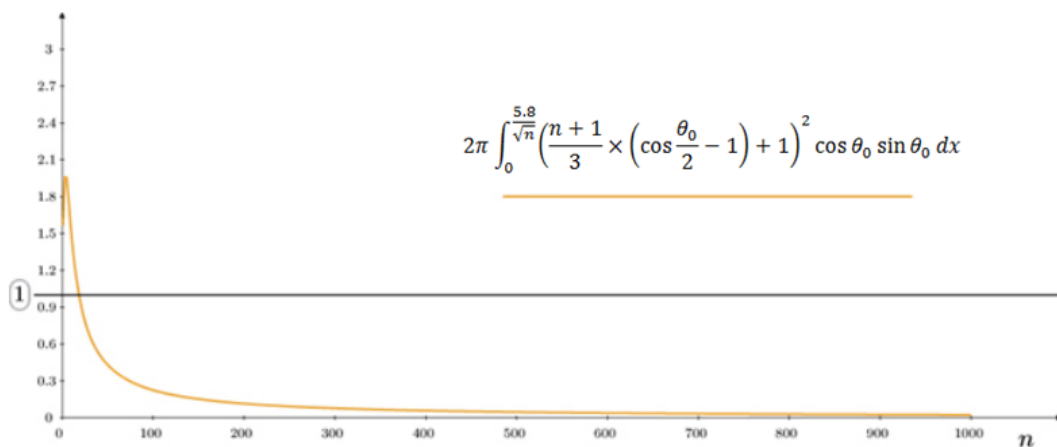


Рисунок 2.15 – Результати виразу (2.24) при інтегруванні для  $n \in [1; 1000]$

Використовуючи функціонал пакету MathCAD [32] при апроксимації кривих, знайдено значення коефіцієнтів  $A = 6$  та  $B = 24.55$ . В результаті, запропонована енергетично-коректна ДФВЗ має такий вигляд:

$$F(n, \gamma) = \frac{n+6}{24.55} \times \left( \frac{n+1}{3} \times (\cos \gamma - 1) + 1 \right)^2, \quad (2.26)$$

На рис. 2.16 наведено результати інтегрування ДФВЗ (2.26) та ДФВЗ Бліна з нормалізаційним коефіцієнтом  $\frac{n+6}{8\pi}$

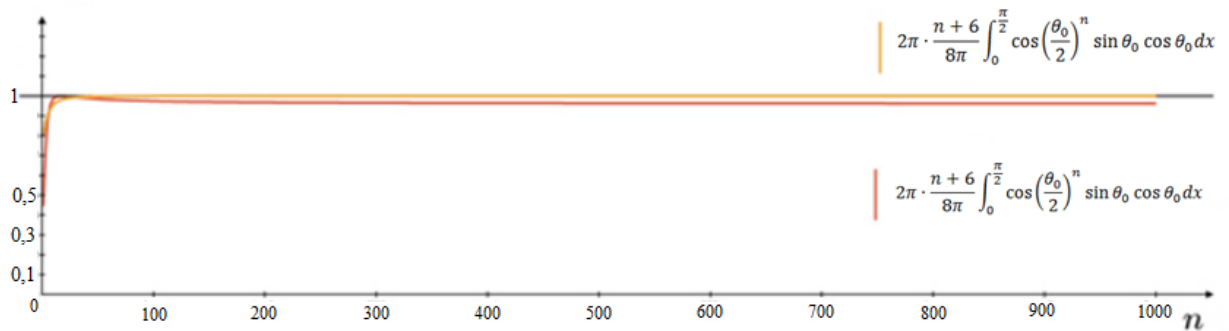


Рисунок 2.16 – Результат величини напівсферичної інтегральної відбивальної здатності поверхні для  $n \in [1;1000]$ .

З наведених на рис. 2.16 даних можна зробити висновок, що запропонована ДФВЗ (2.26) є енергетично-коректною для  $n \in [1;1000]$ , що дозволяє використовувати її у алгоритмах розрахунку глобального освітлення. На рис. 2.17 наведено приклад сформованого зображення з використанням енергетично-коректної двопроменевої функції відбивної здатності поверхні.



Рисунок 2.17 – Приклад формування тривимірного об'єкту з використанням енергетично-коректної ДФВЗ

## 2.6 Апроксимований метод з'єднання зображення на основі алгоритму зворотного трасування променів

Розроблений спосіб поєднання зображення, що базується на інтерполяції та алгоритмі зворотного трасування променів. Найбільш поширені методи інтерполяції: лінійна, білінійна, кубічна. Лінійна інтерполяція – інтерполяція алгебраїчним двочленом

функції  $f$ , заданої в двох точках  $x_0$  і  $x_1$  відрізка  $[a, b]$ . При використанні невеликих об'ємів сегментів внутрішніх пікселів для розрахунку колірних компонентів досить використовувати найпростіше усереднення:

$$C_{cp} = \frac{C_l + C_r}{2}, \quad (2.27)$$

де  $C_{cp}$  – колір внутрішніх пікселів сегмента,  $C_l$  – колір зовнішнього лівого пікселя,  $C_r$  – колір зовнішнього правого пікселя [33].

Білінійна інтерполяція – розширення лінійної інтерполяції для функцій з двома змінними. Основна ідея способу зводиться до проведення звичайної лінійної інтерполяції спочатку в одному напрямку, та навпаки. Спосіб білінійної інтерполяції використовує матрицю  $2 \times 2$  чи  $4$  сусідніх пікселя, оточуючих шуканий піксель для отримання нового значення. Білінійна інтерполяція включає обробку відомого середнього значення яскравості безлічі пікселів, заснованого на дробовій частині обчислюваного пікселя.

Кубічна інтерполяція — інтерполяція функції однієї змінної поліномом третього степеня. Вона визначає значення рівня сірого від середньозважених 16 найближчих пікселів для зазначених вхідних координат пікселів і присвоює це значення вихідним координатам. Зображення виходить різкіше, ніж при білінійній інтерполяції, і воно не має таких артефактів як при інтерполяції методом найближчого сусіда.

При кубічній інтерполяції в одновимірному просторі використовуються чотири точки, по дві з кожного боку від інтерпольованої. Така інтерполяція проходить в одному напрямку (горизонтальному), на відміну від інтерполяції в двовимірному просторі, для якої необхідно ще й перпендикулярний напрямок та 16 точок. Кубічна інтерполяція в двовимірному просторі називається бікубічною. На рисунку 2.18 зображені інтерпольовані точки.

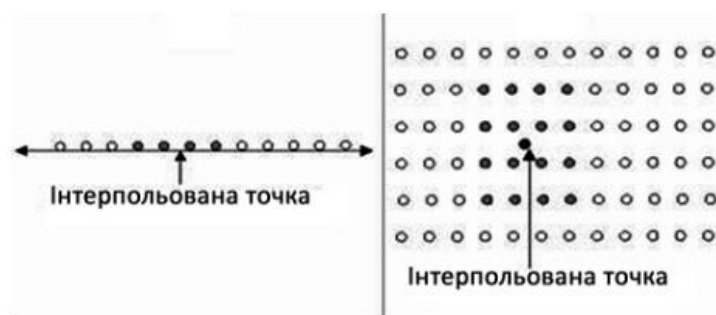


Рисунок 2.18 – Точки сітки для кубічної (зліва) та для бікубічної (справа) інтерполяції

Процес з'єднання зображення на основі алгоритму зворотнього трасування променів із застосуванням рядкової інтерполяції складається з наступних пунктів:

- а) колір лівого пікселя сегмента приймається рівним результату попереднього трасування;
- б) виконується трасування променя через правий піксель сегмента;
- в) розраховуються колірні компоненти внутрішніх пікселів шляхом лінійної інтерполяції за формулою 2.27.

Нижчеперераховані етапи обробки одного фрагмента рядка:

- задається довжина сегмента рядка або фрагмента рядка екрану;
- виконується трасування 1-го пікселя рядка;
- у циклі відбувається перебір піксельних сегментів:

При цьому необхідна кількість умовних процесорних елементів для трасування одного рядка буде визначатися як:

$$N_{PE_t} = \left\lceil \frac{X_{max}}{L_{segm} + 1} + 1 \right\rceil, \tag{2.28}$$

де  $X_{max}$  - розмір екрану по горизонталі, пікселів,  $L_{segm}$  - довжина сегменту пікселів.

Загальна кількість процесорних елементів буде визначатись так:

$$N_{PE} = N_{PE_t} \times Y_{max}, \tag{2.29}$$

де  $Y_{max}$  - розмір екрану по вертикалі, пікселів,  $N_{PE}$  – кількість процесорних елементів.

Введемо додаткову перевірку контролю збереження якості зображення. Розглянемо більш довгий сегмент рядка, що містить 8 пікселів. Нехай  $PE1$  і  $PE2$  – віртуальні процесорні елементи, які виконують трасування. Тоді процес обробки буде виглядати таким чином (рис. 2.19) [34].

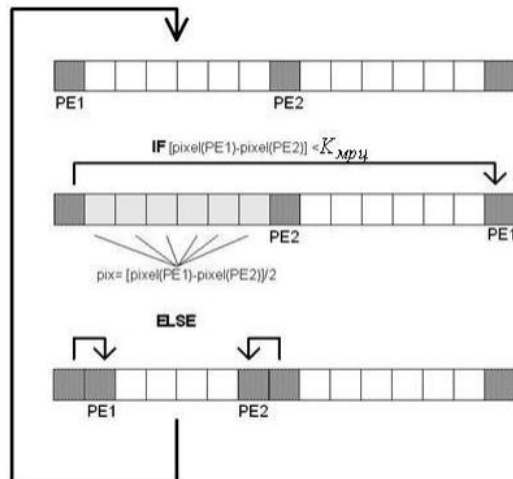




Рисунок 2.19 – Покрокова обробка пікселів

При порівнянні колірних параметрів генерується перемикач *InterpolOK*, який приймає значення 1 при їх допустимому розходженні, інакше - 0. Якщо перемикач встановлено, то виконується перехід на інтерполяцію. Інакше проводиться перезавантаження пари процесорних елементів і перерахунок відповідних пікселів сегмента, зазвичай, через центральний піксель. Операція завершується по *InterpolOK* = 1.

Враховуючи те, що в розрахунок колірних компонент інтерпольованої пікселів вводиться логічна вершина, яка порівнює розбіжність у колірних компонентах граничних пікселів сегмента з коефіцієнтом *K*, то формула (2.27) прийме вигляд (2.30):

$$C_{cp} = \left\{ \begin{array}{l} \frac{C_l + C_r}{2}, \text{ if } K \leq 2,3 \\ RayTracer(), \text{ if } K > 2,3 \end{array} \right\}, \quad (2.30)$$

де  $C_{cp}$  – колір внутрішніх пікселів сегмента, *RayTracer()* – виконання трасування через центральний піксель сегменту,  $C_l$  – колір зовнішнього лівого пікселя,  $C_r$  – колір зовнішнього правого пікселя. Таким чином, цикл перебору піксельних сегментів рядка (або її фрагмента) виконуватиметься таким чином:

- а) колір лівого пікселя сегмента приймається рівним результату попереднього трасування;
- б) виконується трасування променя через правий піксель сегмента;
- в) розраховується коефіцієнт відмінності в кольорі лівого і правого пікселя сегмента;
- г) якщо значення коефіцієнта менше допустимого, то розраховуються колірні компоненти внутрішніх пікселів, інакше довжина сегмента (крок трасування) зменшується вдвічі, і виконується перехід на крок б).

Таким чином, можуть використовуватися сегменти пікселів змінної довжини: для фонових і граничних ділянок зображення – більшого розміру, а для центральної частини екрану – меншого. Необхідна кількість умовних процесорних елементів для трасування одного рядка  $N_{PE_r}$  розміром  $X_{max}$  пікселів буде визначатися таким чином:

$$N_{PE_r} = \left\lceil \frac{X_{max}}{\sum_{i=0}^{K_S} L_{segm_i} + 1} \right\rceil + 1, \quad (2.31)$$

де  $L_{segm_i}$  - довжина *i*-го сегменту пікселів,  $K_S$  – кількість піксельних сегментів.

## 2.7 Висновки до другого розділу

У розділі було запропоновано розробку нових моделей відбивної здатності поверхні, а саме:

1) розроблено апроксимацію функції Шліка. Результати досліджень показали, що при використанні модифікації Шліка максимальна відносна похибка при відтворенні відблиску складає не більше 3,2% порівняно з 10% при використанні оригінальної формули Шліка. Крім того, значно точніше відтворюється зона відблиску, серед недоліків: підвищення складності розрахунків.

2) запропоновано апроксимацію ДФВЗ квадратичною функцією. Метод має значно меншу обчислювальну складність і може ефективно використовуватись для відтворення металевих поверхонь

3) було розроблено нову модель відбивної здатності поверхні на основі еталонної моделі Фонга. Нова модель описується функцією:  $\cos^n \gamma = \left( \frac{(n+1)}{3} \times (\cos \gamma - 1) + 1 \right)^2$ . Для обчислення запропонованої функції використовується друга степінь. Таким чином, спостерігається суттєве підвищення продуктивності порівняно з використанням функції Бліна. Запропонований метод дозволяє достатньо точно (з відносною похибкою не більше 3%) апроксимувати не лише епіцентр відблиску, а також зону затухання.

А також проведено такі дослідження:

1) запропоновано методи коректного відтворення кольорів при зафарбовуванні тривимірних графічних об'єктів, які основані на лінійному інтерполюванні інтенсивностей складових кольору і враховують перспективне проектування, що дозволяє підвищити реалістичність формування графічних сцен за рахунок установлення відповідності кольору точок поверхонь в об'єктній та екранній системах координат;

2) запропоновано модифікацію двопроменевої функції відбивної здатності поверхні для систем високореалістичної графіки, що є енергетично-коректною, тобто забезпечує виконання закону збереження енергії при відбитті світла від поверхні об'єкта. Знайдено нормувальний коефіцієнт для приведення вихідної функції до енергетично-коректної;

3) розроблено метод синтезу зображень на основі алгоритму зворотного трасування променів, який використовує піксельні сегменти або піксельні блоки і додає міжпіксельну інтерполяцію в кінець трасування поточного променя або групи променів відповідно.

На основі вищепроведених розробок є необхідність реалізувати дані у графічних засобах комп'ютерної графіки з проведенням експериментів.

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ У ГРАФІЧНИХ ЗАСОБАХ КОМП'ЮТЕРНОЇ ГРАФІКИ. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

#### 3.1 Апаратна реалізація ДФВЗ

У другому розділі було проведено аналіз існуючих моделей, а також розроблено нову модель відбивної здатності поверхні на основі еталонної моделі Фонга. Розроблена модель описується функцією:  $\cos^n \gamma = \left( \frac{(n+1)}{3} \times (\cos \gamma - 1) + 1 \right)^2$ . Упровадження та реалізація даної моделі представляє собою проектування пристрою для визначення спекулярної складової кольору та відбивної здатності поверхні у кожній точці сцени [35]. Принципова схема, що реалізує дану функцію зображена на рис. 3.1. Пристрій для реалізації розробленої моделі містить два блоки постійної пам'яті, два суматори, п'ять регістрів, п'ять блоків множення, схему порівняння і блок ключів.

Пристрій працює наступним чином. У другий регістр під номером 2 записується значення коефіцієнта  $n$  спекулярності поверхні. Для цього значення коефіцієнта подається на інформаційний вхід 17, а на другий керуючий вхід під номером 18 - рівень логічної одиниці. У перший регістр під номером 1 на рис. 3.1 записується значення косинуса кута  $\cos \gamma = \vec{N} \times \vec{N}$  між вектором нормалі  $\vec{N}$  та вектором  $\vec{N}$ . Значення  $\cos \gamma$  подається на перший інформаційний вхід під номером 32. У третій регістр записується значення інтенсивності  $R$ , в четвертий – значення інтенсивності  $G$  та в п'ятий – значення інтенсивності  $B$  складових кольору, помножених на коефіцієнт  $K_S$  дзеркального відбиття поверхні. Значення коефіцієнта спекулярності  $n$  з другого регістру номером 2 поступає на вхід першого блоку постійної пам'яті, що показаний під номером 9 (зображено на рис. 3.1), де зберігаються відповідні граничні значення ДФВЗ, після яких отриманні значення необхідно обнулити. Граничне допустиме значення ДФВЗ з виходу першого блоку постійної пам'яті поступає на перший вхід схеми порівняння під номером 11, на другий вхід якої поступає вираз  $\left( \frac{(n+1)}{3} \times (\cos \gamma - 1) + 1 \right)^2$ . Він формується таким чином, значення  $\cos \gamma$  з регістру 1 подається на перший вхід суматора 6, на кожний розряд другого входу якого подається код одиниці, що забезпечує віднімання від операнду значення 1. Дійсно, прямий код 1 - 000...01, зворотній код 1 - 111...10. Тому, в молодшому розряді зворотного коду 1 завжди присутній «0». Для отримання додаткового коду 1 потрібно до молодшого розряду додати 1, тобто доповняльний код 1 дорівнює 111... 11.

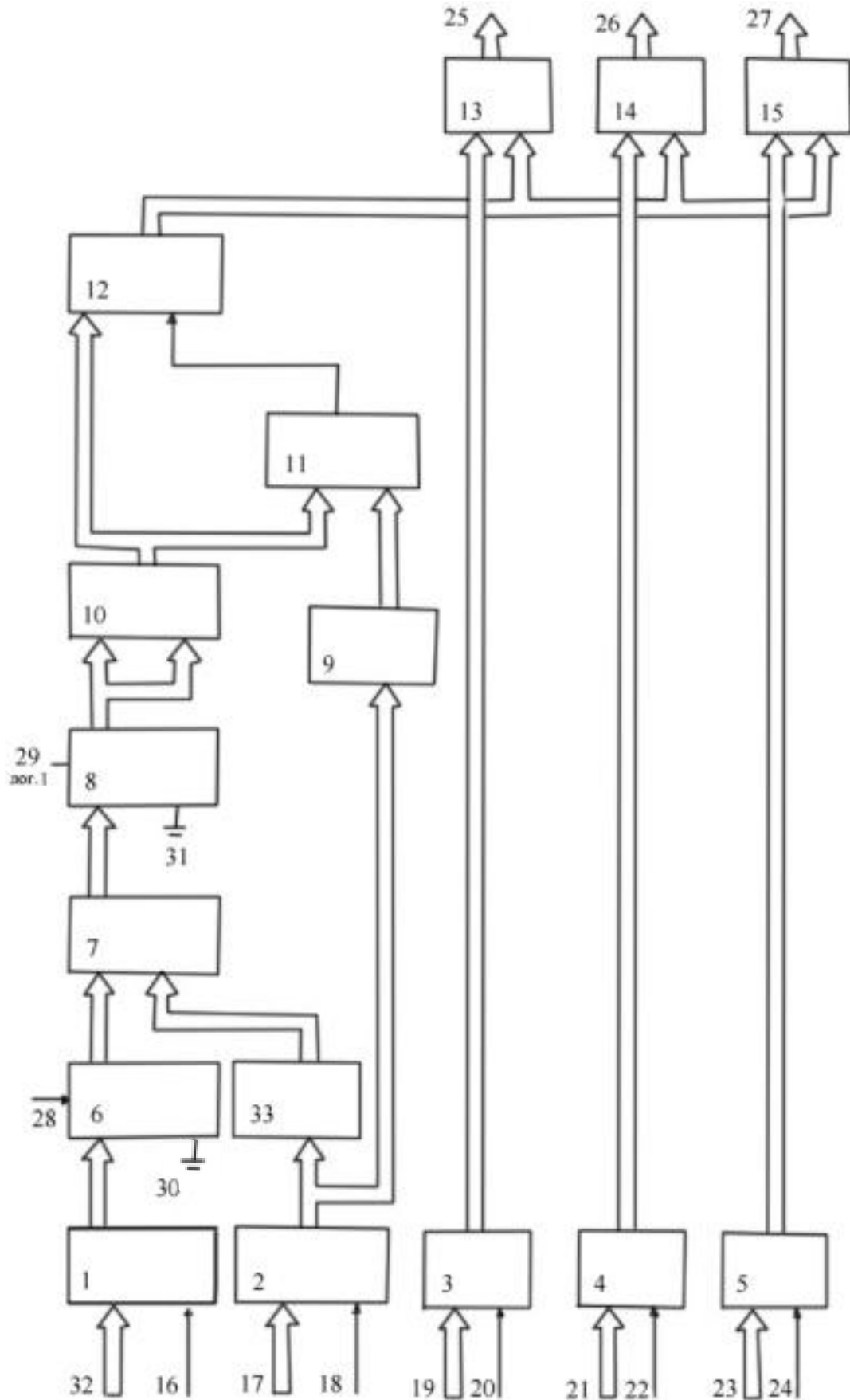


Рисунок 3.1 – Схема пристрою для формування відблиску

Вхід переносу суматора номером 6 заземлено, що забезпечує формування на його виході значення  $\cos \gamma - 1$ , що поступає на перший вхід блоку множення. Значення коефіцієнта спекулярності  $n$  з регістру 2 поступає на вхід другого блоку 33 постійної пам'яті, де зберігаються узгодженні значення  $\frac{(n+1)}{3}$ . На другий вхід блоку 7 множення подається значення з другого блоку пам'яті 33. Другий вхід другого суматора 8 заземлено, а на вхід переносу 29 подано рівень логічної одиниці, тому на виході суматора 8 формується значення  $\frac{(n+1)}{3} \times (\cos \gamma - 1) + 1$ , що подається на входи блоку множення 10, на виході якого отримаємо вираз  $\left(\frac{(n+1)}{3} \times (\cos \gamma - 1) + 1\right)^2$ , що подається на схему порівняння 11 і на блок ключів 12.

У функції Бліна використовується ДФВЗ виду  $\cos^n \gamma$ , де  $n$  належать значення від 1 до 1000. Для запропонованої функції використовується друга степінь. Таким чином, спостерігається суттєве підвищення продуктивності у порівнянні з використанням функції Бліна. Запропонований метод дозволяє достатньо точно (з похибкою не більше 3%) апроксимувати епіцентр відблиску та зону затухання. Функція, що пропонується, має низьку обчислювальну складність, порівняно з існуючими методами.

При використанні ДФВЗ  $\left(\frac{(n+1)}{3} \times (\cos \gamma - 1) + 1\right)^2$  абсолютна похибка апроксимації епіцентра відблиску в чотири рази менша порівняно з ДФВЗ Шліка, причому для її розрахунку не виконується трудомістка операція ділення. Порівняно із ДФВЗ Шліка запропонована функція при програмній реалізації розраховується швидше в 1,7 рази. Комп'ютерне моделювання показало, що для функції  $\left(\frac{(n+1)}{3} \times (\cos \gamma - 1) + 1\right)^2$  не спостерігається візуальних відмінностей між тестовим фігурами, про що свідчить значення нормалізованої середньоквадратичної помилки  $NMSE$  (normalized mean square error), яке менше за 0,0001 [35].

### 3.2 Розробка шейдерних програм та плагінів на базі отриманих результатів

Для розробки програмного забезпечення було обрано мову програмування Java [36], для розробки програмних шейдерів обрано мову HLSL [37]. Мова програмування Java – об'єктно-орієнтована мова програмування. Всі сутності в мові Java є об'єктами, за винятком деяких основних типів (primitive types) [36]. Програми на Java транслюються в байт-код, що виконується віртуальною машиною Java (JVM).

Java – незалежна від платформи мова програмування. Таким чином, код програми можна запускати під управлінням операційних систем Windows, Linux, FreeBSD, Solaris,

Apple Mac та ін. Важливою перевагою мови програмування Java є досить висока безпека. Так як усі програми виконуються і контролюються Java-машиною, то при будь-якій спробі несанкціонованого доступу або з'єднання програма миттєво перериває свою роботу. Крім цього, існує величезна кількість безкоштовних програмних бібліотек доступних до мови Java. Основним недоліком мови є надмірне навантаження на оперативну пам'ять при виконанні програм.

Для програмної реалізації розроблених функцій відбивної здатності поверхні використовується шейдерна мова високого рівня HLSL – High Level Shader Language. Дана мова була створена корпорацією Microsoft і включена в пакет DirectX 12.0. Мова HLSL підтримує скалярні й векторні типи, а також матриці та структури.

Шейдерна мова HLSL багато в чому нагадує мову програмування C та дозволяє розробникам писати код мовою, що транслює його в асемблер, який сприймається графічним пристроєм. Завдяки HLSL можна отримати доступ до усіх функцій відеокарти.

Існує декілька версій HLSL, які мають відмінність за набором команд і системним вимогам, та усі вони підтримують розробку вершинних і піксельних шейдерів. Вершинний шейдер виконує обробку геометрії – змінює параметри вершини, такі як: позиція, текстурні координати, колір вершини. Допустима кількість команд досягає однієї-двох сотень. Піксельний шейдер проводить операції з кольором і текстурами, кількість інструкцій значно обмежена. Шейдерна високорівнева мова HLSL дозволяє використовувати структури, процедури і функції, створювати шейдери, які мають доступ до всього функціоналу графічних пристроїв [37].

Для розробки програмного забезпечення було обрано середовище розробки IntelliJ IDEA. IntelliJ IDEA – комерційне інтегроване середовище розробки для Java від компанії JetBrains. Перша версія IntelliJ IDEA з'явилася у січні 2001 року й дуже швидко здобула популярність, як перша Java IDE з широким набором інтегрованих інструментів для декомпозиції програми на максимально автономні та елементарні частини. Дизайн середовища дозволяє продуктивно працювати програмістам, дозволяючи їм сконцентруватися на розробці функціональності, тоді як IntelliJ IDEA бере на себе виконання рутинних операцій [38]. Інтерфейс середовища IntelliJ IDEA показано на рис. 3.2.

Для розробки вершинних та піксельних шейдерів обрано потужний графічний конвеєр середовища Render Monkey 1.80 компанії ATI Technologies шляхом створення вершинних і піксельних шейдерів, які передбачені специфікаціями DirectX HLSL.

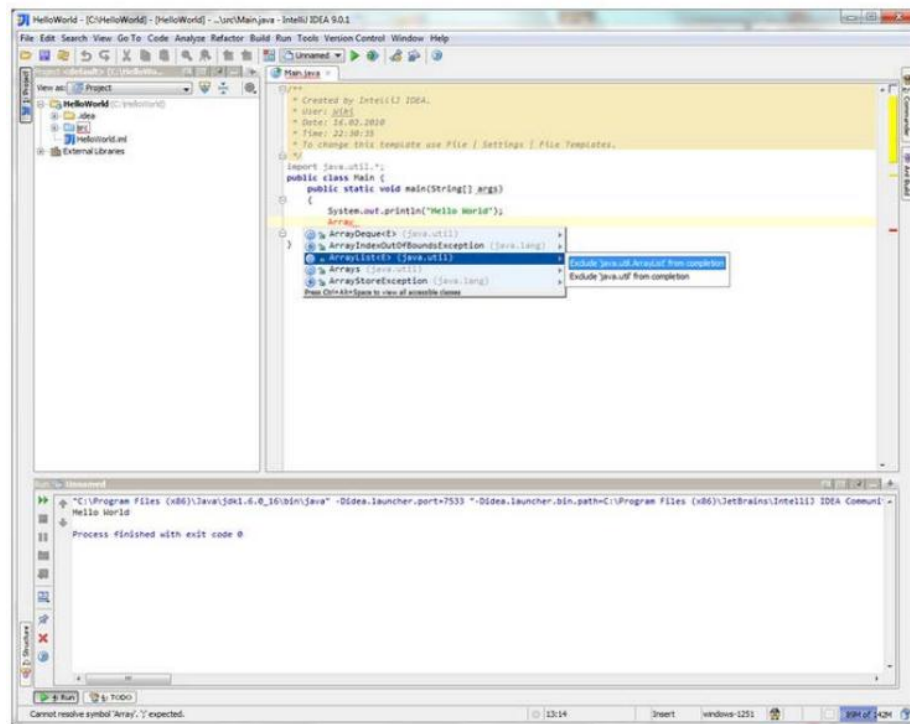


Рисунок 3.2 – Інтерфейс середовища IntelliJ IDEA

Запропоновані програмні продукти можуть бути інтегровані в професійному пакеті Autodesk 3ds Max, за допомогою установки відповідних плагінів.

Плагін – незалежно компільований програмний модуль, що динамічно підключається до основної програми і призначений для розширення та/або використання її можливостей. Плагіни зазвичай виконуються у вигляді бібліотек загального користування [39].

Основний додаток надають сервіси, які плагін може використовувати, наприклад, для можливості реєстрації себе в основному додатку, а також протокол обміну даними з іншими плагінами. Плагіни є залежними від сервісів, що надаються основним додатком і часто окремо не використовуються. На відміну від них, основний додаток незалежно оперує плагінами, надаючи кінцевим користувачам можливість динамічно додавати і оновлювати плагіни без необхідності внесення змін в основний додаток. Autodesk 3ds Max SDK дозволяє створювати безліч різноманітних типів плагінів. Серед них такі як: рендери, модифікатори, плагіни матеріалів, текстур і атмосфери, і, звичайно ж, імпортери та експортери геометрії. Всі плагіни представляють собою звичайну DLL (динамічно підключаему бібліотеку) яка експортує декілька функцій. Для кожного типу плагіна передбачено своє розширення для імені файлу, таким чином Autodesk 3ds Max розпізнає їх тип під час завантаження. Для експортерів зумовлено розширення <.dle>.

У Autodesk 3ds Max [40] до першої групи плагінів відносяться стандартні плагіни, що зберігаються в папці stdplugins і завантажуються при запуску програми. Їх список відображається в наведеному на рис. 3.3 діалозі Plug-in Manager (меню Customize - Plug-in Manager).

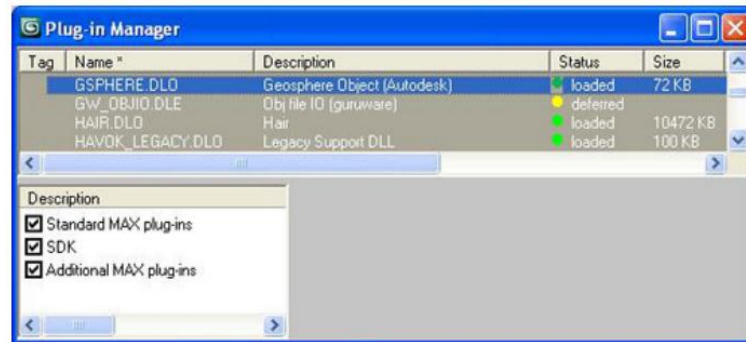


Рисунок 3.3 – Менеджер плагінів в Autodesk 3ds Max

Ці плагіни, їх кількість близько 350, реалізують значну частину функціоналу програми. Наприклад, плагін prim.dlo забезпечує створення стандартних примітивів, сплайнів і стандартних джерел світла.

Інші, нестандартні плагіни переважніше розміщувати в інших папках і завантажувати по мірі необхідності.

SDK (Software Development Kit) – набір засобів розробки, маються в професійній версії Autodesk 3ds Max і містять заголовні і бібліотечні файли, що забезпечують доступ до класів та інших даних Autodesk 3ds Max. Крім того, у пакеті мається помічник і велика кількість прикладів. Плагін Autodesk 3ds Max може бути написаний на мові MAXScript і на мові C++ з вживанням Autodesk 3ds Max SDK.

У першому випадку плагін створюється в середовищі Autodesk 3ds Max, а в другому - в середовищі Microsoft Visual Studio. У багатьох випадках MAXScript може забезпечити такий же функціонал, як і SDK. Однак швидкість SDK-плагінів вище.

SDK-плагін – це відкомпільований і зібраний DLL-файл C++ (DLL, Dynamic Linked Library, що динамічно підключається). Після постановки завдання написання SDK-плагіна, необхідне створення належного інтерфейсу користувача і реалізації процедури. Наприклад, плагін gSphere.dlo забезпечує наведений на рис. 3.4 інтерфейс і програмно підтримує відповідну реакцію програми на передбачені інтерфейсом дії.





Рисунок 3.4 – Інтерфейс плагіну gSphere.dlo

На рис. 3.5 показаний приклад створення простого плагіна: примітив має один параметр Size, для управління яким використовується редаговане поле з лічильником.

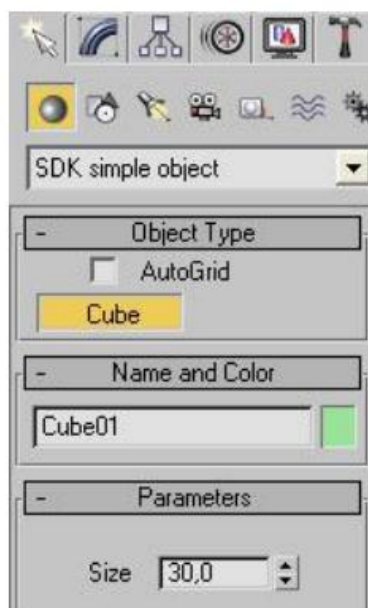


Рисунок 3.5 – Інтерфейс плагіну

Після копіювання і запуску коду в MAXScript Editor, плагін буде доступний на вкладці Create - Geometry - Scripted Primitives (рис. 3.6).



Рисунок 3.6 – Виклик плагіну

Розроблені в роботі програмні продукти призначені для підвищення реалістичності формування відблиску на тривимірних об'єктах і перспективно-коректного зафарбування і можуть бути використані в якості плагінів для професійного пакету Autodesk 3ds Max.

### **3.3 Розробка програмних засобів для відтворення кольорів у комп'ютерній графіці**

У магістерській роботі розроблено програмний модуль для моделювання та тестування методів зафарбовування, який виконує такі функції:

- формування тривимірних графічних об'єктів у динамічному та статичному режимах за вибраним методом;
- визначення за методом порівняння зображень візуальних відмінностей між двома зображеннями, сформованими різними методами;
- вибір різних моделей освітлення для моделювання, значень коефіцієнтів спекулярності, інтенсивностей розсіяного та точкового джерел світла, коефіцієнтів розсіяної, дифузної та спекулярної складових кольору;
- визначення часу формування графічних об'єктів при їх формуванні в статичному режимі;
- визначення кількості кадрів у секунду при формуванні динамічних зображень з виділенням окремо часу, який витрачено на процедуру зафарбовування;

- визначення загальної кількості трикутників у триангуляційній сітці, якою задано графічний об'єкт.

Програму розроблено з використанням мови програмування Java на основі бібліотеки idx3d. На рис. 3.7 наведено об'єктно-орієнтовану модель програмного модуля, а в інструкції програміста список класів програмного модуля.

Графічний конвеєр в програмі включає етап геометричних перетворень, де над об'єктом виконують афінні та видові перетворення, що дозволяє здійснювати переміщення об'єкта в просторі, та етап візуалізації, на якому виконується видалення невидимих поверхонь за методом z-буфера та зафарбовування об'єкта згідно з обраним методом.

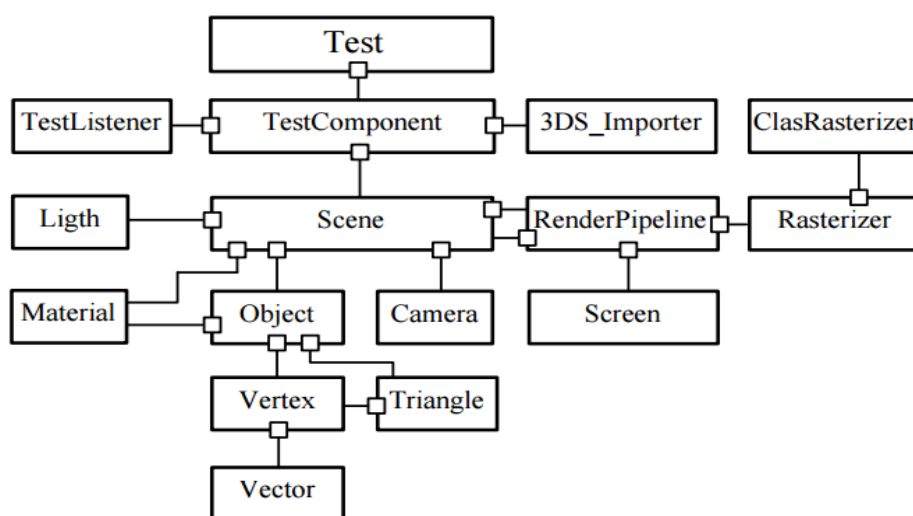


Рисунок 3.7 – Об'єктно-орієнтована модель програмного модуля

Графічний інтерфейс створеного програмного модуля включає чотири функціональні частини, що зображені на рис. 3.8.

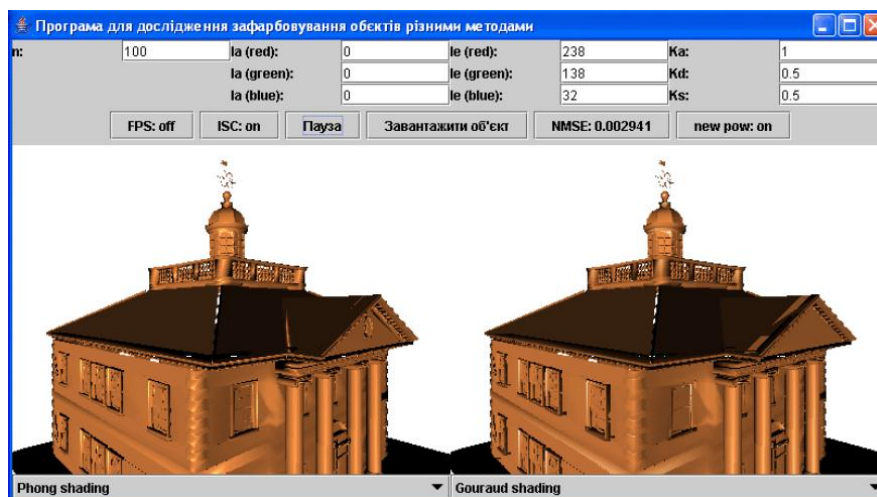


Рисунок 3.8 – Графічний інтерфейс створеного програмного модуля

Перша частина представляє собою набір полів редагування JEdit, що дають можливість користувачу змінювати параметри зафарбовування (рис. 3.9).

100	la (red):	0	le (red):	238	Ka:	1
	la (green):	0	le (green):	138	Kd:	0.5
	la (blue):	0	le (blue):	32	Ks:	0.5

Рисунок 3.9 – Набір полів редагування JEdit

За допомогою полів редагування користувач має можливість змінювати коефіцієнт спекулярності  $p$ ; інтенсивності розсіяного  $I_a$  та точкового  $I_e$  джерел світла; коефіцієнти  $K_a$ ,  $K_d$ ,  $K_s$  розсіяної, дифузної, спекулярної складових кольору.

Друга частина інтерфейсу користувача (рис. 3.10) включає набір кнопок для керування процесом зафарбовування. Вибір типу ДФВЗ для розрахунку спекулярної складової кольору здійснюється за допомогою кнопки “new row”. Значення “off” свідчить про те, що використовується ДФВЗ типу  $\cos^n \gamma$ , а “on” – про те, що використовується інші ДФВЗ. Вибір об’єктів для зафарбовування здійснюється з використанням кнопки “Вибрати об’єкт”. Кнопка “Пауза” призупиняє процес зафарбовування. При активізації кнопки “NMSE” у відповідному полі виводиться значення нормованої середньоквадратичної похибки [41].

У полі “render time” виводиться значення проміжку часу, за який здійснюється зафарбовування фігури, в полі “FPS” – кількість кадрів у секунду при рендерингу об’єкта, а в полі “x/y pol” – співвідношення між загальною кількістю трикутників та трикутниками, в яких спостерігається відблиск або його частина.

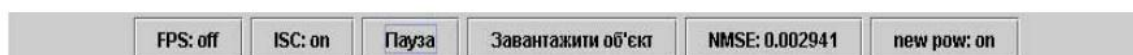


Рисунок 3.10 – Набір кнопок для контролю процесом

Третя частина інтерфейсу користувача призначена для виводу згенерованих тривимірних об’єктів.

Четверта частина інтерфейсу призначена для вибору методу зафарбовування тривимірного об’єкту (рис. 3.11).



Рисунок 3.11 – Частина інтерфейсу для вибору методу зафарбування

Для встановлення потрібного ступеня відбиття поверхнею світла, використовується поле параметру  $n$ . Коефіцієнт спекулярності поверхні, що приймає значення від 0 до 1000.

Для завантаження потрібного об'єкта потрібно натиснути мишкою на кнопку “Завантажити об'єкт” та в вікні “Open”, вибрати потрібний об'єкт, що має розширення \*.3ds.

Програма складається з наступних основних класів:

*Test* – реалізовано інтерфейс користувача та забезпечено вибір методу зафарбовування (в масиві *rasterizers* знаходяться класи растеризації, для додавання новий метод, необхідно спочатку написати клас растеризації та додати його в даний масив таким чином: `new <назва нового методу зафарбовування()>`;

*TestComponent* – забезпечує відображення зафарбованого об'єкта на екрані. Зокрема, в цьому класі реалізовано: завантаження файлів із об'єктами у програму (тобто файли в яких міститься набір векторів та координат точок об'єкта, що повністю описує об'єкт); додавання в сцену джерела світла; забезпечення афінних перетворень об'єкта, що зафарбовується; реалізація зупинки об'єкта, що обертається; встановлення кольорів зафарбовування для об'єкта; визначення NMSE; визначення часу зафарбовування об'єкта.

Попередньо описані класи містять в собі підкласи, що перераховані в табл. 3.1.

Таблиця 3.1 - Короткий опис підкласів, що використовуються в програмі

Назва класу	Опис класу
<code>idx3d_3ds_Exporter</code>	Реалізує збереження об'єкта в файл
<code>idx3d_3ds_Importer</code>	Завантаження об'єкта з файлу (він повинен бути розміщений в тій ж директорії, що й програма та мати розширення *.3ds )
<code>idx3d_3ds_Camera</code>	Реалізує розташування камери в сцені
<code>idx3d_3ds_Color</code>	Реалізуються різні операції з кольором
<code>idx3d_3ds_CoreObject</code>	Забезпечує реалізацію основних методів роботи з об'єктом, та основні властивості того ж об'єкта

### 3.4 Експериментальні дослідження

Основними критеріями, за якими оцінюють засоби формування графічних зображень, є реалістичність та продуктивність. У загальноприйнятому випадку продуктивність визначається обсягом роботи, виконаної за одиницю часу. Такий еталон комплексно оцінює продуктивність всіх етапів графічного конвеєра, оскільки формування зображення є багатоетапним. Зокрема, за таким критерієм можна оцінювати в графічних

системах однотипні засоби, такі як, засоби формування рельєфних поверхонь, або запропонований метод з існуючим методом. У цьому випадку співвідношення продуктивності розробленого методу з базовим дозволить судити про ефективність методу за умови, що досягнуто візуальної ідентичності відтворення сцени.

Для перевірки працездатності розроблених засобів важливо їх інтегрування до складу графічного конвеєра. У цьому випадку буде забезпечено не тільки перевірку працездатності розробленого засобу, але і його взаємоузгодженість з іншими процедурами, які впливають на час генерації сцени.

Час роботи програми може бути оцінено кількістю тактів синхронізації для цієї програми, помноженим на тривалість такту синхронізації. Для визначення загальної кількості тактів необхідно визначити кількість складових операцій та їх тактів з подальшим перемноженням і додаванням отриманих доданків.

У роботі використовувалося еталонне тестування для порівняльного аналізу продуктивності (методом прогону контрольних завдань). При цьому використовувалися еталонні фігури.

Для досягнення високої реалістичності в комп'ютерній графіці необхідно точно відтворити властивості поверхні та фізично правильно описати ефекти освітлення на сцені.

Порівнювати розроблені методи рендерингу доцільно з методом, який забезпечує найбільш точне відтворення графічного зображення для базового напрямку. Візуальну відмінність між двома зображеннями оцінюють за нормованою середньоквадратичною похибкою (NMSE), яка розраховується за формулою:

$$NMSE = \frac{\sum_i (R_1(i) - R_2(i))^2 + (G_1(i) - G_2(i))^2 + (B_1(i) - B_2(i))^2}{\sum_i R_1(i)^2 + G_1(i)^2 + B_1(i)^2}, \quad (3.1)$$

де  $i$  - кількість пікселів, з яких складається об'єкт,  $R_1(i), G_1(i), B_1(i), R_2(i), G_2(i), B_2(i)$  - інтенсивності кольору червоної, зеленої та синьої складових кольору  $i$ -го пікселя зображення відповідно еталонного та сформованого об'єкту.

У комп'ютерній графіці під час тестування зображень використовують такі оцінки:

- $NMSE$  не більше за 0,0001, то візуально зображення не відрізняються один від одного;
- $NMSE$  знаходиться в діапазоні [0,0001 - 0,00025], то два зображення мають незначні відмінності;
- $NMSE$  знаходиться в діапазоні [0,00025 - 0,001], то зображення мають візуально помітні відмінності;
- $NMSE$  більша за 0,001, то два зображення суттєво відрізняються один від одного.

Результати моделювання засвідчують, що при використанні розроблених ДФВЗ має місце візуальна ідентичність сформованих фігур по відношенню до класичної реалізації.

Запропоновано модифікацію моделі Шліка, яку отримано евристично та підтверджено експериментально. Було показано, що при використанні апроксимації максимальна відносна похибка при відтворенні відблиску складає не більше 0,4% порівняно з 10% при використанні формули Шліка. Крім того, значно точніше відтворюється блюмінг відблиску.

При моделюванні використано всі етапи графічного конвеєра. Досліджувались модифікація ДФВЗ Шліка, квадратична, апроксимація ДФВЗ Фонга. Результати моделювання наведено в табл. 3.2.

Таблиця 3.2 - Максимальні значення NMSE для розроблених ДФВЗ

ДФВЗ	Максимальне значення <i>NMSE</i> при формуванні тестових фігур
Модифікація ДФВЗ Шліка: $f(y) = \frac{\cos^a x}{n - n \cos x + \cos x}$	0,00011
Квадратична $\cos^n x = a \times \cos^2 x + b \times \cos x + c$	0,00026 при $n < 170$ . В інших випадках $NMSE < 0,0002$
Косинус-квадратична апроксимація моделі Фонга $\left(\frac{n+1}{3} \times (\cos \gamma - 1) + 1\right)^2$	0,00008

Тестування показало, що модель, розроблена на основі функції  $t = \left(\frac{n+1}{3} \times (\cos \gamma - 1) + 1\right)^2$  має досить високу якість формування відблиску у тривимірних об'єктах. Дана модель є менш ресурсомісткою ніж еталонна модель Фонга та має досить точну зону формування відблиску.

Було розроблено таку апроксимацію функції Шліка:  
 $f(y) = \frac{\cos^a x}{n - n \cos x + \cos x}$  дослідження даної моделі показало, що дана модель більш точно формує зону затухання двопрменевої функції відбивної здатності поверхні.

Дослідження квадратичної апроксимації ДФВЗ, яка має просту апаратну реалізацію, показало, що дана модель дозволяє підвищити точність відтворення епіцентру відблиску більш ніж в 10 разів.

При використанні ДФВЗ  $t = \left(\frac{(n+1)}{3} \times (\cos \gamma - 1) + 1\right)^2$  абсолютна похибка апроксимації епіцентру відблиска в три рази менша порівнюючи з ДФВЗ Шліка, причому для її розрахунку не потрібно виконувати трудомістку операцію ділення. Порівняно із ДФВЗ Шліка запропонована функція при програмній реалізації розраховується в 1,7 рази швидше. Комп'ютерне моделювання показало, що для функції  $t = \left(\frac{(n+1)}{3} \times (\cos \gamma - 1) + 1\right)^2$  не спостерігається візуальних відмінностей між тестовим фігурами, про що свідчить значення NMSE, яке менше за 0,0001. На рис. 3.12 зображені еталонні фігури, що будуть тестуватись нижче.

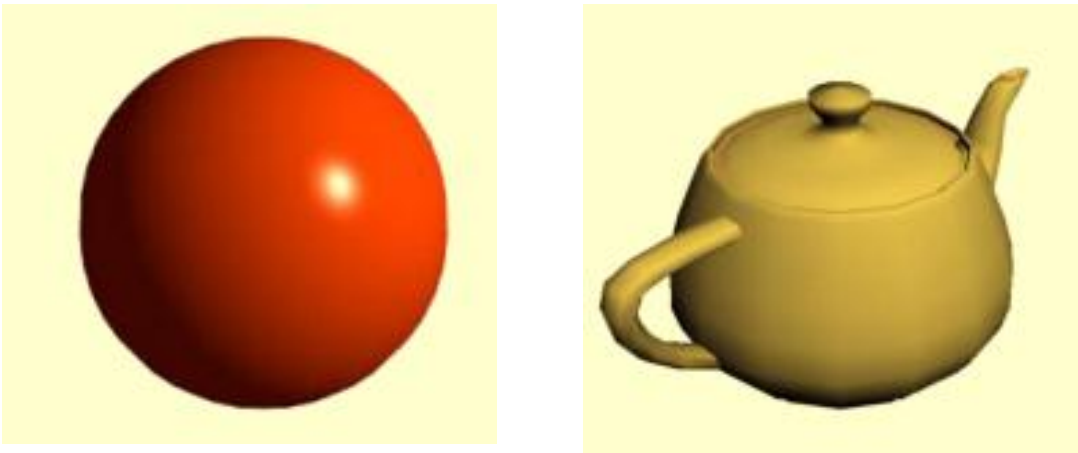
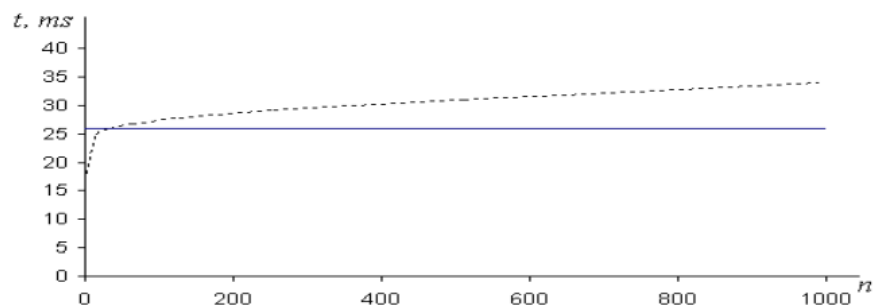


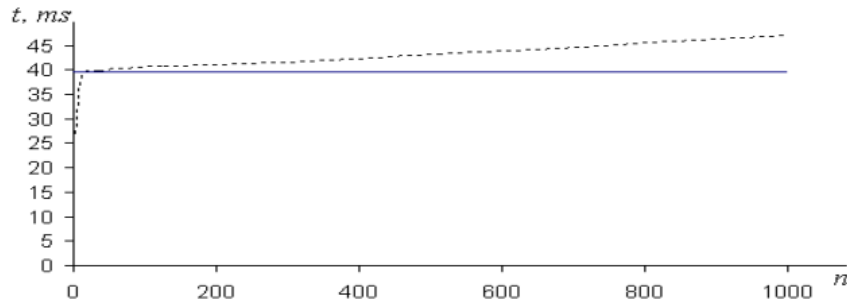
Рисунок 3.12 – Приклади фігури для тестування: «сфера» та «чайник Ньюелла»

Рисунок 3.13 наведено графіки залежності часу зафарбовування тестових фігур при використанні ДФВЗ Бліна та ДФВЗ  $t = \left(\frac{(n+1)}{3} \times (\cos \gamma - 1) + 1\right)^2$ . Із графіків видно, що досягається підвищення продуктивності до 20%.



а) сфера





б) чайник Ньюелла

Рисунок 3.12 – Залежність часу формування фігур від коефіцієнта спекулярності:

----- згідно базового методу;

— з використанням ДФВЗ типу  $t = \left( \frac{(n+1)}{3} \times (\cos \gamma - 1) + 1 \right)^2$ .

### 3.5 Висновки до третього розділу

В третьому розділі було розроблено апаратний засіб: пристрій для визначення спекулярної складової кольору та відбивної здатності поверхні на основі апроксимації ДФВЗ косинус-квадратичною функцією.

Розроблено програмний модуль для моделювання та тестування методів зафарбовування, що виконує такі функції:

- формування тривимірних графічних об'єктів у статичному та динамічному режимах за вибраним методом;
- визначення за методом порівняння зображень візуальних відмінностей між двома зображеннями, сформованими за різними методами;
- обирання для моделювання різних моделей освітлення, значень коефіцієнтів спекулярності, інтенсивностей розсіяного та точкового джерел світла, коефіцієнтів розсіяної, дифузної та спекулярної складових кольору;
- визначення часу формування графічних об'єктів при їх формуванні в статичному режимі;

Для усіх розроблених моделей обчислено нормовану середньоквадратичну похибку  $NMSE$ , для наглядного показу того що, в усіх розроблених методах присутнє покращення продуктивності процесу формування тривимірних графічних сцен.

## РОЗДІЛ 4

### ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

В даному розділі проведено аналіз потенційних небезпечних і шкідливих виробничих факторів, причин пожеж. Розглянуто заходи, які дозволяють забезпечити гігієну праці та виробничу санітарію. На підставі аналізу розроблено заходи з техніки безпеки і рекомендації з пожежної профілактики. І оскільки завданням на дипломне проектування є програмне забезпечення, то аналіз потенційно небезпечних і шкідливих виробничих факторів виконується для персонального комп'ютера, на якому передбачається реалізація та розробка методів та засобів підвищення реалістичності відтворення кольорів у засобах комп'ютерної графіки.

#### 4.1 Аналіз стану умов праці

Для створення реалізації поставленої задачі достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

Оформлення та розробка проекту за фізичним навантаженням відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни. За ступенем нервово-психічної напруги виконання роботи можна віднести до II – III ступеня і кваліфікувати як помірно напружений – напружений за умови успішного виконання поставлених завдань.

Під час виконання робіт використовують ПК та периферійні пристрої (лазерні та струменеві), що призводить до навантаження на окремі системи організму. Такі перекося у напруженні різних систем організму, що трапляються під час роботи з ПК, зокрема, значна напруженість зорового аналізатора і довготривале малорухоме положення перед екраном, не тільки не зменшують загального напруження, а навпаки, призводять до його посилення і появи стресових реакцій.

Найбільшому ризику виникнення різноманітних порушень піддаються: органи зору, м'язово-скелетна система, нервово-психічна діяльність, репродуктивна функція у жінок.

Тобто наявні психофізіологічні небезпечні та шкідливі фактори:

а) фізичного перевантаження:

- статичного;
- динамічного;

б) нервово-психічного перевантаження:

- розумового перенапруження;
- монотонності праці; - перенапруження аналізаторів;
- емоційних перевантажень.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, а також інші профілактичні заходи [40]. Роботу за дипломним проектом визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви: - для розробників програм тривалістю 15 хвилин через кожен годину роботи.

#### **4.2 Аналіз потенційних небезпечних і шкідливих виробничих факторів при роботі з персональним комп'ютером**

Основними характеристиками персонального комп'ютера є наступні:

- 1) робоча напруга  $U = 220 \pm 5\%$ ;
- 2) робочий струм  $I = 2A$ ;
- 3) споживана потужність  $P = 350 \text{ Вт}$ .

Роботу користувача розробленої підсистеми слід віднести до категорії Ia (легкі фізичні роботи). До даної категорії відносяться всі види діяльності, які виконуються сидячи, з періодичним ходінням, і не потребують фізичного напруження [41].

При експлуатації даного програмного продукту існують такі небезпечні і шкідливі виробничі фактори:

1) фізичні:

- підвищений рівень напруги електричної мережі, замикання якої може статися через тіло людини;
- підвищена або знижена вологість повітря;
- підвищена або знижена рухомість повітря;
- підвищений рівень статичної електрики;
- підвищена напруженість електричного поля;
- відсутність або нестача природного світла;
- знижена освітленість робочої зони;

- підвищений рівень шуму на робочому місці;
  - підвищений рівень електромагнітного випромінювання;
  - знижена контрастність;
- 2) психофізіологічні:
- 3) фізичні перевантаження:
- статичні;
  - динамічні;
  - нервово-психічні перевантаження:
  - розумове перенапруження;
  - монотонність праці;
  - перенапруження аналізаторів;
  - емоційні перевантаження.

### **4.3 Заходи з охорони праці**

#### **4.3.1 Загальні заходи безпеки**

Відповідно до санітарно-гігієнічних нормативів та правил експлуатації обладнання приклад деяких заходів безпеки:

1) Заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв передбачають:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення);
- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціювання повітря:
  - а) якщо об'єм приміщення  $20 \text{ м}^3$ , то потрібно подати не менш як  $30 \text{ м}^3/\text{год}$  повітря;
  - б) якщо об'єм приміщення у межах від  $20$  до  $40 \text{ м}^3$ , то потрібно подати не менш як  $20 \text{ м}^3/\text{год}$  повітря;
  - в) якщо об'єм приміщення становить понад  $40 \text{ м}^3$ , допускається природна вентиляція, у випадку, коли немає виділення шкідливих речовин.

- зниження рівня шуму та вібрації:

а) у джерелі виникнення, шляхом застосування раціональних конструкцій, нових матеріалів і технологічних процесів;

б) звукоізолювання устаткування за допомогою глушників, резонаторів, кожухів, захисних конструкцій, оздоблення стін, стелі, підлоги тощо;

в) використання засобів індивідуального захисту).

2) Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:

- постійно стежити за справним станом електромережі, розподільних щитків, вимикачів, штепсельних розеток, лампових патронів, а також мережевих кабелів живлення, за допомогою яких електроприлади під'єднують до електромережі;

- постійно стежити за справністю ізоляції електромережі та мережевих кабелів, не допускаючи їхньої експлуатації з пошкодженою ізоляцією;

- не тягнути за мережевий кабель, щоб витягти вилку з розетки;

- не закривати меблями, різноманітним інвентарем вимикачі, штепсельні розетки;

- не підключати одночасно декілька потужних електропристроїв до однієї розетки, що може викликати надмірне нагрівання провідників, руйнування їхньої ізоляції, розплавлення і загоряння полімерних матеріалів;

- не залишати включені електроприлади без нагляду;

- не допускати потрапляння всередину електроприладів крізь вентиляційні отвори рідин або металевих предметів, а також не закривати їх та підтримувати в належній чистоті, щоб уникнути перегрівання та займання приладу;

- не ставити на електроприлади матеріали, які можуть під дією теплоти, що виділяється, загорітися (канцелярські товари, сувенірну продукцію тощо).

#### **4.3.2 Електробезпека**

Основним небезпечним фактором при роботі з ЕОМ є небезпека ураження людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані виявити наявності електричної напруги на обладнанні.

Проходячи через тіло людини, електричний струм чинить на нього складний вплив, що є сукупністю термічної (нагрів тканин і біологічних середовищ), електролітичної (розкладання крові і плазми) і біологічної (роздратування і збудження нервових волокон та інших органів тканин організму) дій.

Тяжкість ураження людини електричним струмом залежить від цілого ряду чинників:

- 1) значення сили струму;
- 2) електричного опору тіла людини і тривалості протікання через нього струму;
- 3) типу і частоти струму;
- 4) індивідуальних властивостей людини і навколишнього середовища.

Приміщення для ЕОМ відноситься до приміщень без підвищеної небезпеки, тобто в приміщення, в яких відсутні умови, що створюють підвищену або особливу небезпеку. Небезпека ураження електричним струмом існує всюди, де використовуються електроустановки, тому приміщення без підвищеної небезпеки не можна назвати безпечними.

Електробезпека забезпечується:

- 1) відповідною конструкцією електроустановок;
- 2) застосуванням технічних способів і засобів захисту;
- 3) організаційними і технічними заходами.

Конструкція електроустановок відповідає умовам їх експлуатації та забезпечує захист персоналу від дотику до струмоведучих частин.

Основними технічними способами і засобами захисту від ураження електричним струмом, що використовуються окремо або в поєднанні один з одним, є:

- 1) захисне заземлення;
- 2) занулення;
- 3) вирівнювання потенціалів;
- 4) мале напруга;
- 5) електричне поділ мереж;
- 6) захисне відключення;
- 7) ізоляція струмоведучих частин;
- 8) компенсація струмів замикання на землю;
- 9) захисні пристрої;
- 10) попереджувальна сигналізація, блокування, знаки безпеки;
- 11) ізолюючі захисні та запобіжні пристосування.

#### **4.3.3 Розрахунок захисного заземлення**

Основними технічними способами і засобами захисту від ураження електричним струмом, що передбачаються в даному дипломному проекті, є:

- 1) захисне заземлення,
- 2) занулення,
- 3) захисне відключення,
- 4) ізоляція струмоведучих частин.

Завдання захисного заземлення - усунення небезпеки ураження струмом у випадку дотику до корпусу та інших струмоведучих металевих частин електроустановок, які опинилися під напругою.

Розрахунок заземлюючого контуру виконується виходячи з умови:

$$R_{ЗК} = \frac{R_3 \times R_{\Pi}}{R_{\Pi} \times n \times \eta_3 + R_3 \times \eta_{\Pi}} \leq 4 \text{ Ом} \quad (4.1)$$

де  $R_3$  - опір заземлювача (стержня, труби, куточка і т.д.), Ом;

$R_{\Pi}$  - Опір лінії, що з'єднує заземлювачі, Ом;

$n$  - кількість заземлювачів;

$\eta_3$  і  $\eta_{\Pi}$  - Коефіцієнти екранування відповідно заземлювача і з'єднує смуги ( $\eta_3 = 0,2 \div 0,9$ ;  $\eta_{\Pi} = 0,1 \div 0,7$ ).

Опір заземлювача розраховується за формулою 4.2

$$R_3 = \frac{\rho}{2 \times \pi \times l} \times \left( \ln \frac{2 \times l}{d} + \frac{1}{2} \times \ln \frac{4 \times t + l}{4 \times t - l} \right) \quad (4.2)$$

де  $\rho$  - питомий опір ґрунту (взяти з довідкової літератури);

$l$  - довжина заземлювача (для труб 2-3 м, для стрижнів до 10 м), м;

$d$  - діаметр заземлювача (для стрижнів 0,01 - 0,03 м, для труб 0,03 - 0,05 м);

$t$  - відстань від середини забитого в ґрунт заземлювача до рівня землі (необхідно враховувати, що відстань від верхнього кінця заземлювача до поверхні землі має бути не менше 0,5), м.

Розрахуємо опір заземлювача:

$$R_3 = \frac{60}{2 \times \pi \times 3} \times \left( \ln \frac{2 \times 3}{0,03} + \frac{1}{2} \times \ln \frac{4 \times 1 + 3}{4 \times 1 - 3} \right) = 19,96$$

Опір лінії, що з'єднує заземлювачі розраховується за формулою 4.3

$$R_3 = \frac{\rho}{2 \times \pi \times l} \times \left( \ln \frac{2 \times L^2}{b \times t} \right) \quad (4.3)$$

де  $L$  - довжина лінії, що з'єднує заземлювачі (при контурному заземленні вона приблизно дорівнює периметру виробничої будівлі), м;

$b$  - ширина смуги (0,03 - при прокладанні всередині будівлі і 0,05 - при прокладанні поза будівлею), м;

$t$  - глибина заземлення від рівня землі (0,5 м).

Розрахуємо опір лінії, що з'єднує заземлювачі

$$R_3 = \frac{60}{2 \times \pi \times 3} \times \ln \frac{2 \times 50^2}{0.03 \times 0.5} = 14.37$$

Необхідна кількість заземлювачів, розраховується за формулою 4.4

$$n = \frac{2 \times R_3}{4 \times \eta_3} \quad (4.4)$$

де 4 - допустимий загальний опір;

2 - коефіцієнт сезонності.

Розрахуємо необхідну кількість заземлювачів,

$$n = \frac{2 \times 19.9}{4 \times 0.5} = 19.9 \approx 20$$

Округлимо результат в більшу сторону і отримуємо необхідну кількість заземлювачів - 20. Маючи всі необхідні дані розрахуємо опір заземлюючого контуру.

$$R_{3к} = \frac{19.96 \times 14.37}{14.37 \times 20 \times 0.5 + 19.96 \times 0.4} = 1.89 \leq 4 \text{ Ом}$$

Опір заземлюючого контуру 1,89 Ом, що відповідає умові  $R_{3к} < 4 \text{ Ом}$ .

#### **4.4 Заходи, що забезпечують виробничу санітарію та гігієну праці**

##### **4.4.1 Мікроклімат**

Трудова діяльність людини завжди протікає в певних метеорологічних умовах, які визначаються поєднанням температури повітря, швидкості його руху і відносної вологості, тиском і тепловим випромінюванням від нагрітих поверхонь. Оскільки експлуатація проектного програмного засобу відбувається в приміщенні, то ці показники в сукупності (за винятком тиску) називаються мікрокліматом виробничого приміщення. На даний час основним нормативним документом, щодо нормалізації мікроклімату є ГОСТ 12.1.004-91 ССБТ [42].

Важкість праці характеризує сукупну дію всіх елементів, складових умови праці, на працездатність людини, його здоров'я, життєдіяльність і відновлення робочої сили. У такому представленні поняття тяжкості праці однаково застосовні як до розумової, так і до фізичної праці. Згідно [42] тяжкість роботи персоналу, який обслуговує ЕОМ, відноситься до легкої категорії 1б (роботи, виконувані сидячи, не вимагаючи систематичного фізичного напруження і перенесення важких предметів). Загальні санітарно-гігієнічні



вимоги до повітря робочої зони. Оптимальні норми мікроклімату в робочій зоні, що забезпечуються для робіт легкої категорії 1б приведені в таблиці 4.3.

Таблиця 4.3 - Оптимальні норми мікроклімату

Період року	Температура, °C	Відносна вологість,%	Швидкість вітру, м / с, не більше
Холодний	21 - 23	60 - 40	0,1
Теплий	22 - 24	60 - 40	0,2

#### 4.4.2 Освітлення

Світло є природною умовою існування людини . Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Гарне освітлення діє тонізуюче, створює гарний настрій, поліпшує протікання основних процесів вищої нервової діяльності.

Збільшення освітленості сприяє поліпшенню працездатності навіть в тих випадках, коли процес праці практично не залежить від зорового сприйняття. При поганому освітленні людина швидко втомлюється, працює менш продуктивно, виникає потенційна небезпека помилкових дій і нещасних випадків.

Штучне освітлення в робочому приміщенні передбачається здійснювати з використанням люмінесцентних джерел світла у світильниках загального освітлення, оскільки люмінесцентні лампи мають високу потужність (80 Вт), тривалий термін служби (до 10000 годин), спектральний складом випромінюваного світла, близький до сонячного. При експлуатації ЕОМ виконується зорова робота IV в розряд точності (середня точність). При цьому нормована освітленість на робочому місці (Ен) дорівнює 200 лк. Джерелом природного освітлення є сонячне світло. У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає [43].

Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовільний і для даного приміщення в світлий час доби достатньо природного освітлення. Світильники загального освітлення розташовуються над робочими поверхнями у рівномірно-прямокутному порядку.

Розрахунок освітлення.

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше -1/8, в побутових – 1/10:

$$S_b = \left(\frac{1}{5} \div \frac{1}{10}\right) \cdot S_n \quad (4.5)$$

де  $S_b$  – площа віконних прорізів,  $m^2$ ;

$S_n$  – площа підлоги,  $m^2$ .

$$S_n = a \cdot b = 4 \cdot 4 = 16 \text{ м}^2,$$

$$S = 1/8 \cdot 16 = 2 \text{ м}^2.$$

Приймаємо 1 вікно площею  $S = 2 \text{ м}^2$ . Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 4 м, ширина 4 м, світильниками ЛПО2П, оснащеними лампами типа ЛБ (дві по 80 Вт) з світловим потоком 5400 лм кожна. Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні.

Розрахунок кількості світильників  $n$  визначається по формулі (4.6):

$$n = \frac{E \times S \times Z \times K}{F \times U \times M} \quad (4.6)$$

де  $E$  – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

$S$  – освітлювана площа,  $m^2$ ;  $S = 16 \text{ м}^2$ ;

$Z$  – поправочний коефіцієнт світильника ( $Z = 1,15$  для ламп розжарювання та ДРЛ;  $Z = 1,1$  для люмінесцентних ламп) приймаємо рівним 1,1;

$K$  – коефіцієнт запасу, враховує зниження освітленості в процесі експлуатації – 1,5;

$U$  – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575;

$M$  – число люмінесцентних ламп в світильнику – 2;

$F$  – світловий потік лампи – 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.6), отримаємо:

$$n = \frac{300 \cdot 16 \cdot 1,1 \cdot 1,5}{5400 \cdot 0,575 \cdot 2} = 1,3$$

Приймаємо освітлювальну установку, яка складається з одного світильника потужністю 130 Вт, напругою – 220 В.

#### 4.5 Рекомендації щодо пожежної безпеки

Виникнення пожежі можливо, якщо на об'єкті є горючі речовини, окислювач і джерела запалювання. Для оцінки пожежної небезпеки слід проаналізувати ймовірність взаємодії цих трьох чинників.

Горючими матеріалами в приміщенні, де розташовані ЕОМ, є:

- 1) поліамід - матеріал корпусу мікросхем, горюча речовина, температура самозаймання 420 ° С;
- 2) полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 ° С, температура самозаймання 530 ° С;
- 3) склотекстоліт ДЦ - матеріал друкованих плат, трудногорючий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;
- 4) пластикат кабельний №.489 - матеріал ізоляції кабелів, горючий матеріал, показник горючості більше 2.1;
- 5) деревина - будівельний і оздоблювальний матеріал, з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, температура запалювання 255 ° С, температура самозаймання 399 ° С.

Згідно [44] таке приміщення належить до категорії "В" (пожежонебезпечної).

Простору всередині приміщень в межах яких можуть утворюватися або знаходитися пожежонебезпечні речовини і матеріали у відповідності з ПУЕ відносяться до пожежонебезпечної зони класу II-Па.

Потенційними джерелами запалювання можуть бути:

- 1) іскри і дуги короткого замикання;
- 2) електрична іскра при замиканні і розмиканні ланцюгів;
- 3) перегіви від тривалого перевантаження;
- 4) відкритий вогонь і продукти горіння;
- 5) наявність речовин, нагрітих вище температури самозаймання;
- 6) розрядна статичну електрику.

Причинами можливого загоряння і пожежі можуть бути:

- 1) несправність електроустановки;
- 2) конструктивні недоліки обладнання;
- 3) коротке замикання в електричних мережах;
- 4) запалювання горючих матеріалів, що знаходяться в безпосередній близькості від електроустановки.

Продуктами згоряння, що виділяються під час пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор та ін.

При горінні пластмас, крім звичних продуктів згоряння, виділяються різні продукти термічного розкладання: хлорангідридні кислоти; формальдегіди; хлористий водень; фосген; синильна кислота; аміак; фенол; ацетон; стирол [45].

Для захисту персоналу від впливу небезпечних і шкідливих факторів пожежі проектом передбачається застосування промислового протигазу фільтруючого з коробкою марки В (жовтий).

Небезпека розвитку пожежі на обчислювальному центрі обумовлюється застосуванням розгалужених систем вентиляції та кондиціонування, розвиненою системою електроживлення ЕОМ. Небезпека загорання в ЕОМ пов'язана з великою кількістю щільно розташованих на платі і блоках електронних вузлів і схем, електричних і комутаційних кабелів, резисторів, конденсаторів, напівпровідникових діодів і транзисторів. Висока щільність елементів в електронних схемах призводить до значного підвищення температури окремих вузлів (80 ... 100 ° С), що може служити причиною запалювання ізоляційних матеріалів. Слабкий опір ізоляційних матеріалів дії температури може викликати порушення ізоляції і привести до короткого замикання.

Пожежна безпека при застосуванні ЕОМ забезпечується:

- 1) системою запобігання пожежі:
- 2) системою протипожежного захисту:
- 3) організаційно-технічними заходами.

Запобігти утворенню горючого середовища (замінити горючі речовини і матеріали на негорючі та важкогорючі) не надається технічно можливим. Тому проектом передбачаються способи і засоби запобігання утворенню (або внесення) в горюче середовище джерел запалювання, таких як:

- 1) застосування електроустаткування, відповідної пожежонебезпечної і вибухонебезпечної зонами відповідно до ПУЕ;
- 2) застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалювання;
- 3) виключення можливості появи іскрового розряду в займистою середовищі з енергією, яка дорівнює і вище мінімальної енергії запалювання.

#### **4.6 Екологія**

Діяльність на тему магістерської роботи, а саме: розробка методів та засобів підвищення реалістичності відтворення кольорів у засобах комп'ютерної графіки в процесі її виконання, впливає на навколишнє природне середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища» [46], Законом України «Про забезпечення санітарного та епідемічного благополуччя населення» [47], Законом України «Про відходи» [48].

В процесі розробки виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- Відпрацьовані люмінесцентні лампи - I клас небезпеки.
- Змінні носії інформації - IV клас небезпеки.
- Відпрацьовані вогнегасники - IV клас небезпеки.
- Макулатура - IV клас небезпеки.

#### **4.7 Висновки до четвертого розділу**

У четвертому розділі виконаний аналіз потенційних небезпек при роботі із засобами обчислювальної техніки, на підставі якого розроблено заходи з техніки безпеки, заходи, що забезпечують виробничу санітарію та гігієну праці, розрахунки природного та штучного освітлень, рекомендації з пожежної профілактики, які підтверджені відповідними розрахунками. Також розглянуто вплив роботи на навколишнє середовище та процеси поводження з відходами ІТ галузі.

## ВИСНОВКИ

В атестаційній дипломній роботі було проведено аналіз методів та засобів зафарбовування для систем комп'ютерної графіки. Проведено аналіз найбільш поширенихДФВЗ і методів зафарбовування та визначено основні задачі дослідження.

Розроблено нові моделі двопрореєвої функції відбивної здатності поверхні, що, на відміну від вже існуючих, мають меншу обчислювальну складність, та більшу точність відтворення епіцентру відблиску, що дає можливість підвищити продуктивність зафарбовування.

У другому розділі запропоновано апроксимація моделі Шліка, що має максимально відносну похибку при відтворенні відблиску не більше 3,2% порівняно з 10% при використанні оригінальної формули Шліка. Крім того, значно точніше відтворюється зона відблиску. Має основний недолік – підвищення складності розрахунків. Запропонована квадратична апроксимація, яка дозволяє відтворювати зону епіцентру відблиску з відносною похибкою не більше 2%. Модель з використанням косинус-квадратичної функції забезпечує суттєве підвищення продуктивності в порівнянні з використанням функції Бліна. Запропонований метод дозволяє достатньо точно (з відносною похибкою не більше 3 %) апроксимувати не лише епіцентр відблиску, а також зону затухання. Розроблені методи коректного відтворення кольорів при зафарбовуванні тривимірних графічних об'єктів, основаних на лінійному інтерполюванні інтенсивностей складових кольору і враховують перспективне проектування. Для систем високореалістичної графіки запропоновано модифікацію ДФВЗ, яка є енергетично-коректною, тобто забезпечує виконання закону збереження енергії при відбитті світла від поверхні об'єкта. Запропоновано спосіб синтезу зображень на основі алгоритму зворотного трасування променів, що використовує піксельні сегменти або піксельні блоки і додає міжпіксельну інтерполяцію в кінець трасування поточного променя або групи променів. Отримані результати можуть бути успішно використані у високопродуктивних системах візуалізації, а також у системах, що працюють із структурами зображень у режимі реального часу.

У третьому розділі реалізовано отримані в ході виконання магістерської роботи моделі освітлення також інтегровано їх у складі графічного конвеєра. Засобами графічного конвеєра було розроблено вершинні і піксельні шейдерні програми мовою HLSL, які досліджують та демонструють роботу розроблених моделей відбивної здатності поверхні. Це дало можливість перевірити працездатність розроблених методів у складі професійного графічного конвеєра. Експериментальні дослідження розроблених моделей

відбивної здатності поверхні показали, що розроблені моделі досить точно відтворюють епіцентр та зону затухання відблиску на поверхнях об'єктів, а також є більш продуктивними за рахунок спрощення обчислювального процесу. Для усіх розроблених моделей обчислено нормовану середньоквадратичну помилку *NMSE*, яка показала те, що при використанні розроблених ДФВЗ має місце лише незначне відхилення в якості від існуючих аналогів.

В четвертому розділі виконано аналіз потенційних небезпек при роботі з засобами обчислювальної техніки, на підставі якого розроблено заходи з техніки безпеки, заходи, що забезпечують виробничу санітарію та гігієну праці, розрахунки природного та штучного освітлень, рекомендації з пожежної профілактики, які підтверджені відповідними розрахунками. Також розглянуто вплив роботи на навколишнє середовище та процеси поводження з відходами в ІТ галузі.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Чорний А. В. Високопродуктивні методи та засоби зафарбовування тривимірних графічних об'єктів. Монографія. / А. В. Чорний // Вінниця: УНІВЕСУМ-Вінниця, 2016. — 190 с.
2. Романюк О. Н. Комп'ютерна графіка / О. Н. Романюк // Вінниця: ВДТУ, 2019. — 129 с.
3. Попов, А. А. DirectX 10 - это просто. Програмуємо графіку на C++ [Електронний ресурс] / А. А. Попов // СПб. : БХВ-Петербург, 2008. — 465 с.
4. Херн Д., Паулин Бейкер М. Комп'ютерна графіка и стандарт OpenGL / Д. Херн, М. Паулин Бейкер // М.: Издат. дом “Вильямс”, 2015. - 1168 с.
5. Романюк С. О., Романюк О. Н. Аналіз методів текстурної фільтрації / С. О. Романюк, О. Н. Романюк // Актуальні проблеми сучасних наук. – 2017.
6. Шикин Е. В. Комп'ютерна графіка. Динамика, реалистические изображения / Е. В. Шикин. — М. : Диалог — МИФИ, 1997. — 288 с.
7. Поляков А. Ю., Брусенцев В. А. / Методы и алгоритмы компьютерной графики, 2-е изд. / А. Ю. Поляков, В. А. Брусенцев // СПб.: БХВ - Петербург. – 2013.
8. Barrera T., Hast A., Bengtsson E. An Alternative Model for Real-Time Rendering of Diffuse Light for Rough Materials / T. Barrera, A. Hast, E. Bengtsson // SCCG'05 Proceedings II, 2005. — P. 27—28.
9. Романюк О.Н. Класифікація двопроменевих функцій відбивної здатності поверхні / О.Н. Романюк, 2018 — 129 с.
10. Вяткін С. І., Романюк О.Н. Анімація тривимірних об'єктів / С. І. Вяткін, О. Н. Романюк // Міжнародний науково-технічний журнал «Вимірювальна та обчислювальна техніка в технологічних процесах». Хмельницький. – 2017.
11. Романюк О. Н., Романюк С.О. Аналіз методів анізотропної фільтрації текстур./ О. Н. Романюк, С. О. Романюк. // Міжнародний науково-технічний журнал «Вимірювальна та обчислювальна техніка». Хмельницький. – 2018.
12. Lighting Models. [Електронний ресурс]. – Режим доступу: <http://glasnost.itcarlow.ie/~powerk/GeneralGraphicsNotes/LightingShadingandColour/lighting.html>
13. Cook R. L., Torrance K. E. A reflectance model for computer graphics / R. L. Cook, K. E. Torrance // ACM Trans. on Grsphics. – Vol. 1. – 1992.
14. Simon's Tech Blog: Microfacet BRDF. [Електронний ресурс]. – Режим доступу: <http://simonstechblog.blogspot.com/2011/12/microfacet-brdf.html>



15. Schlick Christophe. A survey of shading and reflectance models / Christophe Schlick. – 2003.
16. Романюк О. Н. Новый подход до підвищення реалістичності зафарбовування тривимірних об'єктів за методом Гуро / О. Н. Романюк // Інформаційні технології та комп'ютерна інженерія. — 2015. — № 2. — С. 106—109.
17. Bishop G. F., Weimer D. M. Fast Phong Shading / G. F. Bishop, D. M. Weimer // Computer Graphics, v20, n4. – 1996
18. Лисиця В. Т. Колірні моделі та закони поширення світла / В. Т. Лисиця // 2012.– С. 19.
19. CMYK color model [Електронний ресурс] – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/CMYK\\_color\\_model](https://en.wikipedia.org/wiki/CMYK_color_model).
20. Комп'ютерна графіка "Навчальний посібник". // Тернопільський національний технічний університет імені Івана Пулюя. – 2017. – С. 44
21. Schlick Christophe. A Customizable Reflectance Model for Everyday Rendering / Christophe Schlick // Fourth Eurographics. – 1993
22. Романюк С. О. Модифікація ДФВЗ Шліка / С. О. Романюк // Електронні інформаційні ресурси: створення, використання, доступ. Збірник матеріалів Міжнародної науково-практичної Інтернет-конференції, – ВНТУ. – 2018 р
23. Цисарж В. В. Математические методы компьютерной графики. Киев: Факт, 2004. 464 с.
24. Корн Г. Справочник по математике (для научных работников и инженеров) / Г. Корн, Т. Корн. — М.: Наука, 1974. — 830 с
25. Farin G. Mathematical Principles for Scientific Computing and Visualization / G. Farin, D. Hansford . — Peters, Ltd. 2008. — 275 p.
26. Рейнбоу В. Компьютерная графика. Энциклопедия / В. Рейнбоу. — С-Пб: Питер, 2003. — 196 с.
27. Эдвард Эйнджел. Интерактивная компьютерная графика. Вводный курс на базе OpenGL, 2 изд.: Пер с англ. – М.: Издательский дом “Вильямс”, 2001. – 592с.
28. Ward G. The radiance lighting simulation and rendering system. Computer Graphics Proceedings by Gurro, July 1994. — pages 459– 472.
29. Порев В.Н. Компьютерная графика / В.Н. Порев – СПб.: БХВ – Петербург, 2002. – 432с.
30. Heckbert P. Fundamentals of Texture Mapping and Image Warping / P. Heckbert // M.Sc. Thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, June, 2017. — 86 p.

31. Гилой В. К. Новые стандарты высокореалистического рендеринга в реальном времени / В. К. Гилой, Г. К. Расселлер // Открытые системы. – 1995. – № 65(13). – С. 35–44
32. Гурский Д., Турбина Е. Вычисления в Mathcad 12 / Д. Гурский, Турбина Е // СПб.: Питер, 2006. — 544 с.
33. Романюк О. Н. Метод підвищення реалістичності відтворення тривимірних графічних об'єктів / О. Н. Романюк // Інформаційні технології та комп'ютерна інженерія. — 2007. — № 1 (8). — С. 192—196.
34. Павлов С. В. Комп'ютерна графіка. Навчальний посібник / С. В. Павлов — Харків: ХНУ, 2001. — 129 с.
35. Schlick C. An inexpensive BRDF model for physically-based rendering // ComputerGraphics Forum. - № 13(3), 1994. —233-246 pp.
36. Савитч Уолтер Язык Java. Курс программирования / Уолтер Савитч. - М.: Вильямс, 2015. - 928 с.
37. Хилл Ф. OpenGL. Программирование компьютерной графики / Ф. Хилл. — СПб. : Питер, 2002. — 1088 с.
38. Давыдов С. Ю., Ефимов А. А., IntelliJ IDEA. Профессиональное программирование на Java / С. Ю. Давыдов, А. А. Ефимов. - М.: БХВ-Петербург, 2015. - 800 с
39. Xiao D. He, Kenneth E. Torrance, Francois X. Sillion, Donald P. Greenberg. A Comprehensive Physical Model for Light Reflection. / D. He Xiao, E. Torrance Kenneth, X. Sillion Francois, P. Greenberg Donald // Technical report, Cornell University, 1991.
40. Малащенко И. В., Моисейкин А. Моделирование свойств поверхности объекта. // Математическая морфология, Том 5 Выпуск 4, 2016
41. Blinn J. Models of Light Reflection For Computer Synthesized Pictures. In Proceedings SIGGRAPH / Blinn J., July 1997. — 126 p.
42. ДСанПіН 3.3.2-007-98 Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. <https://zakon.rada.gov.ua/rada/show/v0007282-98> - 10.12.1998
43. НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. <https://zakon.rada.gov.ua/laws/show/z0508-18> - 14.02.2018
44. ГОСТ 12.1.004-91 ССБТ. Пожарная безопасность. Общие требования <http://docs.cntd.ru/document/gost-12-1-004-91-ssbt>
45. ДБН В.2.5-28:2018 Природне і штучне освітлення. <http://dreamdim.ua/ru/novyi-dbn-v-2-5-28-2018-pryrodne-i-shtuchne-osvitlennya/> - 01.04.2019p

46. НПАОП 40.1-1.32-01 (ДНАОП 0.00-1.32-01) Правила будови електроустановок. Електрообладнання спеціальних установок <https://zakon.rada.gov.ua/rada/show/v0272203-01> - 21.06.2001

47. ГОСТ 12.1.044-89 ССБТ. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения. <http://docs.cntd.ru/document/gost-12-1-044-89>

48. Закон України «Про охорону навколишнього природного середовища» - <https://zakon.rada.gov.ua/laws/main/1264-12> - 12.10.2018р.

49. Закон України «Про забезпечення санітарного та епідемічного благополуччя населення» - <https://zakon.rada.gov.ua/laws/main/4004-12> - 04.10.2018р.

50. Закон України «Про відходи» - <https://zakon.rada.gov.ua/laws/main/187/98-%D0%B2%D1%80> - 01.05.2019р.

**Додаток А**  
**Лістинг програмної реалізації**

```
1 package main;
2 import idx3d.*;
3 import idx3d.GouraudRasterizer;
4 import idx3d.PhongRasterizer;
5 import java.awt.*;
6 import java.awt.event.*;
7 import javax.swing.*;
8 import javax.swing.event.*;
9 public class Test extends JFrame
10 {
11 private JComboBox comboBoxLeft;
12 private JComboBox comboBoxRight;
13 private JComboBox comboBoxLeftLM;
14 private JComboBox comboBoxRightLM;
15 private TestComponent testLeft;
16 private TestComponent testRight;
17 private JButton disableSButton;
18 private JButton modelButton;
19 private JButton disableMSEButton;
20 private JButton cosButton;
21 private JButton benchmarkButton;
22 private JTextField nField;
23 private JTextField larField;
24 private JTextField lagField;
25 private JTextField labField;
26 private JTextField lerField;
27 private JTextField legField;
28 private JTextField lebField;
29 private JTextField kaField;
30 private JTextField kdField;
31 private JTextField ksField;
32 private MyDocumentListener mdl;
```

```

33     public static idx3d_Rasterizer[] rasterizers = new idx3d_Rasterizer[] {
34         new PhongRasterizer(),
35         new PhongDeltaGourandRasterizer(),
36         new GouraudRasterizer(),
37         new GouraudRasterizer4(),
38         new GouraudRasterizerPerspectiva(),
39         new Approx_cos_line(),
40         new Approximation_cos_n(),
41         new Triangl(),
42         new Phong_New_Shlik(),
43         new NormalVector(),
44         new log2_n()
45     };
46     public static void main(String[] args)
47     {
48         new Test();
49     }
50     public Test()
51     {
52         mdl = new MyDocumentListener();
53         this.resize(320, 240);
54         this.setTitle("Програма для дослідження зафарбовування об'єктів різними
методами");
55         this.move(200, 200);
56         comboBoxLeft = new JComboBox();
57         comboBoxRight = new JComboBox();
58         for (int i = 0; i < rasterizers.length; i++)
59         {
60             comboBoxLeft.addItem(rasterizers[i].getRasterizerName());
61             comboBoxRight.addItem(rasterizers[i].getRasterizerName());
62         }
63         comboBoxLeft.addActionListener(new ActionListener()
64         {
65             public void actionPerformed(ActionEvent e)
66             {

```

```
67     int index = comboBoxLeft.getSelectedIndex();
68     testLeft.setRasterizer(rasterizers[index]);
69     }
70     });
71     comboBoxRight.addActionListener(new ActionListener()
72     {
73     public void actionPerformed(ActionEvent e)
74     {
75     int index = comboBoxRight.getSelectedIndex();
76     testRight.setRasterizer(rasterizers[index]);
77     }
78     });
79     comboBoxLeftLM = new JComboBox();
80     comboBoxRightLM = new JComboBox();
81     testLeft = new TestComponent();
82     testRight = new TestComponent();
83     testLeft.setRasterizer(rasterizers[0]);
84     testRight.setRasterizer(rasterizers[11]);
85     Box box = Box.createHorizontalBox();
86     Box boxLeft = Box.createVerticalBox();
87     Box boxRight = Box.createVerticalBox();
88     Panel panel = new Panel(new GridLayout(3, 8));
89     boxLeft.add(testLeft);
90     boxLeft.add(comboBoxLeft);
91     boxRight.add(testRight);
92     boxRight.add(comboBoxRight);
93     box.add(boxLeft);
94     box.add(boxRight);
95     panel.add(new JLabel("n: "));
96     nField = new JTextField("100");
97     nField.getDocument().addDocumentListener(mdl);
98     panel.add(nField);
99     panel.add(new JLabel("la (red): "));
100    larField = new JTextField("0");
101    larField.getDocument().addDocumentListener(mdl);
```

```
102 panel.add(larField);
103 panel.add(new JLabel("le (red): "));
104 lerField = new JTextField("238");
105 lerField.getDocument().addDocumentListener(mdl);
106 panel.add(lerField);
107 panel.add(new JLabel("Ka: "));
108 kaField = new JTextField("1");
109 kaField.getDocument().addDocumentListener(mdl);
110 panel.add(kaField);
111 panel.add(new JLabel(""));
112 panel.add(new JLabel(""));
113 panel.add(new JLabel("la (green): "));
114 lagField = new JTextField("0");
115 lagField.getDocument().addDocumentListener(mdl);
116 panel.add(lagField);
117 panel.add(new JLabel("le (green): "));
118 legField = new JTextField("138");
119 legField.getDocument().addDocumentListener(mdl);
120 panel.add(legField);
121 panel.add(new JLabel("Kd: "));
122 kdField = new JTextField("1");
123 kdField.getDocument().addDocumentListener(mdl);
124 panel.add(kdField);
125 panel.add(new JLabel(""));
126 panel.add(new JLabel(""));
127 panel.add(new JLabel("la (blue): "));
128 labField = new JTextField("0");
129 labField.getDocument().addDocumentListener(mdl);
130 panel.add(labField);
131 panel.add(new JLabel("le (blue): "));
132 lebField = new JTextField("32");
133 lebField.getDocument().addDocumentListener(mdl);
134 panel.add(lebField);
135 panel.add(new JLabel("Ks: "));
136 ksField = new JTextField("1");
```

```

137 ksField.getDocument().addDocumentListener(mdl);
138 panel.add(ksField);
139 Panel panelButtons = new Panel();
140 JButton pauseButton = new JButton("Пауза");
141 pauseButton.addActionListener(new ActionListener()
142 {
143     public void actionPerformed(ActionEvent e)
144     {
145         testLeft.pause();
146     }
147 });
148 modelButton = new JButton("Завантажити об'єкт");
149 modelButton.addActionListener(new ActionListener()
150 {
151     public void actionPerformed(ActionEvent e)
152     {
153         JFileChooser chooser = new JFileChooser();
154         ExampleFileFilter filter = new ExampleFileFilter();
155         filter.addExtension("3ds");
156         filter.setDescription("3D Studio model files");
157         chooser.setFileFilter(filter);
158         if (chooser.showOpenDialog(modelButton) ==
JFileChooser.APPROVE_OPTION)
159             testLeft.setNewModelFromFile(chooser.getSelectedFile().getName());
160     }
161 });
162 cosButton = new JButton("new pow: on");
163 cosButton.addActionListener(new ActionListener()
164 {
165     public void actionPerformed(ActionEvent e)
166     {
167         cosButton.setText("new pow: " + (testLeft.setUseNewPower() ? "on" : "off"));
168     }
169 });
170 disableSButton = new JButton("ISC: on");

```



```
171  disableSButton.addActionListener(new ActionListener()
172  {
173  public void actionPerformed(ActionEvent e)
174  {
175  disableSButton.setText("ISC: " + (testLeft.setInnerSpecCheck() ? "on" : "off"));
176  }
177  });
178  disableMSEButton = new JButton("NMSE: off");
179  disableMSEButton.addActionListener(new ActionListener()
180  {
181  public void actionPerformed(ActionEvent e)
182  {
183  TestComponent.doMSE = !TestComponent.doMSE;
184  if (!TestComponent.doMSE) disableMSEButton.setText("NMSE: off");
185  }
186  });
187  benchmarkButton = new JButton("FPS: off");
188  benchmarkButton.addActionListener(new ActionListener()
189  {
190  public void actionPerformed(ActionEvent e)
191  {
192  if (TestComponent.doBenchmark)
193  {
194  benchmarkButton.setText("FPS: off");
195  testRight.enable();
196  }
197  else
198  {
199  benchmarkButton.setText("FPS: ...");
200  testRight.disable();
201  TestComponent.doMSE = false;
202  testLeft.resetBenchmark();
203  }
204  TestComponent.doBenchmark = !TestComponent.doBenchmark;
205  }
```

```

206 });
207 TestComponent.listener = new TestListener()
208 {
209     public void calcMSE(double value) { disableMSEButton.setText("NMSE: " +
(value > 1E10 ? "+inf" : ""+value)); }
210     public void calcFPS(double value, double renderTime, int polyCount, int
specPolyCount)
211     {
212         benchmarkButton.setText("FPS: " + value +
213         ", render time: " + renderTime + " ms,"/* +
214         polyCount + "/" + (specPolyCount == -1 ? "-" : ""+specPolyCount) + "
215         pol"*/);
216     }
217 };
218 panelButtons.add(benchmarkButton);
219 panelButtons.add(disableSButton);
220 panelButtons.add(pauseButton);
221 panelButtons.add(modelButton);
222 panelButtons.add(disableMSEButton);
223 panelButtons.add(cosButton);
224 this.getContentPane().setLayout(new BorderLayout(getContentPane(),
BoxLayout.Y_AXIS));
225 this.getContentPane().add(panel);
226 this.getContentPane().add(panelButtons);
227 this.getContentPane().add(box);
228 this.pack();
229 this.addWindowListener(new WindowAdapter() {
230     public void windowClosing(WindowEvent e)
231     {
232         System.exit(0);
233     }
234 });
235 int valueR = 0xFF & Integer.parseInt(lerField.getText());
236 int valueG = 0xFF & Integer.parseInt(legField.getText());
237 int valueB = 0xFF & Integer.parseInt(lebField.getText());

```

```
238 int value = idx3d_Color.getColor(valueR, valueG, valueB);
239 testLeft.setMaterialColor(value);
240 this.show();
241 this.toFront();
242 }
243 class MyDocumentListener implements DocumentListener
244 {
245 public void changedUpdate(DocumentEvent e)
246 {
247 try
248 {
249 if (nField.getDocument() == e.getDocument())
250 {
251 float value = Float.parseFloat(nField.getText());
252 testLeft.setNValue(value);
253 }
254 else if (lerField.getDocument() == e.getDocument() || legField.getDocument() ==
255 e.getDocument() || lebField.getDocument() == e.getDocument())
256 {
257 int valueR = 0xFF & Integer.parseInt(lerField.getText());
258 int valueG = 0xFF & Integer.parseInt(legField.getText());
259 int valueB = 0xFF & Integer.parseInt(lebField.getText());
260 int value = idx3d_Color.getColor(valueR, valueG, valueB);
261 testLeft.setMaterialColor(value);
262 } }
263 catch (Exception exc)
264 {
265 }
266 }
267 public void insertUpdate(DocumentEvent e){ changedUpdate(e);}
268 public void removeUpdate(DocumentEvent e){ changedUpdate(e);}
269 }
270 }
271 package idx3d;
272 public class Quadratic_Interpolation extends idx3d_Rasterizer
```

```

273  {
274  private idx3d_Vector h;
275  int i1, i2, i3;
276  float ii12r, ii12g, ii12b;
277  float ii13r, ii13g, ii13b;
278  float ii23r, ii23g, ii23b;
279  float liir, liig, liib;
280  float riir, riig, riib;
281  float lir, lig, lib;
282  float rir, rig, rib;
283  float i4r, i4g, i4b;
284  double Pr[];
285  double Pg[];
286  double Pb[];
287  private idx3d_Vector vA, vB, vC,nA, nB, nC ;
288  int X1,X2,X3,X4;
289  int Y1,Y2,Y3,Y4;
290  public Quadratic_Interpolation()
291  {
292  }
293  protected void GetParam()
294  {
295  i1 = calcColor(p1.n);
296  i2 = calcColor(p2.n);
297  i3 = calcColor(p3.n);
298  X1=p1.x; Y1=p1.y;
299  X2=p2.x; Y2=p2.y;
300  X3=p3.x; Y3=p3.y;
301  X1=x1; Y1=y1;
302  X2=x2; Y2=y2;
303  X3=x3; Y3=y3;
304  // X4=x4>>16; Y4=y4;
305  //////////////////////////////////////
306  double I1r = idx3d_Color.getRed(i1);
307  double I2r = idx3d_Color.getRed(i2);

```

```

308 double I3r = idx3d_Color.getRed(i3);
309 double I12r =(I1r+I2r)/2;
310 double I13r =(I1r+I3r)/2;
311 double I23r =(I3r+I2r)/2;
312 double C2=4*I12r-3*I1r-I2r;
313 double C12=I1r+I2r-2*I12r;
314 double C3=4*I13r-3*I1r-I3r;
315 double C13=I1r+I3r-2*I13r;
316 double C23=4*I1r-4*I12r-4*I13r+4*I23r;
317 double D23=2*X2*X2*Y3-2*X2*Y2*X3;
318 double D32=2*Y2*X3*X3-2*X2*X3*Y3;
319 double E23=X2*Y2*Y3-Y2*Y2*X3;
320 double E32=Y2*X3*Y3-X2*Y3*Y3;
321 Pr= new double[6];
322 Pr[4]=((4*C13*Y2-C23*Y3)*D23-(4*C12-C23*Y2)*D32      )/(E32*D23-
E23*D32);
323 Pr[5]=((4*C12*Y3-C23*Y2)*E32-(4*C13*Y2-C23*Y3)*E23)/(E32*D23-
E23*D32);
324 Pr[0]=I1r;
325 Pr[1]=(C3*X2-C2*X3)/(X2*Y3-Y2*X3);
326 Pr[2]=(C2*Y3-C3*Y2)/(X2*Y3-Y2*X3);
327 Pr[3]=(2*C12-Pr[5]*X2*X2-Pr[4]*X2*Y2)/Y2*Y2;
328 //////////////////////////////////////////////////
329 double I1g = idx3d_Color.getGreen(i1);
330 double I2g = idx3d_Color.getGreen(i2);
331 double I3g = idx3d_Color.getGreen(i3);
332 double I12g =(I1g+I2g)/2;
333 double I13g =(I1g+I3g)/2;
334 double I23g =(I3g+I2g)/2;
335 C2=4*I12g-3*I1g-I2g;
336 C12=I1g+I2g-2*I12g;
337 C3=4*I13g-3*I1g-I3g;
338 C13=I1g+I3g-2*I13g;
339 C23=4*I1g-4*I12g-4*I13g+4*I23g;
340 Pg= new double[6];

```

```

341 Pg[4]=((4*C13*Y2-C23*Y3)*D23-(4*C12-C23*Y2)*D32      )/(E32*D23-
E23*D32);
342 Pg[5]=((4*C12*Y3-C23*Y2)*E32-(4*C13*Y2-C23*Y3)*E23)/(E32*D23-
E23*D32);
343 Pg[0]=I1g;
344 Pg[1]=(C3*X2-C2*X3)/(X2*Y3-Y2*X3);
345 Pg[2]=(C2*Y3-C3*Y2)/(X2*Y3-Y2*X3);
346 Pg[3]=(2*C12-Pg[5]*X2*X2-Pg[4]*X2*Y2)/Y2*Y2;
347 ///////////////////////////////////////////////////////////////////
348 double I1b = idx3d_Color.getBlue(i1);
349 double I2b = idx3d_Color.getBlue(i2);
350 double I3b = idx3d_Color.getBlue(i3);
351 double I12b =(I1b+I2b)/2;
352 double I13b =(I1b+I3b)/2;
353 double I23b =(I3b+I2b)/2;
354 C2=4*I12b-3*I1b-I2b;
355 C12=I1b+I2b-2*I12b;
356 C3=4*I13b-3*I1b-I3b;
357 C13=I1b+I3b-2*I13b;
358 C23=4*I1b-4*I12b-4*I13b-4*I23b;
359 Pb= new double[6];
360 Pb[4]=((4*C13*Y2-C23*Y3)*D23-(4*C12-C23*Y2)*D32      )/(E32*D23-
E23*D32);
361 Pb[5]=((4*C12*Y3-C23*Y2)*E32-(4*C13*Y2-C23*Y3)*E23)/(E32*D23-
E23*D32);
362 Pb[0]=I1b;
363 Pb[1]=(C3*X2-C2*X3)/(X2*Y3-Y2*X3);
364 Pb[2]=(C2*Y3-C3*Y2)/(X2*Y3-Y2*X3);
365 Pb[3]=(2*C12-Pb[5]*X2*X2-Pb[4]*X2*Y2)/Y2*Y2;
366 }
367 protected void initRender()
368 {
369 h = idx3d_Vector.add(scene.light[0].v, scene.defaultCamera.lookat);
370 /* int dy12 = Math.abs(p1.y-p2.y);
371 int dy13 = Math.abs(p3.y-p1.y);

```

```

372 int dy23 = Math.abs(p3.y-p2.y);
373 double coordx= p1.n.x;
374 double coordy= p1.n.y;
375 double coordz= p1.n.z;

376 vA = new idx3d_Vector(p1.n2);
377 vB = new idx3d_Vector(p2.n2);
378 vC = new idx3d_Vector(p3.n2);
379 GetParam();
380 ///////////////////////////////////////////////////////////////////
381 i1 = calcColor(p1.n2);
382 i2 = calcColor(p2.n2);
383 i3 = calcColor(p3.n2);
384 if (dy12 != 0)
385 {
386 ii12r = ((float)(idx3d_Color.getRed(i2) - idx3d_Color.getRed(i1))) / (float)dy12;
387 ii12g = ((float)(idx3d_Color.getGreen(i2) - idx3d_Color.getGreen(i1))) /
(float)dy12;
388 ii12b = ((float)(idx3d_Color.getBlue(i2) - idx3d_Color.getBlue(i1))) / (float)dy12;
389 }
390 if (dy13 != 0)
391 {
392 ii13r = ((float)(idx3d_Color.getRed(i3) - idx3d_Color.getRed(i1))) / (float)dy13;
393 ii13g = ((float)(idx3d_Color.getGreen(i3) - idx3d_Color.getGreen(i1))) /
(float)dy13;
394 ii13b = ((float)(idx3d_Color.getBlue(i3) - idx3d_Color.getBlue(i1))) / (float)dy13;
395 }
396 if (dy23 != 0)
397 {
398 ii23r = ((float)(idx3d_Color.getRed(i3) - idx3d_Color.getRed(i2))) / (float)dy23;
399 ii23g = ((float)(idx3d_Color.getGreen(i3) - idx3d_Color.getGreen(i2))) /
(float)dy23;
400 ii23b = ((float)(idx3d_Color.getBlue(i3) - idx3d_Color.getBlue(i2))) / (float)dy23;
401 }
402 i4r = (dy12*ii13r) + idx3d_Color.getRed(i1);

```

```
403  i4g = (dy12*ii13g) + idx3d_Color.getGreen(i1);
404  i4b = (dy12*ii13b) + idx3d_Color.getBlue(i1); */
405  }
406  protected void prerenderFirstPart()
407  {
408  lir = idx3d_Color.getRed(i1);
409  lig = idx3d_Color.getGreen(i1);
410  lib = idx3d_Color.getBlue(i1);
411  rir = idx3d_Color.getRed(i1);
412  rig = idx3d_Color.getGreen(i1);
413  rib = idx3d_Color.getBlue(i1);
414  if (dx > 0)
415  {
416  liir = ii12r;
417  liig = ii12g;
418  liib = ii12b;
419  riir = ii13r;
420  riig = ii13g;
421  riib = ii13b;
422  }
423  else
424  {
425  riir = ii12r;
426  riig = ii12g;
427  riib = ii12b;
428  liir = ii13r;
429  liig = ii13g;
430  liib = ii13b;
431  }
432  GetParam();
433  }
434  protected void prerenderSecondPart()
435  {
436  if (dx > 0)
437  {
```



```
438  lir = idx3d_Color.getRed(i2);
439  lig = idx3d_Color.getGreen(i2);
440  lib = idx3d_Color.getBlue(i2);
441  rir = i4r;
442  rig = i4g;
443  rib = i4b;
444  liir = ii23r;
445  liig = ii23g;
446  liib = ii23b;
447  riir = ii13r;
448  riig = ii13g;
449  riib = ii13b;
450  }
451  else
452  {
453  rir = idx3d_Color.getRed(i2);
454  rig = idx3d_Color.getGreen(i2);
455  rib = idx3d_Color.getBlue(i2);
456  lir = i4r;
457  lig = i4g;
458  lib = i4b;
459  riir = ii23r;
460  riig = ii23g;
461  riib = ii23b;
462  liir = ii13r;
463  liig = ii13g;
464  liib = ii13b;
465  }
466  GetParam();
467  }
468  protected void prerenderLine()
469  {
470  renderLine();
471  lir += liir;
472  lig += liig;
```

```

473 lib += liib;
474 rir += riir;
475 rig += riig;
476 rib += riib;
477 }
478 public String getRasterizerName()
479 {
480 return "Quadratic_Interpolation";
481 }
482 protected void renderLine()
483 {
484 int cdx = xR - xL;
485 if (cdx == 0) return;
486 float iir = (rir - lir) / cdx;
487 float iig = (rig - lig) / cdx;
488 float iib = (rib - lib) / cdx;
489 float cir = lir;
490 float cig = lig;
491 float cib = lib;
492 for (x=xL;x<xR;x++)
493 {
494 double Ixyr=Pr[5]*x*x+Pr[4]*x*y+Pr[3]*y*y+Pr[2]*x+Pr[1]*y+Pr[0];
495 double Ixyg=Pg[5]*x*x+Pg[4]*x*y+Pg[3]*y*y+Pg[2]*x+Pg[1]*y+Pg[0];
496 double Ixyb=Pb[5]*x*x+Pb[4]*x*y+Pb[3]*y*y+Pb[2]*x+Pb[1]*y+Pb[0];
497 float cir = (float)Ixyr;
498 float cig = (float)Ixyg;
499 float cib = (float)Ixyb;

500 ///////////////////////////////////////////////////////////////////
501 float cir=0;
502 float cig=0;
503 float cib=0;
504 if(x==X1 ){ cir=255; cig=255; cib=255;}
505 else
506 { cir=100; cig=100; cib=100;}

```

```

507  if(x==X2 ){ cir=255; cig=255; cib=255;}
508  else
509  { cir=100; cig=100; cib=100;}
510  if(x==X3 ){ cir=255; cig=255; cib=255;}
511  else
512  { cir=100; cig=100; cib=100;}
513  if(cir<0){ cir=0;}
514  if(cig<0){ cig=0;}
515  if(cib<0){ cib=0;}
516  pos=x+offset;
517  if (z<zBuffer[pos])
518  {
519  screen.p[pos]=0xFF000000 | idx3d_Color.getColor(Math.round(cir > 255 ? 255 :
cir),
520  Math.round(cig > 255 ? 255 : cig), Math.round(cib > 255 ? 255: cib));
521  zBuffer[pos]=z;
522  if (useIdBuffer) idBuffer[pos]=currentId;
523  }
524  // cir += iir;
525  // cig += iig;
526  // cib += iib;
527  } }
528  private int calcColor(idx3d_Vector vector)
529  {
530  float angle = idx3d_Vector.angle(scene.light[0].v, vector);
531  if (angle < 0) angle = 0;
532  int c = idx3d_Color.scale(color, (int)(angle*255));
533  angle = idx3d_Vector.angle(h, vector);
534  if (angle < 0) angle = 0;
535  angle = (float)Math.pow(angle, n);
536  int c2 = idx3d_Color.scale(0xFFFFFFFF, (int)(angle*255));
537  c = idx3d_Color.add(c, c2);
538  return c;
539  }
540  protected void beforeRenderTriangle()

```

```

541  {
542  }
543  }
544  package idx3d;
545  float4 Gold: register(c0);
546  sampler BaseMap: register(s0);
547  sampler BumpMap: register(s1);
548  float4 main(float2 texCoord: TEXCOORD0, float3 lightVec: TEXCOORD1) :
COLOR {
549  float4 base = tex2D(BaseMap, texCoord);
550  float3 bump = tex2D(BumpMap, texCoord) * 2.0 - 1.0;
551  // shine = diffuse and specular
552  float shine = saturate(dot(normalize(lightVec), normalize(bump)));
553  return (shine * 0.2) * base + pow(shine, 2) * Gold;
554  }
555  float4x4 view_proj_matrix: register(c0);
556  float4x4 view_matrix: register(c4);
557  struct VS_OUTPUT {
558  float4 Pos: POSITION;
559  float2 texCoord: TEXCOORD0;
560  float3 lightVec: TEXCOORD1;
561  };
562  VS_OUTPUT main(float4 Pos: POSITION, float3 normal: NORMAL, float3
tangent:
563  TANGENT, float3 binormal: BINORMAL) {
564  VS_OUTPUT Out;
565  Out.Pos = mul(view_proj_matrix, Pos);
566  // Some object-linear texgen, specific for this particular model
567  Out.texCoord.x = Pos.x * 0.0065 + 0.46;
568  Out.texCoord.y = Pos.z * 0.0065 + 0.46;
569  float3 Pview = -mul(view_matrix, Pos);
570  // Move our tangent-space into eye-space
571  float3 vtang = mul(view_matrix, tangent);
572  float3 vbinorm = mul(view_matrix, binormal);
573  float3 vnorm = mul(view_matrix, normal);

```

```
574 // We use lightVec = viewVec, that is, the camera is also the light.
575 float3 lightVec = Pview;
576 // Transform light vector from eye-space to tangent-soace
577 Out.lightVec.x = dot(lightVec, vtang);
578 Out.lightVec.y = dot(lightVec, vbinorm);
579 Out.lightVec.z = dot(lightVec, vnorm);
580 return Out;
581 }
```

Додаток Б  
Презентація

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ імені В. ДАЛЯ  
Кафедра комп'ютерних наук та інженерії

## Методи удосконалення реалістичності відтворення кольорів у засобах комп'ютерної графіки

Виконала ст. гр. КІ-18дм  
Букша К.С.  
Науковий керівник  
Щербакова М.Є.

Рисунок Б.1 – Слайд 1

**МЕТА роботи** – збільшення реалістичності відображення кольорів у системах комп'ютерної графіки. Для досягнення мети необхідно розв'язати ряд наступних **задач**:

- провести аналіз методів зафарбовування та вже існуючих моделей освітлення у комп'ютерній графіці.
- створити нові методи та моделі освітлення на основі двопрменевої функції відбивної здатності поверхні.
- розробити засоби підвищення реалістичності відтворення моделей освітлення завдяки підвищенню фізичної адекватності.
- розробити методи перспективно-коректного зафарбування об'єктів.
- розробити програмне забезпечення для моделювання отриманих результатів.
- дослідити ефективність розроблених моделей.

**ОБ'ЄКТ дослідження** – процес зафарбовування тривимірних графічних об'єктів у системах комп'ютерної графіки.

**ПРЕДМЕТ дослідження** – високопродуктивні засоби зафарбовування тривимірних об'єктів.

Рисунок Б.2 – Слайд 2

## Класифікація ДФВЗ

ДФВЗ (двопроменева функція відбивної здатності поверхні) - функція, що обчислює оптичні властивості поверхні.

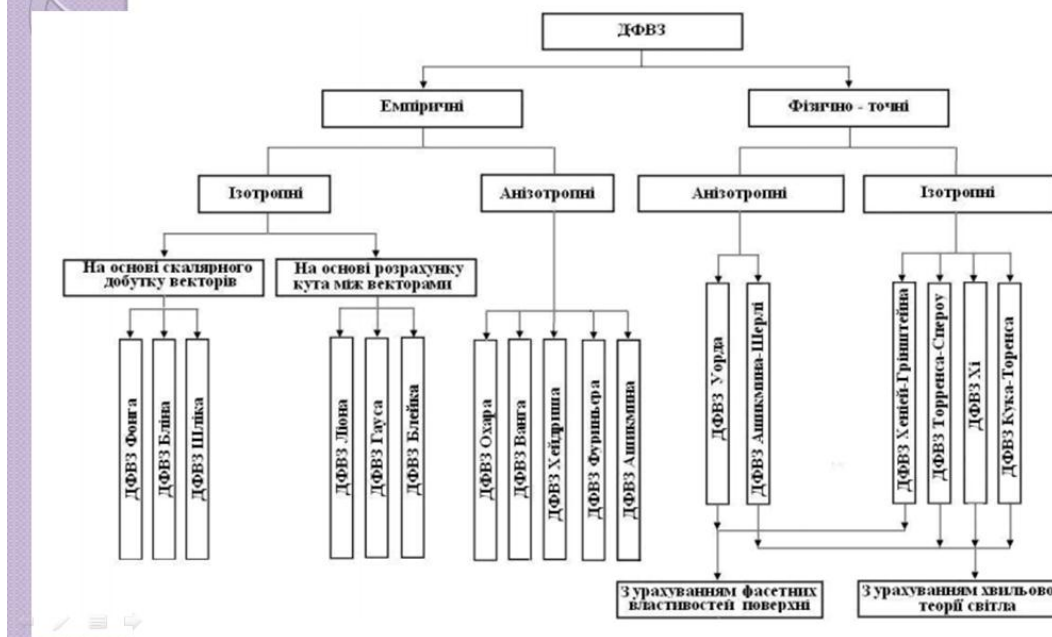
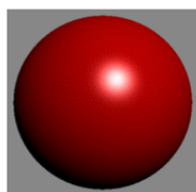


Рисунок Б.3 – Слайд 3

## Моделі двопрменевої функції відбивної здатності поверхні



## Методи зафарбовування тривимірних об'єктів



Метод Гуро

Пом'якшує тоновий перехід між ребрами ділянки, обчислюючи інтенсивність кольору для кожного пікселя, де проходить інтерполяція кольорів через полігони

Метод Фонга



Вектор нормалі до поверхні лінійно інтерполюється на багатокутник по нормалям в вершинах багатокутника. Багатокутник розкладається на трикутники.

Рисунок Б.4 – Слайд 4

## Моделі відтворення кольору

За принципом дії моделі умовно розбивають на наступні класи:

- адитивні (RGB) - побудовані на складанні кольорів;
- субтрактивні (СМУК) - основу якого становить операція віднімання кольорів;
- перцепційні (HSB) - базуються на сприйнятті кольору.

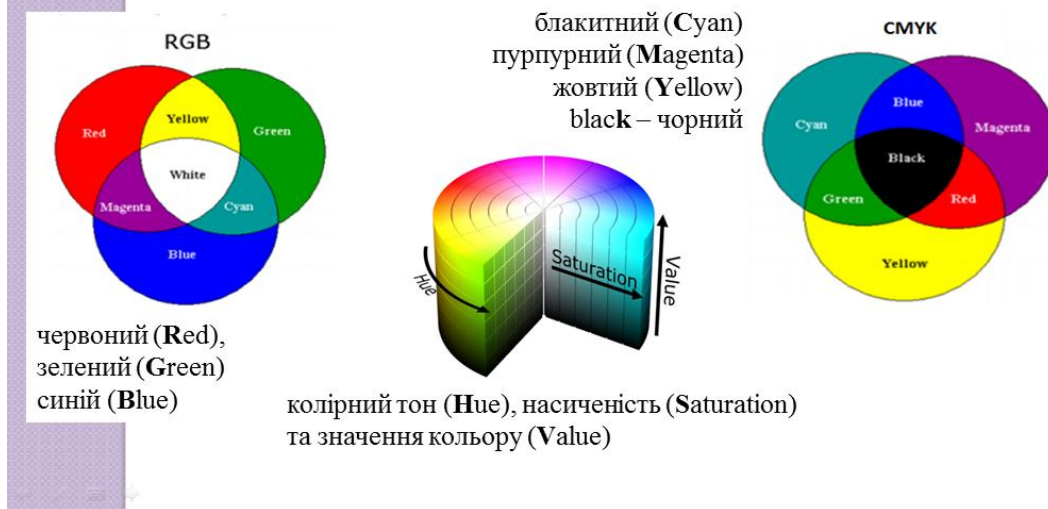


Рисунок Б.5 – Слайд 5

## Апроксимація ДФВЗ Шліка

$$\text{ДФВЗ Шліка: } S_n(\cos x) = \frac{\cos x}{n - n \cos x + \cos x}$$

**Недолік:** повільне спадання функції у зоні згасання відблиску.

**Пропонується така апроксимація:**

$$H(\gamma) = \frac{\cos \gamma}{0.5n(1 - \cos \gamma^2) + (n - n \cos \gamma + \cos \gamma)}$$

**Висновок:** апроксимована функція забезпечує більш реалістичне відтворення зони блюмінгу.

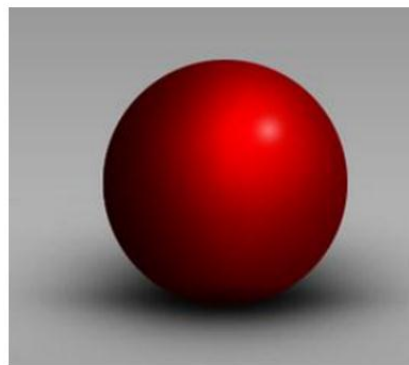
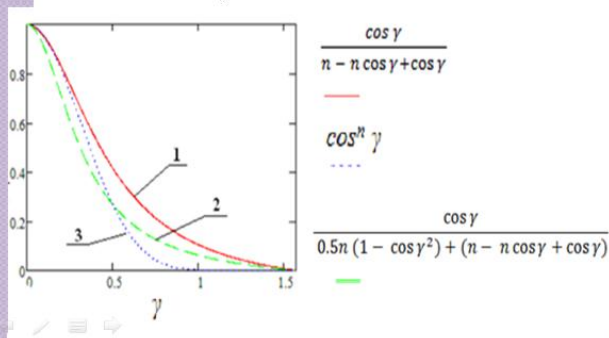
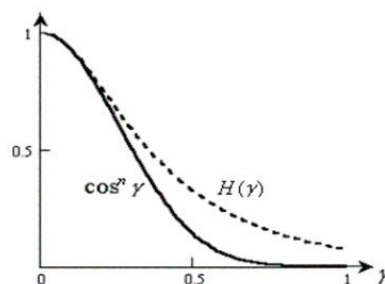


Рисунок Б.6 – Слайд 6



## Апроксимація ДФВЗ квадратичною функцією

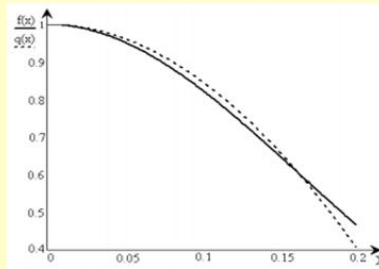
$$\cos^n x = a \cdot \cos^2 x + b \cdot \cos x + c$$

при  $x=0$   $\cos^n \gamma = 1$ , тому можна записати, що  $a+b+c=1$   
якщо  $\cos x=0$ , то  $\cos^n \gamma$  також дорівнюватиме 0, звідси:  
 $a \cdot 0 + b \cdot 0 + c = 0$   
 $c=0$ ,  $a = 1 - b$

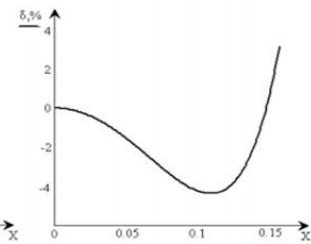


$$\gamma = \arctg\left(\frac{1}{\sqrt{n-1}}\right)$$

$$a = \frac{\cos^{n-1}\left(\arctg\left(\frac{1}{\sqrt{n-1}}\right)\right) - 1}{\cos\left(\arctg\left(\frac{1}{\sqrt{n-1}}\right)\right) - 1}$$



Квадратична апроксимація



Відносна похибка

**Недолік:** різке спадання зони затухання відблиску.

Рисунок Б.7 – Слайд 7

## Модель з використанням косинус – квадратичної функції

Для апроксимації дистрибутивної функції  $\cos^n \gamma$  використаємо косинус-квадратичну функцію  
 $\cos^n \gamma = W(\gamma, n)$   $W(\gamma, n) = (\zeta \cdot (\cos \gamma - 1) + 1)^2$   
 $(\zeta \cdot (\cos \gamma - 1) + 1)^2 = 0$   
 $\gamma = \arccos((\zeta - 1) / \zeta)$   
 $\cos^n \gamma = (\zeta \cdot (\cos \gamma - 1) + 1)^2$

Знайдемо визначений інтеграл:

$$\int_0^{\arccos\left(\frac{\zeta-1}{\zeta}\right)} [\zeta \cdot (\cos \gamma - 1) + 1]^2 \cdot \sin \gamma d\gamma = \frac{1}{3 \cdot \zeta}$$

Прирівнявши праві частини отриманих виразів, знаходимо, що:  $\zeta = (n+1)/3$

Нова формула має такий вигляд:

$$\cos^n \gamma \approx \left( \frac{(n+1)}{3} \cdot (\cos \gamma - 1) + 1 \right)^2$$

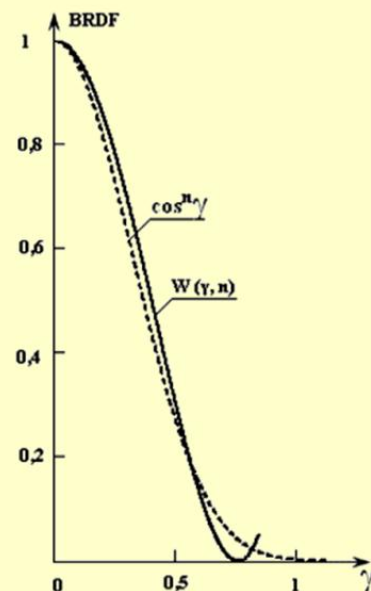
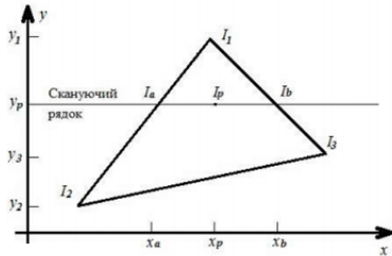


Рисунок Б.8 – Слайд 8

## Тонування за методом Гуро

**Метод тонування Гуро** - метод зафарбовування в тривимірній комп'ютерній графіці, призначений для створення ілюзії гладкої криволінійної поверхні, описаної у вигляді полігональної сітки з плоскими гранями, шляхом інтерполяції кольорів сусідніх граней.

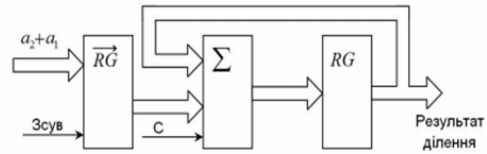


Згідно з методом Гуро кожен полігон зафарбовується шляхом лінійної інтерполяції значень інтенсивності кольору у вершинах, спочатку уздовж кожного ребра, а потім і між ребрами уздовж кожного скануючого рядка

Прирости інтенсивності вздовж рядка rasterизації трикутника (PPT)  $\Delta I = \frac{I_b - I_a}{\Delta x}$ , де  $I_b, I_a$  - значення складових інтенсивності кольору в початковій та кінцевій точках PPT;  $\Delta x$  - довжина рядка rasterизації.

Апроксимаційна формула для реалізації операції ділення

$$\frac{1}{k} = i_1 \frac{1}{2} + i_2 \frac{1}{2^2} + \dots + i_j \frac{1}{2^j} + \dots + i_n \frac{1}{2^n}$$



**Результат:** даний підхід направлений на підвищення швидкодії зафарбовування методом заміни складної операції ділення спрощеною формулою, що передбачає певну кількість накопичувальних додавань та зсувів

Рисунок Б.9 – Слайд 9

## Створення енергетично-коректної двопроменевої функції відбивної здатності поверхні

Направлена напівсферична інтегральна відбивальна здатності поверхні:

$$R(\hat{L}) = \int_{\Omega} f_r(\hat{L}, \hat{V}) \cos \theta_0 d\omega_0 \quad \text{Енергетично-коректна ДФВЗ} \quad \int_{\Omega} f_r(\hat{L}, \hat{V}) \cos \theta_0 d\omega_0 \leq 1$$

має відповідати умові:

Умова відповідності закону збереження енергії для запропонованої ДФВЗ:

$$\int_{\Omega} \left( \frac{n+1}{3} \cdot (\cos \theta_k - 1) + 1 \right)^2 \cos \theta_0 d\omega_0 \leq 1$$

Коефіцієнт для запропонованої ДФВЗ має вигляд:  $\frac{n+A}{B}$  де А і В – деякі сталі

Запропонована енергетично коректна ДФВЗ:

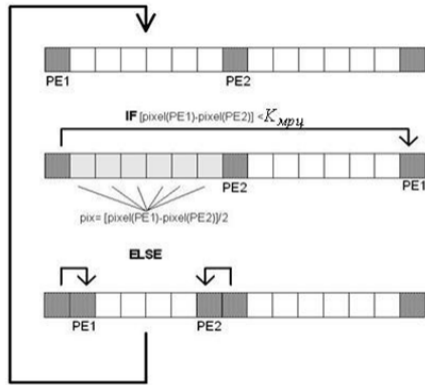
$$F(n, \gamma) = \frac{n+6}{24.55} \left( \frac{n+1}{3} \cdot (\cos \gamma - 1) + 1 \right)^2$$



Рисунок Б.10 – Слайд 10

## Апроксимований метод з'єднання зображення на основі алгоритму зворотного трасування променів

де  $C_{cp}$  – колір внутрішніх пікселів сегмента,  $RayTracer()$  – виконання трасування через центральний піксель сегмента,  $C_l$  – колір зовнішнього лівого пікселя,  $C_r$  – колір зовнішнього правого

$$C_{cp} = \begin{cases} \frac{C_l + C_r}{2}, & \text{if } K \leq 2,3 \\ RayTracer(), & \text{if } K > 2,3 \end{cases}$$


Етапи обробки одного фрагмента рядка:

- задається довжина сегмента рядка або фрагмента рядка екрану;
- виконується трасування 1-го пікселя рядка;
- перебір у циклі піксельних сегментів:

Необхідна кількість умовних процесорних елементів для трасування одного рядка  $N_{PEr}$  розміром  $X_{max}$  пікселів визначається так:

$$N_{PEr} = \left\lceil \frac{X_{max}}{\sum_{i=0}^{K_S} L_{segm_i} + 1} \right\rceil + 1, \text{ де } L_{segm_i} - \text{довжина } i\text{-го сегменту пікселів, } K_S - \text{кількість піксельних сегментів.}$$

**Результат:** метод синтезу зображень на основі алгоритму зворотного трасування променів, який використовує піксельні сегменти або піксельні блоки і додає міжпіксельну інтерполяцію в кінець трасування поточного променя або групи променів відповідно

Рисунок Б.11 – Слайд 11

## Апаратна реалізація ДФВЗ

Для запропонованої функції використовується друга степінь.

$$\left( \frac{(n+1)}{3} \times (\cos \gamma - 1) + 1 \right)^2$$

Таким чином, спостерігається суттєве підвищення продуктивності в порівнянні з використанням функції Бліна.

Запропонований метод дозволяє достатньо точно (з відносною похибкою не більше 3 %) апроксимувати не лише епіцентр відблиску, а також зону затухання. Порівняно із існуючими підходами, функція, що пропонується, має низьку обчислювальну складність.

**Схема пристрою для визначення спекулярної складової кольору та відбивної здатності поверхні у кожній точці сцени**

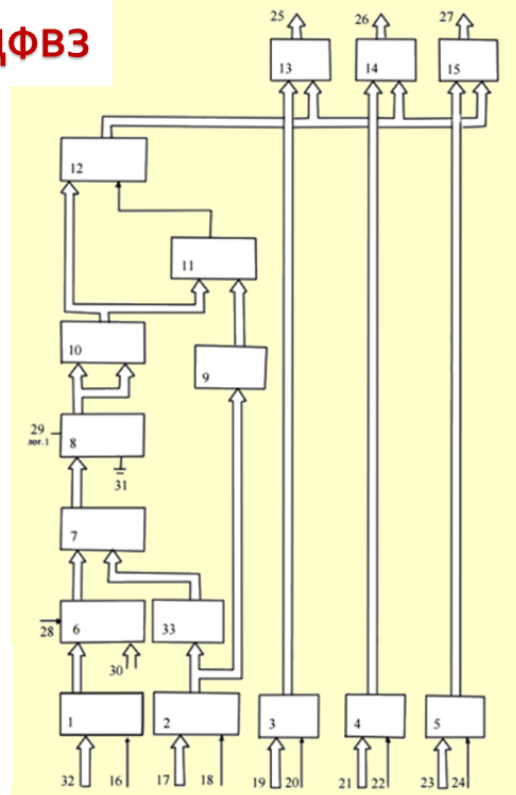


Рисунок Б.12 – Слайд 12



## Розробка програмних засобів

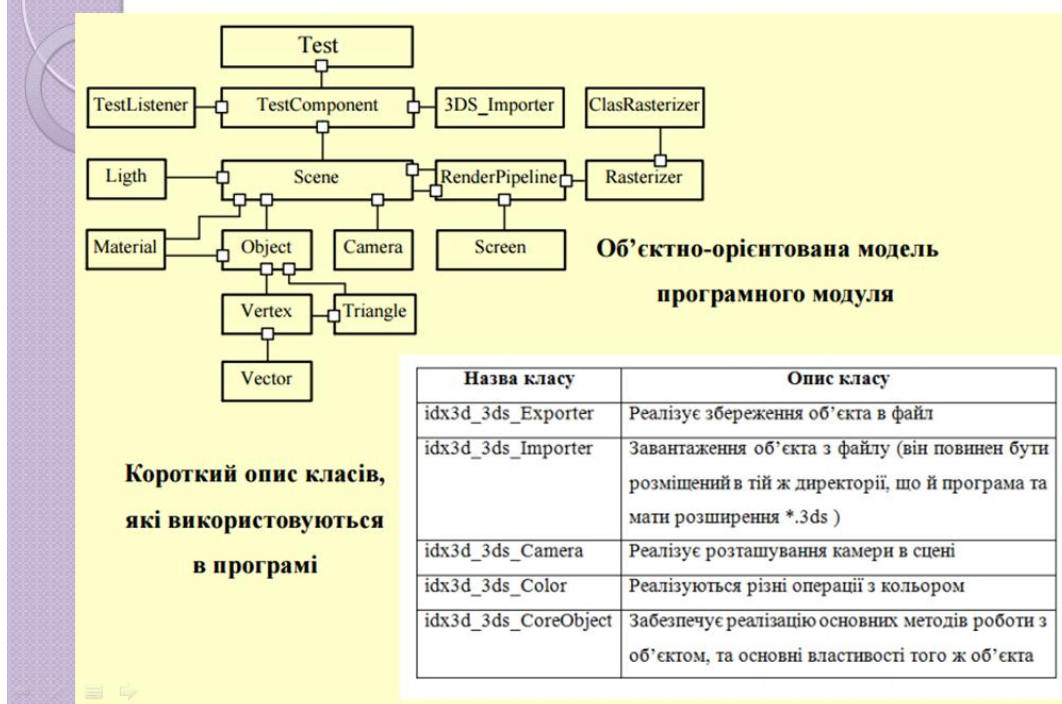


Рисунок Б.13 – Слайд 13

## Програмна реалізація методів зафарбовування в складі графічного редактора

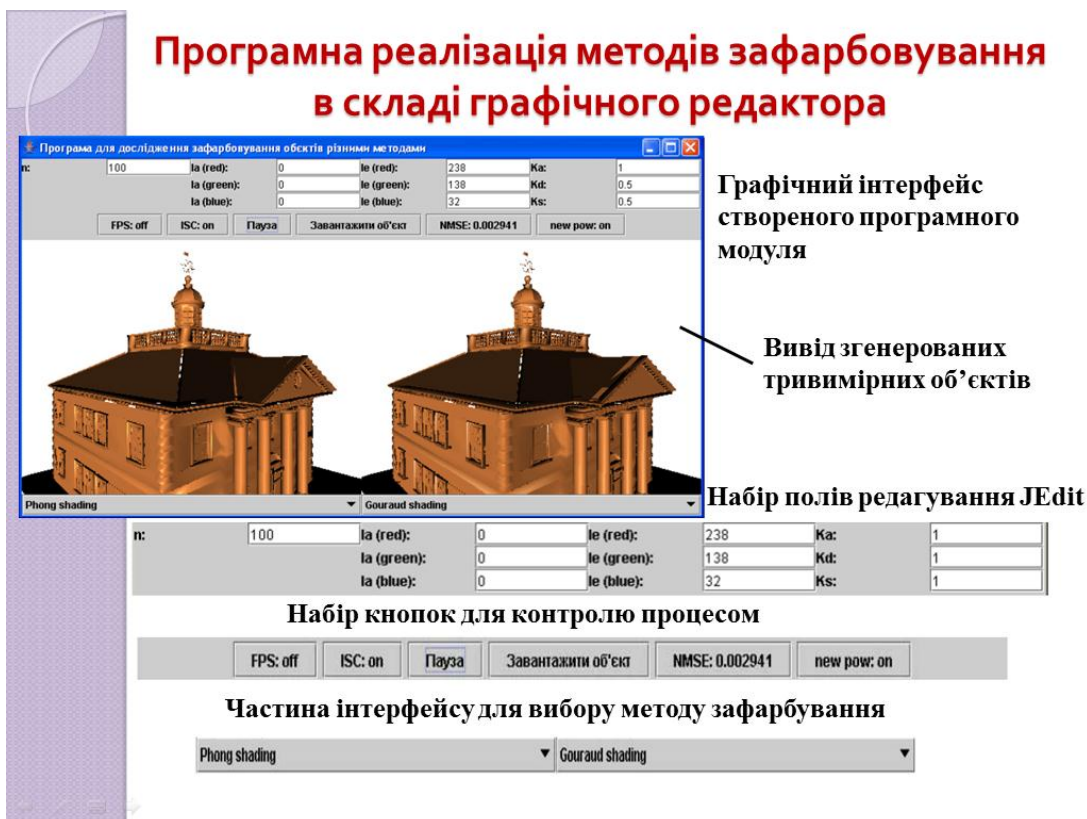
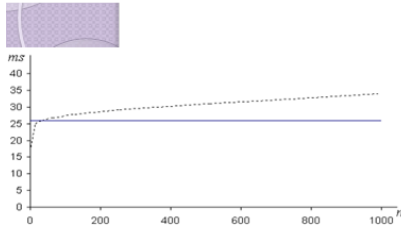


Рисунок Б.14 – Слайд 14

## Експериментальні дослідження



----- згідно базового методу;  
 — з використанням ДФВЗ типу  
 $t = \left( \frac{n+1}{3} \times (\cos \gamma - 1) + 1 \right)^2$ .

У програмному модулю візуальна відмінність між двома зображеннями оцінювалася за нормованою середньоквадратичною похибкою

$$NMSE = \frac{\sum_i (R_1(i) - R_2(i))^2 + (G_1(i) - G_2(i))^2 + (B_1(i) - B_2(i))^2}{\sum_i R_1(i)^2 + G_1(i)^2 + B_1(i)^2}$$

$NMSE \leq 0,0001$  – візуально зображення не відрізняються;  
 $NMSE = [0,0001 - 0,00025]$  – існують незначні відмінності;  
 $NMSE = [0,00025 - 0,001]$  – існують візуально помітні відмінності;  
 $NMSE \geq 0,001$  – суттєва різниця зображень один від одного.

ДФВЗ	Максимальне значення $NMSE$ при формуванні тестових фігур
Модифікація ДФВЗ Шліка: $f(y) = \frac{\cos^a x}{n - n \cos x + \cos x}$	0,00011
Косинус-квадратична апроксимація моделі Фонга $\left( \frac{n+1}{3} \times (\cos \gamma - 1) + 1 \right)^2$	0,00008

Рисунок Б.15 – Слайд 15

## Приклади сформованих зображень

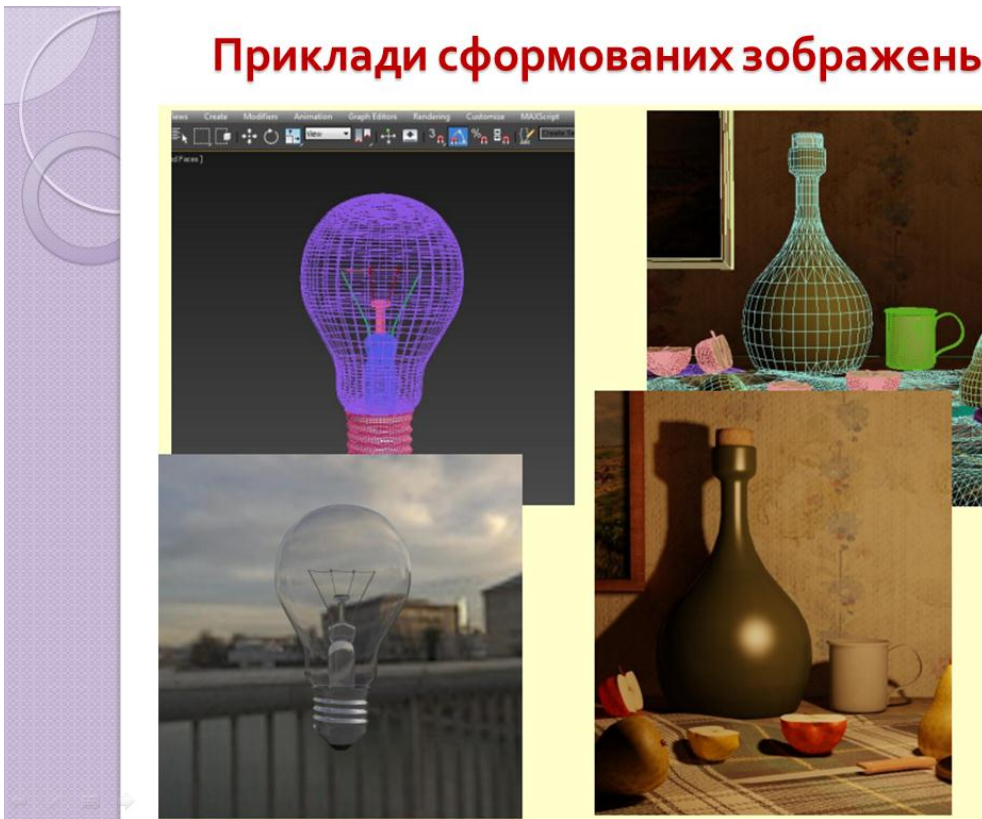


Рисунок Б.16 – Слайд 16

## Висновки

### Наукова новизна одержаних результатів:

- Подальшого розвитку дістали моделі двопроменевої функції відбивної здатності поверхні.
- Розроблено апроксимацію ДФВЗ квадратичною функцією.
- Розроблено модель з використанням косинус-квадратичної функції.
- Подальшого розвитку отримала модифікація функції Шліка, яка значно точніше відтворює як зону відблиску, так і зону блюмінгу.
- Запропоновано методи перспективно-коректного зафарбування.
- Розроблено енергетично-коректну ДФВЗ, яка забезпечує виконання закону збереження енергії, що забезпечує підвищення реалістичності.

### Практичне значення одержаних результатів:

- Розроблено шейдерні програми, які інтегровано в професійний графічний конвеєр.
- Розроблено схему пристрою для визначення спекулярної складової кольору та відбивної здатності поверхні у кожній точці сцени.
- Отримані експериментальні оцінки для розроблених моделей і методів.

### Результати опубліковано у збірниках:

науково-практичних праць ІТ – Ідея 2019,

ІХ Всеукраїнській науково-практичній конференції «ЕЛЕКТРОНІКА ТА ТЕЛЕКОМУНІКАЦІЇ»,

XXII Міжнародній науково-технічній конференції «ТЕХНОЛОГІЯ-2019»

Рисунок Б.17 – Слайд 17

**ДЯКУЮ ЗА  
УВАГУ**

Рисунок Б.18 – Слайд 18