Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інформаційних технологій та комп'ютерної інженерії
(повне найменування інституту, назва факультету (відділення))

Кафедра програмного забезпечення
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему:

«Методи та засоби високопродуктивного формування тривимірних графічних зображень»

Виконала: студентка 2-го курсу,
групи 2ПІ-20м
спеціальності 121 – Інженерія
програмного забезпечення
(шифр і назва напряму підготовки, спеціальності)
Яковенко О. О.
(прізвище та ініціали)

Керівник: д.т.н., проф., зав. каф. ПЗ
Романюк О. Н.
(прізвище та ініціали)
«____»_____2021 р.

Опонент: проф. каф. МПА
Василевський О. М.
(прізвище та ініціали)
«____»_____2021 р.

Допущено до захисту
Завідувач кафедри ПЗ
д.т.н., проф. Романюк О. Н.
(прізвище та ініціали)
«____»_____2021 р.

Вінниця ВНТУ - 2021 рік

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра програмного забезпечення

Рівень вищої освіти ІІ-й (магістерський)

Галузь знань 12 – Інформаційні технології

Спеціальність 121 – Інженерія програмного забезпечення

Освітньо-професійна програма – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
Романюк О. Н.
«13» вересня 2021 р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Яковенко Олесі Олегівні

1. Тема роботи – методи та засоби високопродуктивного формування тривимірних графічних зображень.

Керівник роботи: Романюк Олександр Никифорович, д.т.н., професор, завідувач кафедри ПЗ, затверджені наказом вищого навчального закладу від «24» вересня 2021 року № 277.

2. Строк подання студентом роботи – 1 грудня 2021 р.

3. Вихідні дані до роботи: базові методи зафарбовування – методи Гуро, Фонга; режим – TrueColor; вихідні дані для зафарбовування – вектори нормалей до вершин трикутників; серединний вектор $\vec{H}$; дані про розташування джерела світла та спостерігача; коефіцієнт спекулярності поверхі; координати вершин трикутників; максимальна розрядність для задання адрес вершин трикутників – 12; вихідні дані – кінцеве зображення, зафарбоване згідно методів Фонга та Гуро.

4. Зміст текстової частини: вступ; аналіз методів і засобів зафарбовування; методи прискореного визначення нормалізованих векторів для задач рендерингу; розробка методів прискореного зафарбовування тривимірних об'єктів; аналіз трудомісткості процедур кінцевої візуалізації; розробка програмних засобів для прискореного зафарбовування тривимірних об'єктів; економічна частина; висновки.

5. Перелік графічного матеріалу: галузі застосування методів і засобів зафарбовування; методи прискореного визначення векторів для задач рендерингу; методи прискореного зафарбовування тривимірних об'єктів програмні засоби для прискореного зафарбовування тривимірних об'єктів.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-5	Романюк О. Н., д.т.н., завідувач кафедри ПЗ	14.09.21	01.12.21
6	Бурєннікова Н. В., д.е.н, професор кафедри ЕПВМ	14.09.21	01.12.21

7. Дата видачі завдання 14 вересня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз методів і засобів зафарбовування	14.09.21 – 28.09.21	Виконано
2	Методи прискореного визначення нормалізованих векторів для задач рендерингу	29.09.21 – 12.10.21	Виконано
3	Розробка методів прискореного зафарбовування тривимірних об’єктів	13.10.21 – 26.10.21	Виконано
4	Аналіз трудомісткості процедур кінцевої візуалізації	27.10.21 – 09.11.21	Виконано
5	Розробка програмних засобів для прискореного зафарбовування тривимірних об’єктів	10.11.21 – 22.11.21	Виконано
6	Економічна частина	23.11.21 – 01.12.21	Виконано

Студентка _____ Яковенко О. О.
(підпис) (прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи _____ Романюк О. Н.
(підпис) (прізвище та ініціали)

Опонент магістерської кваліфікаційної роботи _____ Василевський О. М.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

УДК 004.92

Яковенко О. О. Методи та засоби високопродуктивного формування тривимірних графічних зображень. Магістерська кваліфікаційна робота зі спеціальності 121 – інженерія програмного забезпечення, освітня програма – інженерія програмного забезпечення. Вінниця: ВНТУ, 2021. 148 с.

На укр. мові. Бібліогр.: 36 назв; рис.: 39; табл. 20.

Проведено аналіз методів і засобів зафарбовування тривимірних об’єктів для задач високореалістичної графіки. Обґрунтовано вибір для модифікації методів Фонга та Гуро.

Запропоновано високопродуктивний метод визначення векторів для задач рендерингу, особливість якого полягає у використанні для апроксимації поліному другого ступеня, що дозволило зменшити час формування зображень графічних сцен. Отримані вирази для визначення векторів нормалей мають порівняно з існуючими меншу обчислювальну складність.

Розроблено новий метод визначення векторів у рядку растеризації, особливість якого полягає у використанні рекурентних співвідношень, що дозволяє суттєво спростити нормалізацію векторів.

Розроблено метод ідентифікації відблиску на поверхнях об’єктів, особливість якого полягає у аналізі розміщення векторів нормалей до вершин трикутника і серединного вектора, що дає можливість виключити з процесу рендерингу визначення спекулярної складової кольору, і, як наслідок, підвищити продуктивність формування графічних зображень.

Розроблено алгоритми та програмні засоби для підвищення продуктивності формування графічних зображень.

Отримані в магістерській роботі наукові та практичні результати можна використати для побудови високопродуктивних засобів рендерингу в комп’ютерних системах візуалізації тривимірних зображень.

ABSTRACT

Yakovenko O. O. Methods and means of highly productive formation of three-dimensional graphic images. Master's thesis in specialty 121 – software engineering, educational program – software engineering. Vinnytsia: VNTU, 2021. 148 p.

In Ukrainian language. Bibliographer: 36 titles; fig.: 39; table. 20.

An analysis of methods and means of painting three-dimensional objects for problems of highly realistic graphics. The choice for modification of Fong and Guro methods is substantiated.

A high-performance method for determining vectors for rendering problems is proposed, the feature of which is the use of a second degree polynomial for approximation, which allowed to reduce the time of image formation of graphic scenes. The obtained expressions for determining the vectors of normals have less computational complexity compared to the existing ones.

A new method for determining vectors in the rasterization line has been developed, the feature of which is the use of recurrent relations, which allows to significantly simplify the normalization of vectors.

A method of glare identification on the surfaces of objects has been developed, the feature of which is the analysis of the placement of normal vectors to the vertices of the triangle and the middle vector, which eliminates the definition of the specular component of color from the rendering process.

Algorithms and software tools have been developed to increase the productivity of graphic image formation.

The scientific and practical results of the master's thesis can be used to build high-performance rendering tools in computer systems for visualization of three-dimensional images.

ЗМІСТ

ВСТУП 8
1 АНАЛІЗ МЕТОДІВ І ЗАСОБІВ ЗАФАРБОВУВАННЯ 12
1.1 Основні етапи формування тривимірних графічних зображень 12
1.2 Аналіз і класифікація моделей дистрибутивної функції відбивної здатності поверхні 16
1.3 Методи зафарбування тривимірних об’єктів 23
1.4 Висновки 29
2 МЕТОДИ ПРИСКОРЕНОГО ВИЗНАЧЕННЯ НОРМАЛІЗОВАНИХ ВЕКТОРІВ ДЛЯ ЗАДАЧ РЕНДЕРИНГУ 30
2.1 Метод спрощення визначення векторів для задач рендерингу 30
2.2 Формування векторів нормалей із використанням сферично-кутової інтерполяції 34
2.3 Висновки 39
3 РОЗРОБКА МЕТОДІВ ПРИСКОРЕНОГО ЗАФАРБОВУВАННЯ ТРИВИМІРНИХ ОБ’ЄКТІВ 40
3.1. Підвищення продуктивності рендерингу Гуро 40
3.2. Метод ідентифікації відблиску на ділянці поверхні, обмеженої трикутником 46
3.3. Особливості організації обчислювального процесу по визначенню вихідних параметрів для зафарбовування 51
3.4 Висновки 57
4 АНАЛІЗ ТРУДОМІСТКОСТІ ПРОЦЕДУР КІНЦЕВОЇ ВІЗУАЛІЗАЦІЇ 59
4.1 Склад і кількість інструкцій для визначення векторів нормалей для середньостатистичного трикутника 59
4.2 Визначення векторів нормалей для середньостатистичного трикутника з використанням сферично-кутової інтерполяції 60
4.3 Визначення векторів нормалей до точок поверхні, обмеженої середньостатистичним трикутником, з використанням квадратичної інтерполяції 62

4.4 Склад і кількість інструкцій для зафарбовування середньостатистичного трикутника за методом Гуро 64
4.5 Висновки 65
5 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ПРИСКОРЕНОГО ЗАФАРБОВУВАННЯ ТРИВИМІРНИХ ОБ'ЄКТІВ 66
5.1. Розробка програмного модулю для зафарбовування тривимірних графічних об'єктів з використання розроблених методів 66
5.2 Керівництво програмісту 69
5.3 Керівництво оператору 72
5.4 Висновки 75
6 ЕКОНОМІЧНА ЧАСТИНА 76
6.1 Проведення комерційного та технологічного аудиту науково-технічної розробки 76
6.2 Розрахунок узагальненого коефіцієнта якості розробки 80
6.3 Розрахунок витрат на здійснення науково-дослідної роботи 83
6.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором 95
6.5 Висновки 99
ВИСНОВКИ 101
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 102
ДОДАТОК А 106
Технічне завдання 106
ДОДАТОК Б 110
Протокол перевірки роботи 110
ДОДАТОК В 111
Лістинг коду 111
ДОДАТОК Г. 133
Ілюстративна частина 133
ДОДАТОК Ґ. 142
Свідоцтва про реєстрацію авторського права 142

ВСТУП

Актуальність теми дослідження. Комп’ютерна графіка забезпечує реалістичне відтворення процесів і об’єктів, а тому отримала широке розповсюдження практично в усіх галузях діяльності людини.

Найбільшу реалістичність мають тривимірні зображення, у яких ілюзія об’ємності досягається за рахунок зміни кольору. На даному етапі розвитку комп’ютерної графіки досягнуто фотореалістичності формування графічних зображень. При цьому використовуються складні обчислення по визначення для кожної точки зображення як адреси координатного простору, так і значення кольорів.

Висока обчислювальна складність рендерингу обмежує його використання для формування динамічних зображень. При цьому важливо досягти прийнятної для користувача зміни кадрів за виділений проміжок часу.

Для зафарбовування графічних об’єктів найбільшого поширення отримали методи Гуро та Фонга. Ці методи використовуються практично в усіх пакетах прикладних програм комп’ютерної графіки [12].

Метод Гуро має порівняно з методом Фонга меншу реалістичність, а тому актуальною задачею є розробка методів покращення його візуальних характеристик.

Метод Фонга, який враховує кривизну поверхонь, що треба відтворити, має високу обчислювальну складність. Тому актуальною задачею є розробка методів і засобів формування графічних зображень для методу Фонга. При цьому важливо зберегти високу реалістичність.

Підвищення реалістичності та продуктивності знаходяться в протиріччі, тому важливо досягти прийнятного компромісу.

Таким чином, розробка нових високопродуктивних методів і засобів зафарбовування реалістичних графічних зображень є актуальними.

Зв’язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення.

Мета і завдання дослідження. Метою роботи є підвищення продуктивності та реалістичності формування тривимірних графічних зображень за рахунок розробки нових методів і засобів.

Задачі дослідження.

1. Проаналізувати методи та засоби зафарбовування тривимірних об’єктів.

2. Розробити методи прискореного визначення нормалізованих векторів для задач рендерингу.

3. Методи прискореного зафарбовування тривимірних об’єктів.

4. Розробити програмні засоби для прискореного зафарбовування тривимірних об’єктів.

5. Провести тестування розроблених методів з метою отримання порівняльних оцінок.

Об’єкт дослідження – процес зафарбовування тривимірних графічних об’єктів.

Предмет дослідження – методи і засоби зафарбовування тривимірних графічних об’єктів.

Методи дослідження. У магістерській кваліфікаційній роботі використовувалися: теорія диференціального числення, аналітична геометрія, теорія чисел для розробки методів зафарбовування; теорія алгоритмів для розробки програм зафарбовування та нормалізації векторів; комп’ютерне моделювання для перевірки теоретичних положень.

Наукова новизна результатів, отриманих у магістерській кваліфікаційній роботі:

1. Запропоновано високопродуктивний метод визначення векторів для задач рендерингу, особливість якого полягає в використанні для апроксимації поліному другого ступеня, що дозволило зменшити час формування графічних

зображень. Отримані вирази для визначення векторів нормалей мають порівняно з існуючими меншу обчислювальну складність.

2. Розроблено новий метод визначення векторів у рядку растеризації, особливість якого полягає у використанні рекурентних співвідношень, що дозволяє суттєво спростити нормалізацію векторів.

3. Розроблено новий метод підвищення реалістичності рендерингу Гуро, особливість якого полягає у визначені перетину відблиску сторонами трикутника з метою додаткової тріангуляції, що дозволило підвищити реалістичність формування графічних зображень за рахунок урахування спекулярної складової кольору.

4. Розроблено метод ідентифікації відблиску на поверхнях об’єктів, особливість якого полягає у аналізі розміщення векторів нормалей до вершин трикутника і серединного вектора, що дає можливість виключити з процесу рендерингу визначення спекулярної складової кольору, і, як наслідок, підвищити продуктивність формування графічних зображень.

Практичне значення одержаних результатів. Розроблено алгоритми та програмні засоби для реалізації розроблених в роботі високопродуктивних методів зафарбовування тривимірних об’єктів.

Особистий внесок. Усі наукові результати отримано здобувачем самостійно. У працях, написаних у співавторстві, автору належить: алгоритм ідентифікації відблиску [14]; алгоритм прискореного зафарбовування [15]; метод спрощення визначення векторів для задач рендерингу [16]; аналітичні вирази для визначення інтенсивності кольору [18]; алгоритм для рендерингу поверхонь тривимірних фігур [19]; алгоритм для зафарбовування поверхонь тривимірних об'єктів за модифікованим методом Гуро [20]; алгоритм для адаптивної нормалізації векторів у рядку растеризації [21]; алгоритм для зафарбовування поверхонь тривимірних об'єктів за модифікованим методом Фонга [22]; алгоритм для розрахунку дистрибутивної функції відбивної здатності поверхні для задач рендерингу [23]; алгоритм для розрахунку спекулярної складової кольору з використанням нової моделі відбивної

здатності поверхні [24]; алгоритм для текстурування з урахуванням перехресного співвідношення чотирьох колінеарних точок [25].

Апробація роботи матеріали магістерської кваліфікаційної роботи доповідались на конференціях: Міжнародна науково-практична інтернет-конференція “Електронні інформаційні ресурси: створення, використання, доступ. Пам’яті Олексія Петровича Стахова”, Міжнародна науково-практична конференція “Інформаційні технології і автоматизація – 2021”, Всеукраїнська науково-практична інтернет-конференція студентів аспірантів та молодих науковців “Молодь в науці: дослідження, проблеми, перспективи (МН-2020)”, Міжнародна науково-практична конференція молодих вчених та студентів “Молодь у світі сучасних технологій”.

Публікації. За тематикою дослідження опубліковано 14 наукових праць. У тому числі: 1 – у ваховому виданні, 6 – у матеріалах конференції, 7 – свідоцтва про реєстрацію авторського права на комп’ютерну програму.

1 АНАЛІЗ МЕТОДІВ І ЗАСОБІВ ЗАФАРБОВУВАННЯ

1.1 Основні етапи формування тривимірних графічних зображень

Комп’ютерна графіка трансформувалась від креслення простих ліній до побудови високодеталізованих сцен та віртуальної реальності. Чим більш високореалістичне зображення, тим більшу кількість обчислень необхідно провести для його формування. Побудова в графічній системі зображення, наближеного до реальності, призвела б до величезних обчислювальних витрат. Тому на практиці для формування зображень використовується ряд спрощених методів і засобів [1, 2].

Для побудови тривимірних графічних зображень необхідно дотримуватись правильного порядку дій, що в об’єднанні створюють графічний 3D-конвеєр. Майже всі програми для формування тривимірних зображень застосовують саме конвеєрну архітектуру.

Графічний конвеєр [2] – це концептуальна схема, яка описує, які етапи необхідно здійснити графічній системі для рендерингу тривимірного зображення на двовимірний екран. Розрахунки в конвеєрі розподілені на декілька етапів, на кожному із таких етапів апаратно або програмно здійснюється визначена функція. При певному виконанні на програмному й апаратному рівнях можуть бути деякі розрізнення, проте зміст етапів конвеєра майже не змінюється. Зазвичай переважна кількість етапів конвеєра реалізована апаратно, це дає можливість проводити оптимізацію. Схема графічного конвеєра найчастіше застосовується під час візуалізації у режимі реального часу. Деякі етапи конвеєру здійснюються паралельно, проте блокуються до того часу, поки не буде здійснений найменш швидкий крок.

Виділяють такі етапи [1, 2] при формуванні тривимірної зображення: підсистема опису сцени, геометрична підсистема та підсистема візуалізації (рисунок 1.1).

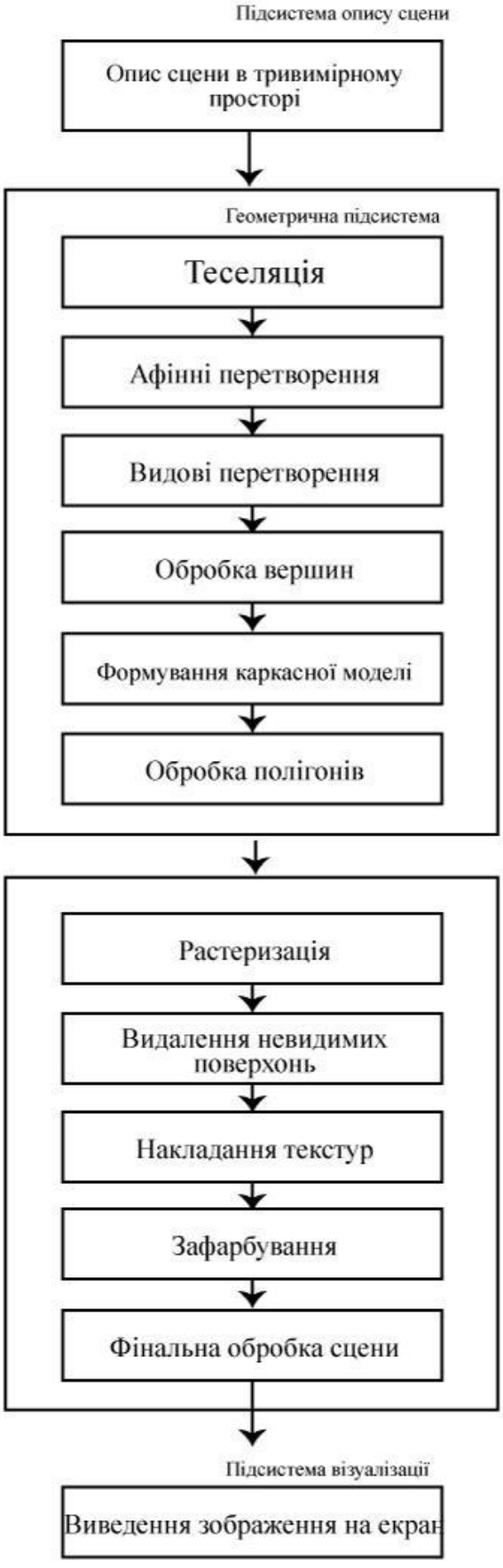


Рисунок 1.1 – Графічний конвеєр

На етапі опису тривимірного зображення визначаються стани складових об’єктів та взаємне розташування, також визначається стратегія подальших дій над об’єктами. На етапі геометричних перетворень здійснюють декомпозицію графічної сцени, а також реалізують афінні перетворення над отриманими об’єктами. Після перетворення з глобального простору в простір спостерігача здійснюють відсікання та вилучення невидимих граней, а також конвертують отримані результати в екранний простір. Далі отримують параметри вершин тривимірної графічної сцени та їх розташування в екранній системі координат, текстурні координати, вектори нормалей, освітленість. Одними із найважливіших процедур на цьому етапі вважаються процедури перетворень і освітлення. Далі здійснюється збір трикутників та формування каркасної моделі. На етапі обробки полігонів оброблюються вхідні примітиви як цілісні об’єкти та, при необхідності, формуються нові. Даний етап з’явився у графічному конвеєрі відносно недавно, після появи стандартів DirectX 10 і Open GL 2.1 [3, 13].

Етап рендерингу [2] – це етап кінцевої візуалізації, на даному етапі формуються пікселі зображення, для яких визначаються екранні координати та інтенсивності кольору.

Останнім часом при формуванні тривимірних зображень застосовують технологію шейдерів. Шейдер [4] – це програма для виконання однієї із стадій графічного конвеєра, що використовується в тривимірній графіці для знаходження кінцевих параметрів зображення або об’єкту. Для того, щоб обробити вершини, полігони і їх пікселі, існують відповідно вершинний (вертексний), геометричний і піксельний шейдери. Максимальне навантаження припадає на останні шейдери як по кількості точок, що обробляються, так і по складності розрахунків. Піксельний шейдер найбільше з’ясовує продуктивність формування тривимірних графічних зображень.

Для того, щоб досягти високої реалістичності в комп’ютерній графіці потрібно точно відобразити властивості поверхні, а також правильно представити ефекти освітлення на сцені. Фактично ефекти освітлення

представляються моделями, в яких обчислюється взаємодія електромагнітної енергії з поверхнею об’єкта.

Згідно з визначеним у комп’ютерній графіці методом, обчислення освітленості точок поверхні розбивається на дві основні задачі. Перша задача відображає знаходження способу обчислення освітленості в даній точці тривимірного простору та розраховується завдяки побудові математичної моделі освітлення. Друга – застосовує модель освітлення для обчислень освітленості тривимірних графічних сцен із певною геометрією й ознаками поверхні та визначається завдяки моделі зафарбовування (Shading model) [5].

Модель освітлення дає можливість обчислити інтенсивність світла, яке випромінюється з встановленої точки поверхні в заданому напрямку спостереження для даних оптичних характеристик поверхонь, відносного знаходження поверхні на сцені, кольору та знаходження джерела світла, властивостей джерел світла та орієнтації площини спостереження. Для зниження об’ємів розрахунків, у багатьох пакетах застосовують емпіричні моделі, засновані на спрощених фотометричних обчисленнях.

При застосуванні глобальних схем освітленості тривимірна сцена визначається як одна система, для якої відображають освітлення враховуючи взаємний вплив об’єктів. Також обчислюють багаторазове відбиття й переломлення світла. Використовуючи такий підхід, отримують зображення високої якості, проте для цього необхідно великий обсяг обчислень і, в результаті, багато часу для формування тривимірного графічного зображення. Тому широке застосування таких моделей у системах реального часу потребує застосування потужних апаратних засобів і розпаралелення обчислювального процесу.

Взаємодія має обмеження лише в разовому відбитті світла від поверхні при побудові графічних зображень, у яких застосовують локальні моделі освітлення [28]. Також обчислюється дифузна й спекулярна складові кольору, а розсіяне світло апроксимується. Такі моделі найпоширеніші нині і використовуються в системах реального часу.

1.2 Аналіз і класифікація моделей дистрибутивної функції відбивної здатності поверхні

Двопроменева функція відбивної здатності поверхні (ДФВЗ) [6] – функція, яка розраховує оптичні властивості поверхні. ДФВЗ має такі параметри:

L_i – вектор вхідного світла;

V_i – вектор відбитого світла.

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

де λ – довжина хвилі, (α,μ_1), (ϕ,μ_2) – параметри, які дорівнюють напрямку падаючого світла і спостереження [2].

Інтенсивність випромінювання I можна визначити як величину променевого потоку через ділянку $ds_\perp$ (рисунок 1.2), перпендикулярну променю, в межах диференціального тілесного кута $d\omega$, що дорівнює площі нескінченно малого елемента на поверхні одиничної сфери

$$I = \frac{d\phi}{ds_\perp d\omega} = \frac{d\phi}{ds\cos\alpha d\omega}, \tag{1.2}$$

де $d\omega = \sin\alpha d\alpha d\mu_1$

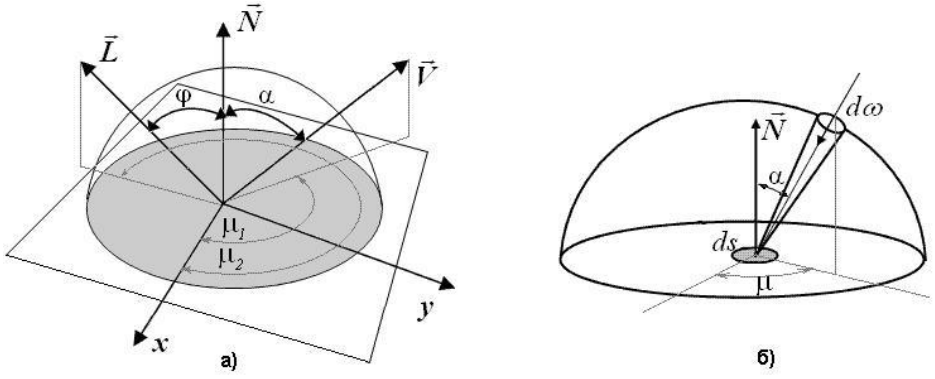


Рисунок 1.2 – Вихідні дані для обчислення ДФВЗ

Для моделювання освітлення застосовуються способи, що орудують з розсіяним світлом та відбитим. Відбите світло має дві складові: дифузна та

спекулярна. Розсіяне (фонове) світло – це світло, що відбивається від поверхні об’єктів. Моделювання даної компоненти потребує значних обчислювальних затрат, тому при застосуванні локальних моделей освітлення апроксимується.

Модель Ламберта [6] моделює ідеальне дифузне освітлення. Вважається, що світло при направленні на поверхню розсіюється рівномірно. Модель Ламберта є однією з найпростіших моделей освітлення, адже дана модель часто є основою інших моделей, практично у будь-якій іншій моделі освітлення можна виділити дифузну складову.

При обчисленні освітлення за Ламбертом враховується тільки нормаль до поверхні N і напрямок на джерело світла L. Інтенсивність дифузійного відбиття розраховується за законом косинусів (закон Ламберта)

$$I_d = I_0 \cdot k_d \cdot \cos\theta, \tag{1.3}$$

де I_0 – інтенсивність джерела світла, $k_d \in [0,1]$ – коефіцієнт дифузійного відбиття, θ – кут між напрямком джерела світла L та нормаллю до поверхні N (рисунок 1.3.). Якщо N та L нормалізовано, то $\cos\theta = N \cdot L$.

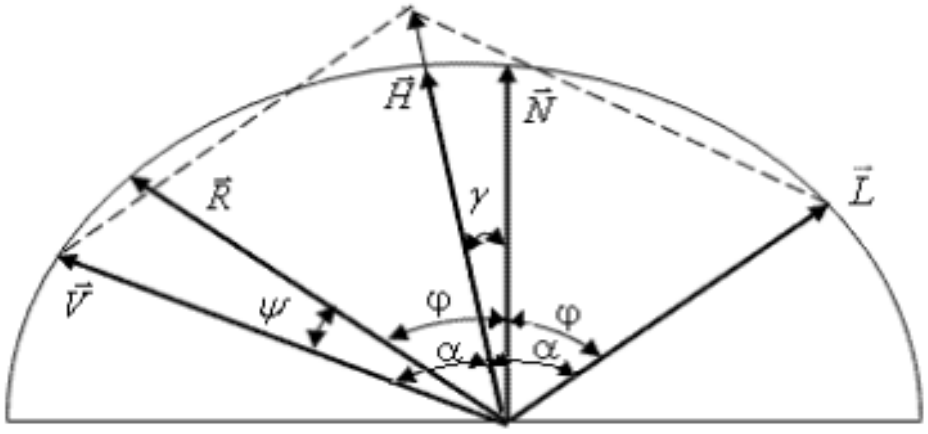


Рисунок 1.3 – Нормалі до поверхні

При використанні моделі освітлення Ламберта, поверхня об’єкта відображається рівномірно яскравою. У реальному світі таке освітлення майже не зустрічається. Модель Ламберта [6] враховує тільки рівномірне розсіювання світла (рисунок 1.4).

Рисунок 1.4 – Освітлення об’єкта із використанням моделі Ламберта

Модель освітлення Фонга [29] – це емпірична модель локального освітлення поверхні об’єкта. Дана модель забезпечує більш гладку зміну вектора нормалі до поверхні. Для блискучих об’єктів дзеркальна складова дуже велика і інтенсивність зменшується занадто швидко. Проте для негладких поверхонь дифузна складова є більшою і інтенсивність зменшується повільно. Фонг вирішив дану проблему ускладнивши модель Ламберта, що має тільки вектор спостереження V і вектор напрямку світла L, додавши третю складову [2,7,30] – вектор відбитого світла R (рисунок 1.5, рисунок 1.6).

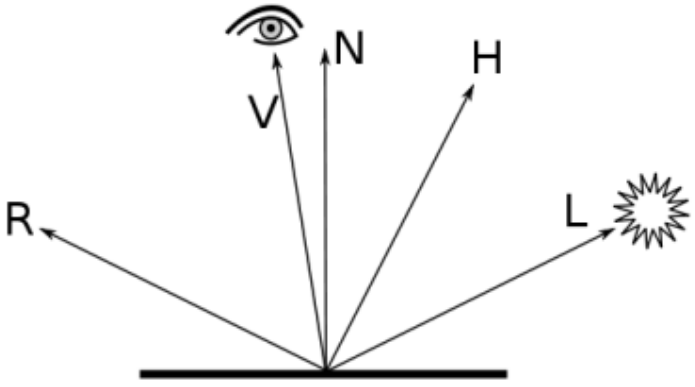


Рисунок 1.5 – Модель освітлення Фонга

Рисунок 1.6 – Приклад із використанням моделі освітлення Фонга

Більшість моделей освітлення засновано на мікрофасетній моделі поверхні [31]. Відповідно до цієї моделі, вся поверхня складається з мікроскопічних граней. Всі ці грані вважаються випадково орієнтованими, але при цьому задається закон розподілу для нормалей H цих граней (рисунок 1.7) Також вважається, що кожна грань задовольняє дуже простій моделі освітлення.

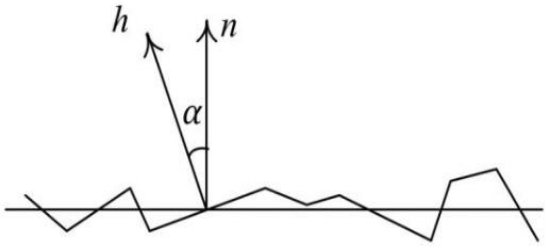


Рисунок 1.7 – Мікрофасетна модель

У довільній точці поверхні є нормаль N до середньої лінії поверхні і нормаль H до конкретної мікрограні. Вектор H є випадковою величиною, закон розподілу якої $D(\alpha)$ є функцією кута між векторами H і N.

У моделі Торренса-Спарроу [32, 33] передбачається, що мікрограні орієнтовані відповідно до деякої функції розподілу, яка описує можливе відхилення нормалі до мікрограні від нормалі до поверхні. Чим більше ця

функція полога, тим більші відхилення допустимі і тим більшої величини пляма відбитого відблиску.

Модель ДФВЗ визначає вектор H між вектором падаючого випромінювання L та вектором спостереження V, вектор H утворює кут θ із векторами L і V. Формула (1.4) мікрофасетної моделі Торренса-Спарроу:

$$f(L,V) = diffuse + \frac{D(\theta_H)F(\theta_d)G(\theta_L,\theta_V)}{4\cos\theta_L\cos\theta_V}, \quad (1.4)$$

де $diffuse$ – функція невідомої форми, D – мікрофасетна дистрибутивна функція, що розраховує найвищу точку спекулярної функції, F – коефіцієнт Френеля, G – фактор затемнення, θ_L і θ_V – кути L і V відносно N, θ_H – кут між N і H, θ_d – кут між L і V [31] (рисунок 1.8).

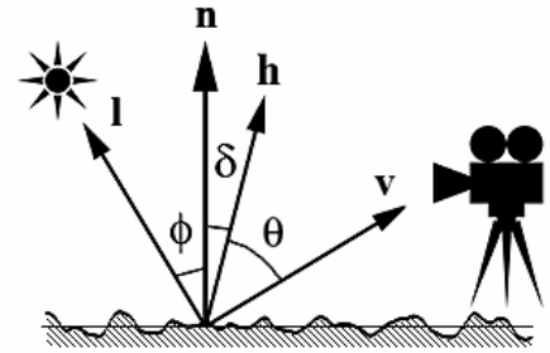


Рисунок 1.8 – Розрахунок освітлення

Модель освітлення Кука-Торренса [33] використовується для розрахунку відбитого світла, розсіяне світло обчислюється за класичною формулою Ламберта, в якій освітленість точки залежить тільки від кута між нормаллю до поверхні в даній точці і положенням джерела світла. Поверхня, що відбиває світло, моделюється як сукупність блискучих мікрограней, орієнтованих в різних напрямках. Кожна мікрограні відображає падаюче світло, і тільки ті грані, які орієнтовані належним чином, відбивають світло в сторону спостерігача і вносять внесок у освітленість [33]. Обчислюється загальна освітленість R:

$$R = sR_s + dR_d, \tag{1.5}$$

де s – ваговий коефіцієнт дзеркального відображення; d – ваговий коефіцієнт дифузного відбиття, при чому: $s + d = 1$, R_s – дзеркальна складова освітленості, R_d – дифузна складова освітленості.

Дифузна складова розраховується за законом Ламберта, а дзеркальна складова за формулою (1.6):

$$R_s = \frac{FDG}{(N,V)}, \tag{1.6}$$

де F – коефіцієнт Френеля, D – частка граней, які орієнтовані до спостерігача, G – коефіцієнт затемнення, N – нормаль до поверхні, V – одиничний вектор, направлений у бік спостерігача.

Коефіцієнт Френеля [33] залежить від коефіцієнта заломлення матеріалу для довжини хвилі, що дорівнює довжині хвилі падаючого світла:

$$F = \frac{1(g-c)^2}{2(g+c)^2}\left\{1+\left[\frac{c(g+c)-1}{c(g-c)+1}\right]^2\right\}, \tag{1.7}$$

де $c = cos(\theta) = (H,V)$, $g = \eta^2 + c^2 + 1$, η – коефіцієнт заломлення, c – косинус кута між вектором V і H,

$$H = \frac{L+V}{|L+V|}, \tag{1.8}$$

де L – одиничний вектор, спрямований в бік джерела світла.

Щоб обчислити коефіцієнт заломлення, потрібно знати коефіцієнт Френеля F_0 для нормального падіння світла. У цьому випадку коефіцієнт заломлення може бути виражений як:

$$\eta = \frac{1+\sqrt{F_0}}{1-\sqrt{F_0}}. \tag{1.9}$$

Частка граней, орієнтованих у напрямку до спостерігача, може бути обчислена за однією з функцій розподілу, наприклад, по функції розподілу Гауса або Бекмана. Розподіл обчислюється за формулою (1.10):

$$D = \frac{1}{m^2 \cos^4 \delta} e^{-\left[\frac{\tan\delta}{m}\right]^2}, \qquad (1.10)$$

де m – середньоквадратичний нахил мікрограней, від 0,2 для гладких матеріалів до 0,6 для шорсткуватих.

Модель освітлення Кука-Торренса [33] є трудомісткою з обчислювальної точки зору, проте найточніше описує взаємодію світла і матеріалу для широкого кола матеріалів.

Модель Орена-Найара [34] є розширенням моделі Ламберта. Відповідно до цієї моделі, кожна мікрограні є ідеальним дифузним рефлектором. Дана модель містить параметр для контролю шорсткуватості поверхні, який визначає, скільки світла відіб'ється в напрямку джерела [34]. Чим більший ступінь шорсткуватості поверхні, тим виразнішим є дифузне відбиття. Шорсткувата поверхня розсіює світло в усіх напрямках, проте нерівномірно. Модель Орена-Найара – це спрощене відображення реальності (рисунок 1.9).

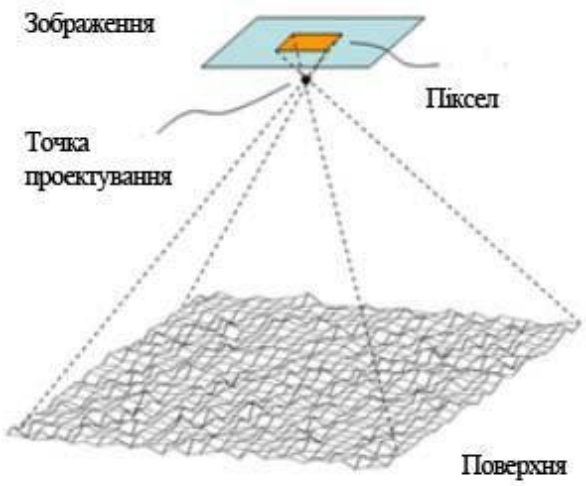


Рисунок 1.9 – Агрегація відображення від шорсткуватих поверхонь

Враховується те, що напрямок до спостерігача V впливає на видиму освітленість (на відміну від класичної моделі Ламберта). Як показано на рис. 1.10, при одному і тому ж положенні джерела світла освітленість залежить

від положення спостерігача. Коли спостерігач знаходиться праворуч, то він бачить в основному освітлені межі, коли він знаходиться ліворуч – неосвітлені.

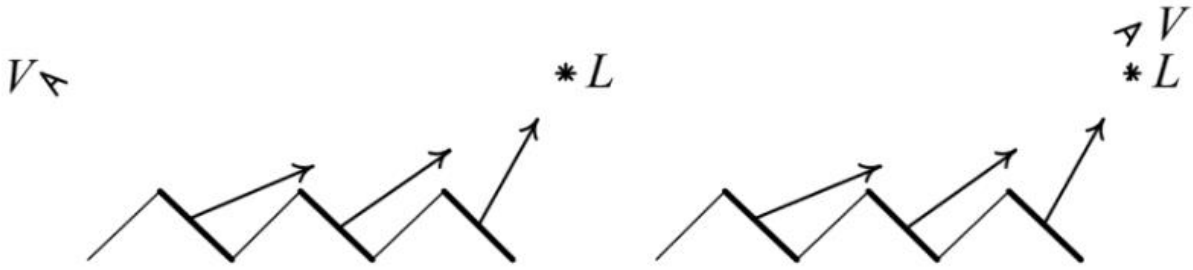


Рисунок 1.10 – Вплив положення спостерігача на видиму освітленість

Шлік запропонував апроксимувати функцію $cos^n \gamma$, яку використовують для розрахунку інтенсивності дзеркальної складової кольору в моделі освітлення Бліна, функцією [35]:

$$H(\gamma) = \frac{cos\,\gamma}{n - n\cos\gamma + \cos\gamma}. \tag{1.11}$$

Характерна особливість такого підходу полягає у тому, що функція визначається через $cos\,\gamma$ і значення коефіцієнта спекулярності n, а не через кут γ, що притаманно практично всім відомим дистрибутивним функціям. Функція Шліка є монотонно спадною, що відповідає характеру зміни спекулярної складової кольору.

1.3 Методи зафарбування тривимірних об’єктів

Генерація об'ємних зображень [8] – досить складна обчислювальна задача, тому на практиці виконують її декомпозицію. Здебільшого частини поверхні графічних об’єктів апроксимують мережею трикутників, які після трансформацій зі світової системи растеризують в екранній системі координат. Складні зображення розбивають на складові частини. Такий процес розбиття поверхні об'єктів на полігони отримав назву теселяції [9]. Одним із найбільш поширених видів теселяції є тріангуляція – розбиття поверхні на трикутники,

так як цей процес найпростіший та містить багато переваг над використанням інших форм полігонів. Деталізація об'єктів, що характеризується щільністю тріангуляційної мережі, має бути достатньою для правильного відображення всіх особливостей об'єкту.

Розбиття поверхні саме на трикутники пояснюється наступними причинами [9]:

– трикутник є найпростішим полігоном, вершини якого однозначно задають грань;

– будь-яку область можна гарантовано розбити на трикутники;

– обчислювальна складність алгоритмів розбиття на трикутники істотно менша, ніж при використанні інших полігонів;

– реалізація процедур рендерингу найбільш проста для області, обмеженою трикутником;

– для трикутника легко визначити три найближчі сусіда, що мають із ним спільні грані.

Тому, що тріангуляція є початковим етапом рендерингу, правильно обрати внутрішню точку таку, щоб площа складових фігур була приблизно однаковою. У такому випадку буде забезпечена однакова ступінь обчислювального завантаження апаратури.

Метод однотонного зафарбовування був одним із перших [3]. Для кожного плоского трикутника розраховується вектор нормалі, а на його основі визначається колір. Проте при використанні такого методу інтенсивності кольору різко змінюються на межах трикутників, адже при заповненні трикутників об’єкта одним кольором не виконувалась його градація. Таке зафарбовування потребує мінімальних затрат часу на обчислення, проте має низьку якість.

Найпоширенішими методами зафарбовування тривимірних обєктів у комп’ютерній графіці є метода зафарбування Гуро [36] і Фонга [29].

Метод Гуро забезпечує плавний тоновий перехід між ребрами полігона, обчислюючи інтенсивність кольору для кожного пікселя, використовуючи

засоби лінійної інтерполяції. Одержана поверхня відображає природнє відбиття світла, тому візуалізація виглядає більш реалістичною.

Процес зафарбовування містить такі етапи:

1. обчислення векторів нормалей до кожної із граней;
2. знаходження нормалі у вершинах багатокутника, завдяки усереднені нормалей усіх граней;
3. обчислення інтенсивності кольору у вершинах багатокутника, використовуючи значення нормалей;
4. зафарбовування ділянки, яка обмежена багатокутником, використовуючи лінійну інтерполяцію інтенсивностей кольору вздовж ребер та між ребрами вздовж кожного рядка растеризації.

Форма відблисків залежить від вибору багатокутників, які розбивають поверхню об’єкта на частини, тому зафарбовування за методом Гуро потрібно застосовувати при використанні простої моделі освітлення з дифузійним відбитком. На рисунку 1.11 зображено приклад зафарбовування об’єктів за методом Гуро.

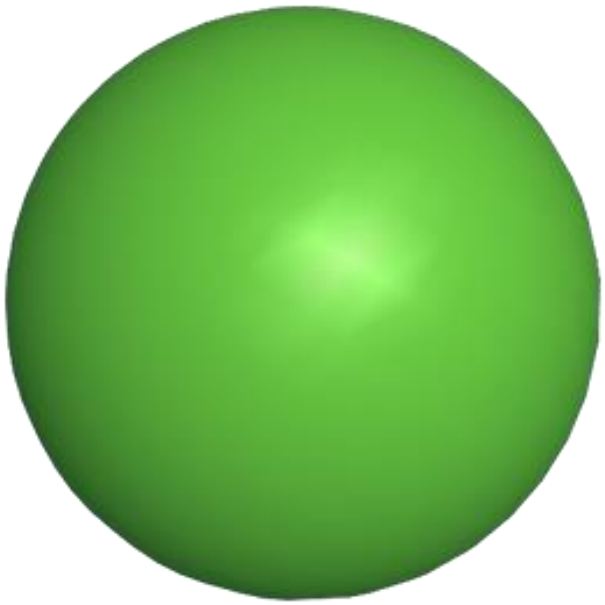


Рисунок 1.11 – Зафарбовування за методом Гуро

Метод Гуро має такі недоліки:

– застосування лінійної інтерполяції для обчислення інтенсивностей, коли частини кольору, такі як спекулярна та дифузна, містять нелінійний характер зміни;

– не враховується локальна кривизна поверхні, оскільки вектори нормалей визначаються тільки для вершин трикутника;

– відблиски відтворюються тільки в разі, якщо вершини трикутників знаходяться в їх зоні (відблиск, який не має спільних точок із вершинами трикутників, або розташований усередині трикутника, не буде сформовано);

– на межах двох трикутників проявляються смуги Маха [10], які пов’язані з літеральним гальмуванням на сітківці ока;

– має місце зміна інтенсивності кольору зображення від кадру до кадру, що виражається в миготінні, особливо відблисків, оскільки при формуванні динамічних зображень змінюється структура та положення вузлів тріангуляційної мережі;

– наявність артефакту типу “зірка”, який полягає у тому, що відблиск, який повинен мати форму еліпса, має форму зірки. Це пояснюється тим, що ділянки відблиску формуються в різних трикутниках і проявляється ефект смуг Маха;

– метод не враховує перспективу об’єкта.

Для того, щоб покращити якість зафарбовування Гуро збільшують щільність тріангуляційної сітки, що забезпечує більш високий ступінь деталізації об’єкта, однак це призводить до збільшення навантаження.

Метод Гуро швидший ніж метод Фонга, але використовуючи його можна одержати відблиски шляхом поділу існуючих полігонів на більш дрібні. Такий метод часто використовують у програмах, де важлива швидкість. У методах Фонга та Гуро спочатку апроксимуються вектори N до поверхонь в вершинах багатокутників, а потім інтерполюється значення вектора N до поверхні всередині багатокутників. Отримані значення вектора N застосовуються для знаходження інтенсивності пікселя. Використання моделі дзеркального

відбитку краще, порівнюючи з інтерполяцією інтенсивності, адже при цьому спостерігається краще відтворення світлових відблисків. У випадках, коли дзеркальний відбиток не застосовується, інтерполяція векторів N дозволяє отримати якісніші результати, ніж інтерполяція інтенсивності, адже апроксимація N в такому випадку здійснюється у кожній точці і формується більш реалістичне зображення.

У зафарбовуванні по Фонгу вектор N до поверхні лінійно інтерполюється на багатокутник по нормалям в вершинах багатокутника, а у зафарбовуванні по Гуро інтерполюються кольори через полігони. Для того, щоб отримати кінцевий колір пікселя в моделі Фонга, необхідно інтерполювати і нормалізувати N поверхні у кожному пікселі. Зафарбування по Фонгу затрачає більше обчислювальних ресурсів, ніж зафарбування Гуро, адже модель відбиття розраховується не лише в вершинах, а в кожному пікселі. Інтерполяція для векторів N у методі Фонга аналогічна до визначення інтенсивності у методі Гуро.

При зафарбовуванні за методом Фонга інтенсивність кольору пікселів розраховують за формулою (1.12):

$$I = I_a \cdot k_a + I_l\left(k_d \cdot \vec{N} \cdot \vec{L} + k_s \cdot \left(\vec{N} \cdot \vec{H}\right)^n\right), \qquad (1.12)$$

де k_a, k_d, k_s – коефіцієнти розсіяного, спекулярного та дифузного відбиття, I_a та I_l – інтенсивності розсіяного і направленого джерел світла.

Основним недоліком метода Фонга є потреба великих обчислювальних витрат, тому що для кожного пікселя, за відомим вектором N, обчислюється значення I. Використовуючи карту відбитків, можна зменшити кількість обчислень. Така карта є набором попередньо розрахованих інтенсивностей для визначеного діапазону значень вектора N. Тому достатньо обчислити карту відбитків лише один раз, а потім, за відомим вектором N, отримувати з неї значення інтенсивності, не перераховуючи їх.

Метод Фонга потребує великих обчислювальних витрат, проте якість зображення краща. Саме тому для формування складних сцен у реальному часі зазвичай використовують метод Гуро.

На рисунку 1.12 зображено приклад зафарбовування за методом Фонга, а на рисунку 1.13 показано порівняння об’єктів, зафарбованих з використанням методу Фонга і методу Гуро.

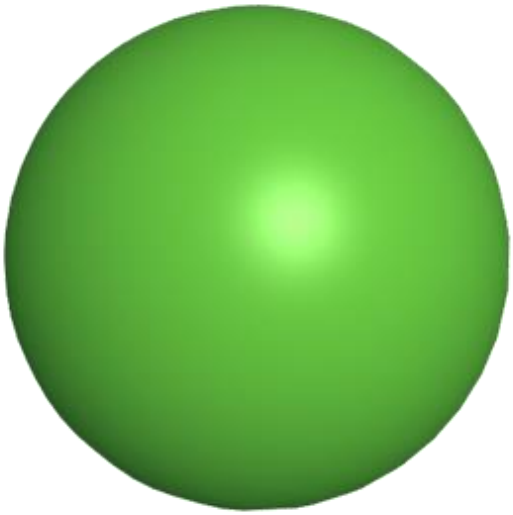


Рисунок 1.12 – Сфера із зафарбовуванням Фонга

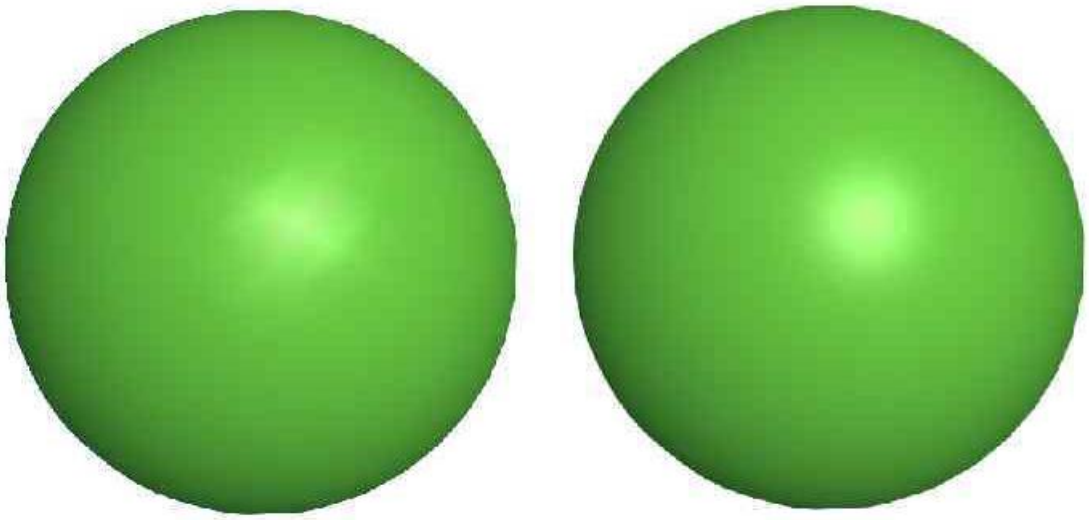


Рисунок 1.13 – Сфера зафарбована методом Гуро (ліворуч), методом Фонга (праворуч)

1.4 Висновки

У даному розділі проаналізовано предметну галузь та визначено основні тенденції розвитку комп’ютерної графіки на сьогоднішній день, також було розглянуто основні стадії графічного конвеєра.

Найпоширенішими у комп’ютерній графіці є моделі Бліна і Фонга, які дають можливість формувати тривимірні графічні зображення і потребують відносно невеликих обчислювальних витрат. Моделі Торренса-Спарроу, Кука-Торренса, Орена-Найяра та Шліка є високореалістичними, проте їх застосовують лише у випадках, коли потрібно побудувати фотореалістичні зображення. Тому в магістерській кваліфікаційній роботі для аналізу було обрано моделі відбиття Фонга, Бліна та Ламберта.

Проаналізовано існуючі методи та засоби для зафарбовування тривимірних графічних зображень. Визначено, що деталізація та реалістичність побудованих тривимірних об’єктів підвищується. Виявлено, що у зв’язку із збільшенням геометричної складності графічних зображень потрібна розробка нових методів та засобів для формування тривимірних об’єктів.

2 МЕТОДИ ПРИСКОРЕНОГО ВИЗНАЧЕННЯ НОРМАЛІЗОВАНИХ ВЕКТОРІВ ДЛЯ ЗАДАЧ РЕНДЕРИНГУ

2.1 Метод спрощення визначення векторів для задач рендерингу

Розрахунок моделей освітлення [1, 11] передбачає нормалізацію векторів, що вимагає достатньо великих обчислювальних витрат. Ураховуючи, що дистрибутивна функція й нормалізовані вектори [1, 11] обчислюються для кожної точки поверхні, то продуктивність формування графічних зображень у значній мірі визначається саме реалізацією зазначених процедур. Тому актуальною задачею є розробка методів спрощеного розрахунку векторів для задач рендерингу.

Нормалізація векторів вимагає великих обчислювальних затрат. Для спрощення розрахунків використаємо квадратичну інтерполяцію векторів. При цьому будемо вважати, що апріорно задані нормалізовані вектори в початковій та кінцевій точці рядка растеризації (РР). Знайдемо проміжні вектори за формулою (2.1):

$$\vec{N}_{i,t} = \vec{G}_i \cdot t^2 + \vec{P}_i \cdot t + \vec{Q}_i . \qquad (2.1)$$

Введемо позначення: $\vec{N}_{i,l}$, $\vec{N}_{i,p}$, $\vec{N}_{i,c}$ – відповідно вектори у початковій, кінцевій та середній точках РР трикутника. Введемо параметричну зміну t.

При нульовому значенні $t(t=0)$ $\vec{N}_{i,l} = \vec{Q}_i$. В кінцевій точці РР растеризації $t = 1$, тому $\vec{N}_{i,p} = \vec{G}_i + \vec{P}_i + \vec{Q}_i$.

В середній точці РР $t = 1/2$, то $N_{i,c} = \frac{\vec{G}_i}{4} + \frac{\vec{P}_i}{2} + \vec{Q}_i$.

Отримуємо таку систему рівнянь:

$$\begin{cases} \vec{N}_{i,l} = \vec{Q}_i, \\ \vec{N}_{i,p} = \vec{G}_i + \vec{P}_i + \vec{Q}_i, \\ N_{i,c} = \dfrac{\vec{G}_i}{4} + \dfrac{\vec{P}_i}{2} + \vec{Q}_i, \end{cases} \quad (2.2)$$

Розв'язком такої системи є:

$$\vec{G}_i = 2 \cdot \vec{N}_{i,p} - 4 \cdot \vec{N}_{i,c} + 2 \cdot \vec{N}_{i,l}, \quad \vec{P}_i = 4 \cdot \vec{N}_{i,c} - \vec{N}_{i,p} - 3 \cdot \vec{N}_{i,l}, \quad \vec{Q}_i = \vec{N}_{i,l}. \quad (2.3)$$

Нехай $h_i = 1/\sqrt{2(1+\vec{N}_{i,l} \cdot \vec{N}_{i,p})}$. Тоді:

$$\vec{N}_{i,c} = h_i(\vec{N}_{i,l} + \vec{N}_{i,p}), \quad (2.4)$$

$$\vec{G}_i = 2(1-2 \cdot h) \cdot (\vec{N}_{i,l} + \vec{N}_{i,p}), \quad \vec{P}_i = (4h-3) \cdot (\vec{N}_{i,l} + \vec{N}_{i,p}) + 2 \cdot \vec{N}_{i,p}, \quad \vec{Q}_i = \vec{N}_{i,l}.$$

З останніх формул визначаємо: $\vec{P}_i = (\vec{N}_{i,p} - \vec{N}_{i,l}) - \vec{G}_i$.

Нехай $r = \vec{N}_{i,l} \cdot \vec{N}_{i,p} = \cos\psi_i$. ψ_i - кут між векторами $\vec{N}_{i,l}$ та $\vec{N}_{i,p}$. Апроксимуємо вираз $1/\sqrt{2(1+r)}$ поліномом Чебішева другої степені: Отримуємо $1/\sqrt{2(1+r)} \approx 0{,}103r^2 - 0{,}306r + 0{,}705$.

Аналіз показав, що Δ_{max} не перевищує 0,002. На рис. 2.1 зображено графік відносної похибки апроксимації.

Отриманий вираз дозволяє зменшити час розрахунку вектора в середній точці РР в понад 2,5 рази.

Зрозуміло, що збільшити точність визначення $1/\sqrt{2(1+r)}$ можна за рахунок використання поліномів Чебішева вищого порядку. Зокрема, для третьої степені $1/\sqrt{2(1+r)} \approx -0{,}059r^3 + 0{,}193r^2 - 0{,}341r + 0{,}707$. В цьому випадку $\Delta_{max} \le 0{,}00038$.

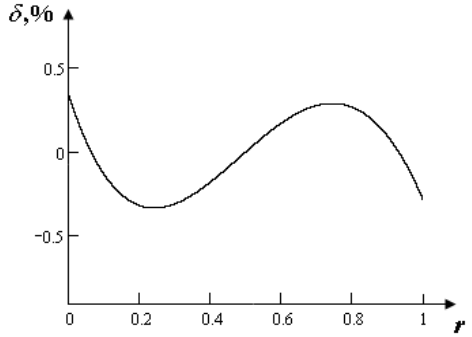


Рисунок 2.1. – δ апроксимації виразу $\frac{1}{\sqrt{2(1+r)}}$ поліномом Чебішева

Знайдемо $|\Delta_{max}|$ використанні формули (2.1) можна визначити за виразом

$$|\Delta_{max}| \le \frac{M'''_{max} \cdot (t-t_l) \cdot (t-t_c) \cdot (t-t_p)}{6}, \qquad (2.5)$$

де t_l, t_c, t_p – значення параметричної змінної в початковій, середній і кінцевій точці РР, M''' – похідна від функції.

Визначимо похідну третьої степені для виразу $\vec{N}(t) = \vec{N}_l \frac{sin((1-t)\psi)}{sin\psi} + \vec{N}_p \frac{sin(t\psi)}{sin\psi}$.

Остання формула дає можливість знайти вектор в точці t ($t \in [0,1]$, ψ – кут між векторами $\vec{N}_l$ і $\vec{N}_p$).

$$\vec{N}'''(t) = \frac{(\vec{N}_l cos(1-t)\cdot\psi) - \vec{N}_p cos(t\cdot\psi))\cdot\psi^3}{sin\psi}. \qquad (2.6)$$

Враховуючи, що $t_l = 0, t_c = 1/2,\ t_p = 1$

$$|\Delta_{max}| \le max\left(\frac{(\vec{N}_l cos(1-t)\cdot\psi) - \vec{N}_p cos(t\cdot\psi))\cdot\psi^3}{sin\psi}\right)\frac{t\cdot\left(t-\frac{1}{2}\right)\cdot(t-1)}{6}. \qquad (2,7)$$

Вираз має два множники.

Визначимо екстремуми першого й другого множника.

Знайдемо похідні від множників і прирівняємо їх до нуля.

Максимальне по модулю значення третьої похідної дорівнює $\frac{\psi^3 \cdot \sqrt{2}}{sin\psi}$, а

виразу $\frac{t\cdot\left(t-\frac{1}{2}\right)\cdot(t-1)}{6}\ -0{,}008.$

$\frac{\psi^3\cdot\sqrt{2}}{sin\psi}$ зростає монотонно. Якщо $\psi=\pi/2$, її значення менше 5,5. Тому $|\Delta_{max}|\le 0{,}044$. Максимальна абсолютна похибка при визначені одиничного вектору буде дорівнювати:

$$\Delta_в=\left|1-\sqrt{(x+\Delta x)^2+(y+\Delta y)^2+(z+\Delta z)^2}\right|. \tag{2.8}$$

Тому

$$\Delta_в\le 1-\sqrt{(x^2+y^2+z^2)+2|\Delta_{max}|(x+y+z)} \tag{2.9}$$

$|x|+|y|+|z|)\le 1{,}5$, тому $\Delta_в\le 1-\sqrt{1+3\cdot\Delta}=0{,}064.$

Експериментальні дослідження показали, що при реалізації на програмному рівні час визначення векторів для середнього трикутника зменшився в 2,8 разів.

Нехай $t=t+\Delta t,$ тоді

$$\vec{N}_{i,t+\Delta t}=\vec{G}_i\cdot(t+\Delta t)^2+\vec{P}_i\cdot(t+\Delta t)+\vec{Q}_i=\vec{N}_{i,t}+\Delta t\cdot(\vec{G}_i(2\cdot t+\Delta t)+\vec{P}_i). \tag{2.10}$$

Введем позначення $W_{i,t}=\Delta t\cdot(\vec{G}_i(2\cdot t+\Delta t)+\vec{P}_i)$. Тоді

$$W_{i,t+\Delta t}=\Delta t\cdot(\vec{G}_i(2\cdot(t+\Delta t)+\Delta t)+\vec{P}_i)=\Delta t\cdot(G_i(2\cdot t+\Delta)+P_i)+2G_i\cdot\Delta t^2 \tag{2.11}$$

Можна записати. що

$$W_{i,t+\Delta t}=W_{i,t}+2G_i\cdot\Delta t^2. \tag{2.12}$$

$2G_i\cdot\Delta t^2$ не змінюється для РР трикутник. Тому в циклі визначення одиничних векторів використовуються мікрооперації накопичуючого додавання.

Експериментальні дослідження показали, що час визначення векторів для середнього трикутника зменшився в $3{,}6$ разів.

Розроблено метод визначення одиничних векторів, особливість якого полягає в використанні для апроксимації поліному другого ступеня.

Отримані вирази для визначення векторів нормалей, які мають порівняно з існуючими меншу обчислювальну складність.

2.2 Формування векторів нормалей із використанням сферично-кутової інтерполяції

Нехай задано вектори $\vec{N}_a$ і $\vec{N}_b$ в кінцевих точках РР. Тоді

$$\vec{N}(w) = \vec{N}_a \frac{sin((1-w)\psi)}{sin\psi} + \vec{N}_b \frac{sin(w\psi)}{sin\psi}, \quad (2.13)$$

де $w \in [0,1]$ – параметрична змінна, а ψ – кут між векторами $\vec{N}_a$ і $\vec{N}_b$ (рисунок 2.2).

Підставимо в останній вираз формулу для синуса різниці кутів.

$$\vec{N}(w) = \vec{N}_a cos(w\psi) - \vec{N}_a \frac{cos\psi \cdot sin(w \cdot \psi)}{sin\psi} + \vec{N}_b \frac{sin(w \cdot \psi)}{sin\psi}. \quad (2.14)$$

Виконавши еквівалентні перетворення, отримуємо

$$\vec{N}(w) = \vec{N}_a cos(w \cdot \psi) + \frac{\vec{N}_b - \vec{N}_a cos\psi}{sin\psi} sin(w \cdot \psi). \quad (2.15)$$

Відомо, що

$$sin\psi = \sqrt{1 - cos^2\psi} = \sqrt{1 - \vec{N}_a \cdot \vec{N}_b}, \; cos\psi = \vec{N}_a \cdot \vec{N}_b \quad (2.16)$$

Тоді

$$\vec{N}(w) = \vec{N}_a \cos(w\psi) + \frac{\vec{N}_b - \vec{N}_a(\vec{N}_b \cdot \vec{N}_a)}{\sqrt{1 - \left(\vec{N}_b \cdot \vec{N}_a\right)^2}} \sin(w\psi). \quad (2.17)$$

Доведемо, що вектор $\vec{N}_k = \frac{\vec{N}_b - \vec{N}_a(\vec{N}_b \cdot \vec{N}_a)}{\sqrt{1 - \left(\vec{N}_b \cdot \vec{N}_a\right)^2}}$ є одиничним.

Знайдемо модуль виразу $\vec{N}_b - \vec{N}_a(\vec{N}_b \cdot \vec{N}_a)$

$$\sqrt{\left(\vec{N}_b - \vec{N}_a(\vec{N}_b \cdot \vec{N}_a)\right)^2} = \sqrt{\vec{N}_b^2 - 2(\vec{N}_b \cdot \vec{N}_a)(\vec{N}_b \cdot \vec{N}_a) + (\vec{N}_b \cdot \vec{N}_a)^2 \vec{N}_a^2}. \quad (2.18)$$

$\vec{N}_a$ і $\vec{N}_b$ мають одиничну довжину, тому

$$\left|\vec{N}_k\right| = \sqrt{1-(\vec{N}_b \cdot \vec{N}_a)^2}\,. \qquad (2.19)$$

Знаменник у наведеному співвідношенні є його модулем, тому вектор $\vec{N}_k$ нормалізовано.

Знайдемо $\vec{N}_b - \vec{N}_a(\vec{N}_b \cdot \vec{N}_a) \perp \vec{N}_a$.

$$(\vec{N}_b - \vec{N}_a(\vec{N}_b \cdot \vec{N}_a)) \cdot \vec{N}_a = \vec{N}_b \cdot \vec{N}_a - \vec{N}_a^2(\vec{N}_b \cdot \vec{N}_a) = \vec{N}_b \cdot \vec{N}_a - (\vec{N}_b \cdot \vec{N}_a) = 0 \qquad (2.20)$$

тому вектор $\vec{N}_k$ перпендикулярний до $\vec{N}_a$. На рисунку 2.3 зображено взаємне розміщення векторів $\vec{N}_a, \vec{N}_b, \vec{N}_k$.

$\vec{N}_k = \dfrac{\vec{N}_b - \vec{N}_a(\vec{N}_b \cdot \vec{N}_a)}{\sqrt{1-(\vec{N}_b \cdot \vec{N}_a)^2}}$, тому формулу (2.13) можна записати так:

$$\vec{N}(w) = \vec{N}_a \cos(w \cdot \psi) + \vec{N}_k \sin(w \cdot \psi), \qquad (2.21)$$

Нехай РР має m – точок, тому кут між двома суміжними векторами дорівнює $\varphi = \psi / m$, де $\psi = \arccos(\vec{N}_a \cdot \vec{N}_b)$ (рисунок 2.2). Враховуючи останнє, вираз (2.21) прийме такий вигляд:

$$\vec{N}(t) = \vec{N}_a \cdot \cos(t \cdot \varphi) + \vec{N}_k \cdot \sin(t \cdot \varphi), \qquad (2.22)$$

де t – позиція пікселя в РР, $t \in [0, l]$.

Знайдемо $\vec{N}(t+1)$ і $\vec{N}(t-1)$.

$$\vec{N}(t+1) = \vec{N}_a \cdot \cos((t+1) \cdot \varphi) + \vec{N}_k \cdot \sin((t+1) \cdot \varphi) =$$

$$\vec{N}_a \cdot \cos(t \cdot \varphi) \cdot \cos\varphi - \vec{N}_a \cdot \sin(t \cdot \varphi) \cdot \sin\varphi + \vec{N}_k \cdot \sin(t \cdot \varphi) \cos\cdot \varphi + \vec{N}_k \cdot \cos(t \cdot \varphi) \cdot \sin\varphi.$$

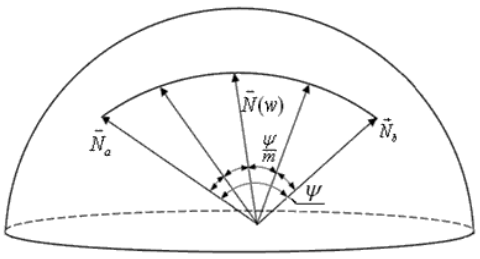


Рисунок 2.2 – Сферично-кутова інтерполяція векторів нормалей

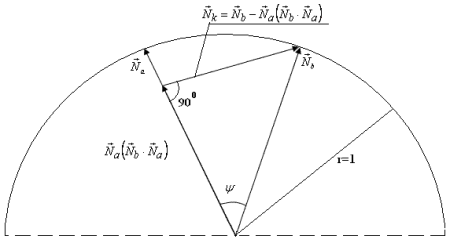


Рисунок 2.3 – Визначення вектору $\vec{N}_k$

$$\vec{N}(t-1) = \vec{N}_a \cdot cos((t-1)\cdot\varphi) + \vec{N}_k \cdot sin((t-1)\cdot\varphi) =$$
$$\vec{N}_a \cdot cos(t\cdot\varphi)\cdot cos\varphi + \vec{N}_a \cdot sin(t\cdot\varphi)\cdot sin\varphi + \vec{N}_k\, sin(t\cdot\varphi)\cdot cos\varphi - \vec{N}_k \cdot cos(t\cdot\varphi)\cdot sin\varphi.$$

Зайдемо $\vec{N}(t+1) + \vec{N}(t-1)$ і $\vec{N}(t-1)$.

$$\vec{N}(t+1) + \vec{N}(t-1) = 2cos\varphi \cdot \left(\vec{N}_a \cdot cos(t\cdot\varphi) + \vec{N}_k \cdot sin(t\cdot\varphi)\right). \quad (2.23)$$

Враховуючи (2.22), знаходимо

$$\vec{N}(t+1) + \vec{N}(t-1) = 2\vec{N}(t)\cdot cos\varphi. \quad (2.24)$$

Тому

$$\vec{N}(t+1) = 2\vec{N}(t)\cdot cos\varphi - \vec{N}(t-1). \quad (2.25)$$

З наведеного виразу можна зробити висновок, що нормалізований вектор при сферично-кутовій інтерполяції можна визначити через два попередні.

Дослідження показали, що при програмній реалізації час визначення векторів нормалей середнього трикутника зменшився в 1.7 разів.

Знайдемо формули для $\vec{N}(t+2)$, $\vec{N}(t+3)$, $\vec{N}(t+4)$.

$$\vec{N}(t+2) = 2\vec{N}(t+1)\cdot cos\varphi - \vec{N}(t).$$

Підставимо вираз $\vec{N}(t+1)$ з формули (2.25). Отримуємо

$$\vec{N}(t+2) = 2\cdot\left(2\vec{N}(t)\cdot cos\varphi - \vec{N}(t-1)\right)\cdot cos\varphi - \vec{N}(t) =$$
$$= 4\vec{N}(t)cos^2\varphi - 2\vec{N}(t-1)\cdot cos\varphi - \vec{N}(t)$$

Виконаємо спрощення

$$\vec{N}(t+2) = \vec{N}(t)\cdot\left(4cos^2\varphi - 1\right) - 2\vec{N}(t-1)\cdot cos\varphi. \quad (2.26)$$

Оскільки

$$4\cos^2\varphi - 1 = 4\left(\frac{1}{2} + \frac{1}{2}\cdot\cos 2\varphi\right) - 1 = 1 + 2\cos 2\varphi,$$

то

$$\vec{N}(t+2) = \vec{N}(t)\cdot\left(2\cos 2\varphi + 1\right) - 2\vec{N}(t-1)\cos\varphi. \qquad (2.27)$$

Формула (2.7) має меншу обчислювальну складність, тому зменшується кількість “довгих” операцій.

$$\vec{N}(t+3) = \cos 2\varphi\cdot\left(4\vec{N}(t)\cdot\cos\varphi - \vec{N}(t-1)\right) - 3\vec{N}(t-1). \qquad (2.28)$$

$$\vec{N}(t+4) = 2\vec{N}(t)\cdot\cos 2\varphi\left(2\cos 2\varphi + 3\right) - 2\vec{N}(t-1)\cos\varphi\left(\cos 2\varphi + 4\right) + \vec{N}(t). \qquad (2.29)$$

У формулах (2.27), (2.28) і (2.29) використовуються однакові вектори. Це дає можливість визначити вектори для чотирьох точок РР.

Розрахунок векторів для трикутника доцільно розбити на такі етапи:

1) розраховуються вектори до точок сторін трикутника і $\vec{N}_k$, $\sin\varphi$, $\cos\varphi$, які в подальшому використовуються для розрахунку нормалей в РР;

2) етап розрахунку векторів у РР трикутника.

Розкладемо $cos(\psi)$ і $cos(\psi/m)$ у ряд Тейлора:

$$cos(\psi) \approx 1 - \frac{1}{2}\cdot\psi^2 + \frac{1}{24}\cdot\psi^4 + O(\psi^6). \qquad (2.30)$$

$$cos(\frac{\psi}{m}) \approx 1 - \frac{1}{2\cdot m^2}\cdot\psi^2 + \frac{1}{24\cdot m^4}\cdot\psi^4 + O(\psi^6). \qquad (2.31)$$

Використаємо два члена розкладу.

Останній вираз напишемо так:

$$cos(\frac{\psi}{m}) \approx 1 - \frac{1}{2\cdot m^2}\cdot\psi^2 = \frac{1}{m^2}(m^2 + 1 - 1 - \frac{1}{2}\cdot\psi^2) = \frac{1}{m^2}(m^2 - 1 + cos\psi) =$$
$$= \frac{1}{m^2}(m^2 - 1 + cos\psi) = \frac{cos\psi - 1}{m^2} + 1.$$

Таким чином:

$$cos(\frac{\psi}{m}) \approx \frac{cos\psi - 1}{m^2} + 1. \qquad (2.32)$$

Розкладемо $cos(\varphi)$ і $cos(\frac{\varphi}{m})$ у ряд Тейлора:

$$cos(\varphi) \approx 1 - \frac{1}{2} \cdot \varphi^2 + \frac{1}{24} \cdot \varphi^4 + O(\varphi^6). \quad (2.33)$$

$$cos(\frac{\varphi}{m}) \approx 1 - \frac{1}{2 \cdot m^2} \cdot \varphi^2 + \frac{1}{24 \cdot m^4} \cdot \varphi^4 + O(\varphi^6). \quad (2.34)$$

Використаємо два члена. Отримаємо:

$$cos(\frac{\varphi}{m}) \approx 1 - \frac{1}{2 \cdot m^2} \cdot \varphi^2 = \frac{1}{m^2}(m^2 + 1 - 1 - \frac{1}{2} \cdot \varphi^2) = \frac{1}{m^2}(m^2 - 1 + cos\varphi) =$$
$$= \frac{1}{m^2}(m^2 - 1 + cos\varphi) = \frac{cos\varphi - 1}{m^2} + 1.$$

Остаточно можемо записати:

$$cos(\frac{\varphi}{m}) \approx \frac{cos\varphi - 1}{m^2} + 1. \quad (2.35)$$

На рисунку 2.4 зображено графік абсолютної похибки Δ апроксимації від кута φ і довжини РР.

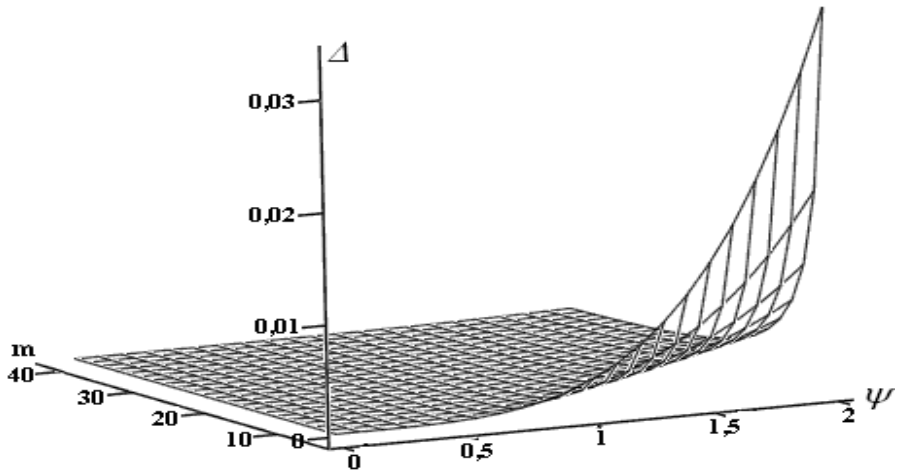


Рисунок 2.4 – Графік зміни абсолютної похибки від ψ і m

Із графіка видно, що для більшої площі поверхні $\Delta \to 0$. Виключення складає частина поверхні, для якої $\varphi > \pi / 2$, де похибка хоча і зростає, але має досить мале значення. Слід відмітити, що останній випадок малоймовірний і, як правило, виключається ще на стадії теселяції об'єкта.

З урахуванням отриманих результатів формулу (2.21) запишемо у вигляді

$$\vec{N}(t+1)=2\vec{N}(t)\cdot(\frac{cos\psi-1}{m^2}+1)-\vec{N}(t-1). \qquad (2.36)$$

Наведений вираз дозволяє виключити з обчислювального процесу розрахунок операції арккосинуса. Множник

$$\frac{cos\psi-1}{m^2}+1=\frac{\vec{N}_l\cdot\vec{N}_p-1}{m^2}+1 \qquad (2.37)$$

розраховується один раз на РР трикутника по значенням векторів $\vec{N}_l, \vec{N}_p$.

2.3 Висновки

1. Розроблено метод визначення одиничних векторів, особливість якого полягає в використання для апроксимації поліному другого ступеня.

Отримані вирази для визначення векторів нормалей мають порівняно з існуючими меншу обчислювальну складність.

2. Розроблено метод визначення векторів у рядку растеризації з використанням рекурентних співвідношень, що дозволяє суттєво спростити нормалізацію векторів.

3 РОЗРОБКА МЕТОДІВ ПРИСКОРЕНОГО ЗАФАРБОВУВАННЯ ТРИВИМІРНИХ ОБ'ЄКТІВ

3.1. Підвищення продуктивності рендерингу Гуро

Нехай задано трикутник ABC. Для задач рендерингу у вершинах трикутника ABC задаються вектори нормалей $\vec{N}_A$, $\vec{N}_B$, $\vec{N}_C$. Для формування відблисків на поверхні знаходять вектор $\vec{H} = (\vec{L} + \vec{V}) / \left|\vec{L} + \vec{V}\right|$. Вважається, що цей вектор є постійним для трикутника.

Встановимо граничне значення q на спекулярну складову кольору, розрахунок якої недоцільний, оскільки світлова пляма непомітна.

За умови, що

$$\left|\vec{N}_A \cdot \vec{H} - \vec{N}_B \cdot \vec{H}\right| \le q,\ \left|\vec{N}_A \cdot \vec{H} - \vec{N}_C \cdot \vec{H}\right| \le q,\ \left|\vec{N}_B \cdot \vec{H} - \vec{N}_C \cdot \vec{H}\right| \le q, \quad (3.1)$$

скалярну складовою кольору не розраховують.

Вектори $\vec{N}_A$, $\vec{N}_B$, $\vec{N}_C$, $\vec{H}$ нормалізовані, тому при

$$\left|\cos\gamma - \cos\beta\right| \le q, \left|\cos\gamma - \cos\lambda\right| \le q, \left|\cos\lambda - \cos\beta\right| \le q \quad (3.2)$$

розрахунком визначення відблисків можна знехтувати. За умови, що хоча б одна із нерівностей не виконується, то трикутник містить відблиск. Цей відблиск перетинає відповідну сторону трикутника.

Визначимо максимальну інтенсивність спекулярної складової кольору на вершинах трикутника (рисунок 3.1).

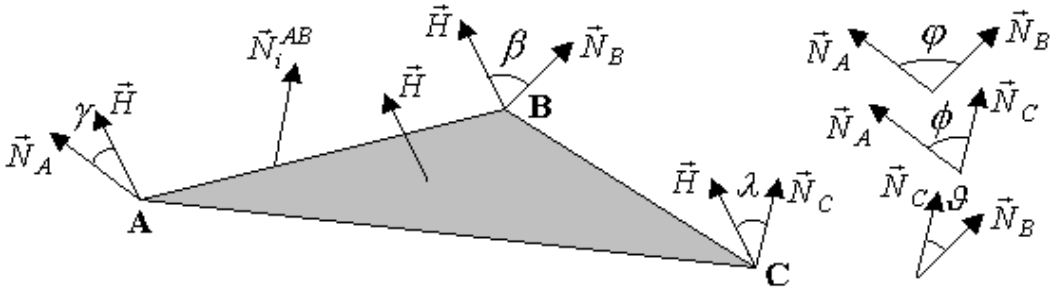


Рисунок 3.1 – Вектори нормалей трикутника ABC

Запишемо рівняння для визначення векторів до точок ребра трикутника

$\vec{N}_i^{AB} = \vec{N}_A + t_1(\vec{N}_B - \vec{N}_A)$.

Для знаходження одиничного вектора $\vec{N}_i^{AB}$.

$$\frac{\vec{N}_i^{AB}}{\left|\vec{N}_i^{AB}\right|} = \frac{\vec{N}_A + t_1 \cdot (\vec{N}_B - \vec{N}_A)}{\sqrt{(\vec{N}_A)^2 + 2t_1 \cdot \vec{N}_A(\vec{N}_B - \vec{N}_A) + t_1{}^2(\vec{N}_A - \vec{N}_A)^2}}. \quad (3.3)$$

Оскільки $\vec{N}_A, \vec{N}_B$ – нормалізовані, то $\vec{N}_A^2 = \vec{N}_B^2 = 1$. Оскільки $\vec{N}_A \cdot \vec{N}_B = \cos\varphi$, то

$$\frac{\vec{N}_i^{AB}}{\left|\vec{N}_i^{AB}\right|} = \frac{\vec{N}_A + t_1 \cdot (\vec{N}_B - \vec{N}_A)}{\sqrt{2t_1{}^2(1-\cos\varphi) - 2t_1(1-\cos\varphi) + 1}}. \quad (3.4)$$

Знайдемо на сторонах трикутника ABC точки, де спекулярна складова кольору має максимальне значення.

Для сторони AB

$$\frac{\vec{N}_i^{AB} \cdot \vec{H}}{\left|\vec{N}_i^{AB}\right|} = \frac{(\vec{N}_A + t_1 \cdot (\vec{N}_B - \vec{N}_A)) \cdot \vec{H}}{\sqrt{2t_1{}^2(1-\cos\varphi) - 2t_1(1-\cos\phi) + 1}} = \frac{\cos\gamma + t_1(\cos\beta - \cos\gamma)}{\sqrt{2t_1{}^2(1-\cos\varphi) - 2t_1(1-\cos\phi) + 1}}, \quad (3.5)$$

γ, β відповідно кут між вектором $\vec{H}$ і векторами $\vec{N}_A$, $\vec{N}_B$.

Знайдемо t_1, при якому скалярна складова кольору на стороні AB приймає максимально можливе значення. Для знаходження екстремальної точки візьмемо похідну від наведеного виразу і прирівняємо її до нуля.

$$\left(\frac{\vec{N}_i^{AB} \cdot \vec{H}}{\left|\vec{N}_i^{AB}\right|}\right)' = \frac{t_1(1-\cos\varphi) \cdot (\cos\beta - \cos\gamma) - \cos\gamma \cdot \cos\kappa + \cos\beta}{\sqrt{2t_1{}^2(1-\cos\varphi) - 2t_1(1-\cos\phi) + 1}} = 0. \quad (3.6)$$

Коренем рівняння (3.6) є

$$t_1 = \frac{\cos\gamma\cos\phi - \cos\beta}{(\cos\phi - 1)(\cos\beta + \cos\gamma)}. \quad (3.7)$$

Аналогічно визначимо t_2, t_3 відповідно для сторін $\grave{A}\tilde{N}$ і $\hat{A}\tilde{N}$. У цих точках спекулярна складова кольору має максимальне значення

$$t_{2=}\frac{\cos\gamma\cos\varphi - \cos\lambda}{(\cos\varphi - 1)(\cos\lambda + \cos\gamma)},\ t_3 = \frac{\cos\beta\cos\vartheta - \cos\lambda}{(\cos\vartheta - 1)(\cos\lambda + \cos\beta)}.$$

Програмно параметри t_1, t_2, t_3 розраховуються наступним чином:

$$double\ cos\_1 = idx3d\_Vector.angle(vA, h);//cos(Na\ H)$$
$$double\ cos\_2 = idx3d\_Vector.angle(vB, h);//cos(Nb\ H)$$
$$double\ cos\_3 = idx3d\_Vector.angle(vC, h);//cos(Nc\ H)$$

$$double\ cos\_4 = idx3d\_Vector.angle(vA, vB);//cos(Na\ Nb)$$
$$double\ cos\_5 = idx3d\_Vector.angle(vA, vC);//cos(Na\ Nc)$$
$$double\ cos\_6 = idx3d\_Vector.angle(vC, vB);//cos(Nc\ Nb)$$

$$double\ t1 = (cos\_1 * cos\_4 - cos\_2) / ((cos\_4 - 1) * (cos\_2 + cos\_1));$$
$$double\ t2 = (cos\_1 * cos\_5 - cos\_3) / ((cos\_5 - 1) * (cos\_3 + cos\_1));$$
$$double\ t3 = (cos\_2 * cos\_6 - cos\_3) / ((cos\_6 - 1) * (cos\_3 + cos\_2));$$

По знайденим значення параметричних змінних t можна знайти точки на сторонах трикутника, де може мати місце перетин з відблиском.

Так, наприклад, для сторони AB

$$x = \lceil x_A + t(x_B - x_A) \rceil, y = \lfloor y_A + t(y_B - y_{A)} \rfloor \qquad (3.8)$$

$$X1 = Math.abs(p1.x + t1 * (p2.x - p1.x));$$
$$X2 = Math.abs(p1.x + t2 * (p3.x - p1.x));$$
$$X3 = Math.abs(p2.x + t3 * (p3.x - p2.x));$$
$$Y1 = Math.abs(p1.y + t1 * (p2.y - p1.y));$$
$$Y2 = Math.abs(p1.y + t2 * (p3.y - p1.y));$$
$$Y3 = Math.abs(p2.y + t3 * (p3.y - p2.y));$$

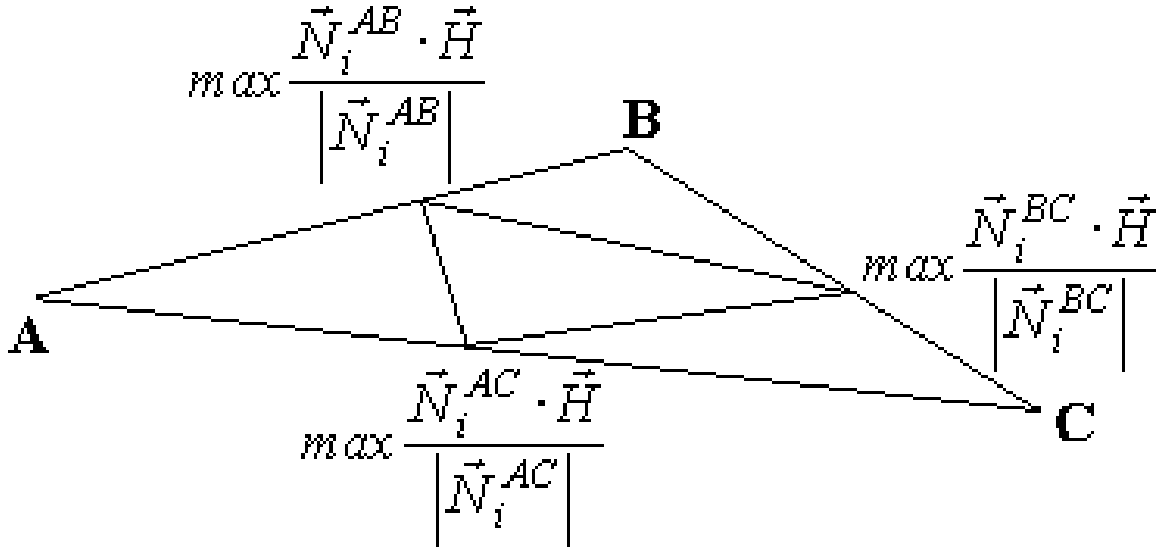


Рисунок 3.2 – Тріангуляція вихідного трикутника

Вектор нормалі в точці (x,y) знаходимо за формулою

$$N_i = \frac{\vec{N}_A(x_B - x) + \vec{N}_B(x - x_A)}{\sqrt{(x_B - x)^2 + (x - x_A)^2 + 2\cos\varphi \cdot (x_B - x)(x - x_A)}}. \quad (3.9)$$

Інтегральну інтенсивність можна знайти згідно формули

$$I = I_a k_a + I_l (k_d \vec{N}_i \cdot \vec{L} + k_s \left(\vec{N}_i \cdot \vec{H}\right)^n). \quad (3.10)$$

На рисунку 3.3 наведено блок-схему алгоритму.

Згідно з алгоритмом виконуються такі дії:

1. Розраховуються $\cos\lambda$, $\cos\beta$, $\cos y$ та аналізуються сторони трикутника на наявність спекулярної складової кольору. Для цього використовується розроблена система нерівностей (3.1).

2. На сторонах трикутника знаходяться точки, де має місце максимальна спекулярна складова кольору.

3. Визначаються значення спекулярної складової кольору в точках, що були знайдені на попередньому етапі.

4. Визначається, чи має місце факт перевищення граничного значення спекулярної складової кольору.

5. Виконується тріангуляція вихідного трикутника.

6. Рендеринг складових трикутників за методом Гуро.

Існує кілька варіантів розбиття трикутника на складові. Тип тріангуляції залежить від факту перетину сторони трикутника світловою плямою. Якщо перетин відблиску всіма сторонами трикутника і інтенсивності спекулярних склалових більші за порогове, то трикутник розбивається на чотири складових (рис. 3.4, а).

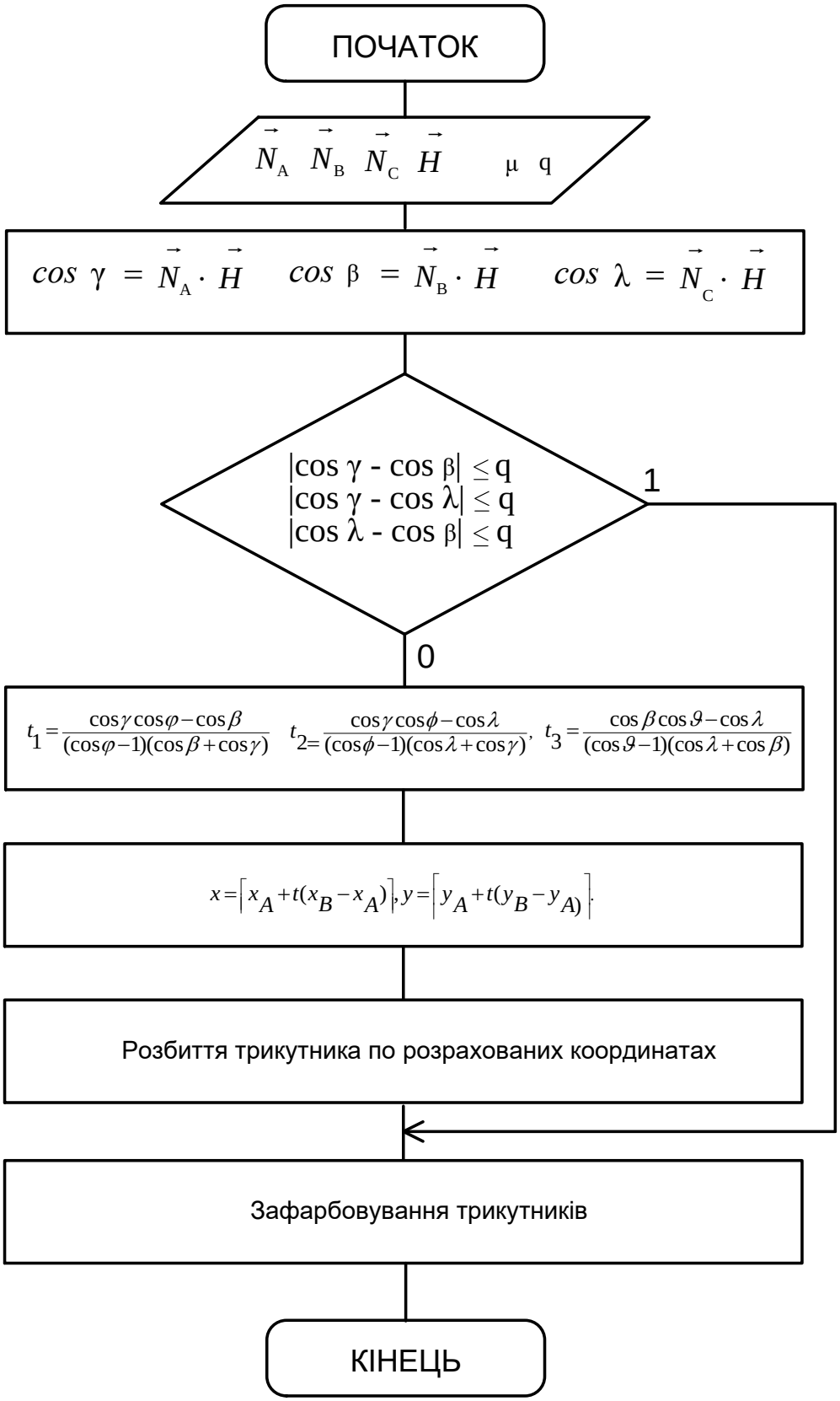


Рисунок 3.3 – Блок -схема алгоритму для покращення зафарбовування об'єктів за методом Гуро

Якщо тільки сторона перетинає ребро, то трикутник розбивається на дві складових (рис. 3.4, б). Якщо значення максимальних інтенсивностей відблиску більші за q у двох точках, то має місце розбиття, яке зображено на рис. 3.4. Трикутник може бути розбитий на три складові так, як відображено на рис. 3.4, с, д.

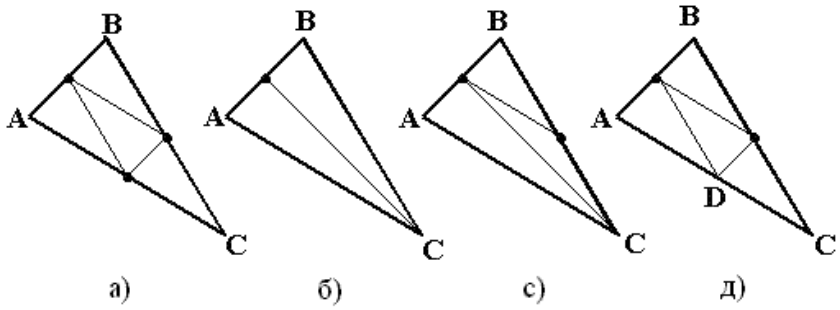


Рис. 3.4. – Розбиття трикутника на складові

На рис. 3.5 наведено приклад формування графічного об’єкту з використанням запропонованого методу.

Рисунок 3.5 – Приклад формування графічного обєкту

Запропонований метод дозволяє підвищити реалістичність формування графічних об’єктів за методом Гуро.

3.2. Метод ідентифікації відблиску на ділянці поверхні, обмеженої трикутником

Значення спекулярної складової залежить від розміщення векторів $\vec{N}$ та $\vec{H}$, де $\vec{H}$ є нормаллю площини трикутника, а $\vec{N}$ – вектор нормалі вершини. Розглянемо три можливі випадки розміщення векторів при зафарбовуванні поверхні, яка обмежена трикутником (рисунок 3.6). У першому та другому випадку вектор $\vec{H}$ розташовано поза межами трикутника, який утворено векторами $\vec{N}_A$, $\vec{N}_B$, $\vec{N}_C$, початкові точки яких зміщено в одній точці.

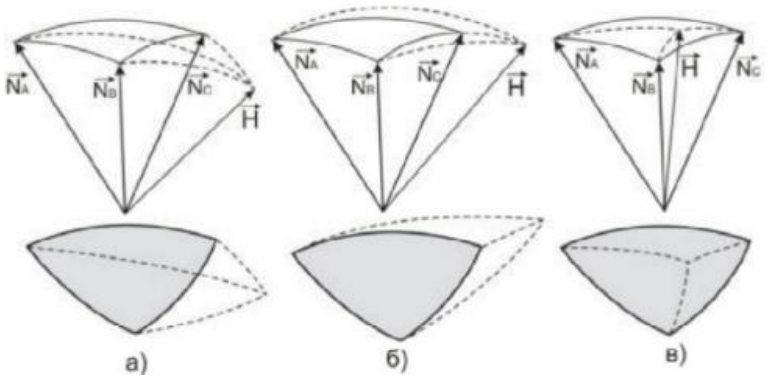


Рисунок 3.6 – Можливі випадки розміщення векторів нормалей

Розглянемо деякі ситуації для того, щоб визначити тип взаємного розташування векторів нормалей. Відомо, що світлова пляма розміщена в межах трикутника, якщо вірна одна з таких умов:

$$\begin{aligned} &\text{а) } \cos\upsilon > 0,\ \cos\vartheta < 0,\ \vec{N}_1 \cdot \vec{H} \geq q; \\ &\text{б) } \cos\upsilon < 0,\ \cos\vartheta > 0,\ \vec{N}_2 \cdot \vec{H} \geq q; \\ &\text{в) } \cos\upsilon > 0,\ \cos\vartheta > 0,\ \left(\left(\vec{N}_1 \times \vec{N}_2\right) \times \vec{H}\right) / \left(\vec{N}_1 \times \vec{N}_2\right) \geq q, \end{aligned} \tag{3.11}$$

де $\vec{N}_1$, $\vec{N}_2$ – вектори нормалей до вершин сторони, q – порогове значення для інтенсивності кольору, що визначається значеннями косинусів кута між $\vec{N}$ і $\vec{H}$,

$$\cos\upsilon = \frac{\left(\vec{N}_1 - \vec{N}_2\right)\cdot\left(\vec{H} - \vec{N}_2\right)}{\left|\vec{N}_1 - \vec{N}_2\right|\left|\vec{H} - \vec{N}_2\right|},\ \cos\vartheta = \frac{\left(\vec{N}_2 - \vec{N}_1\right)\cdot\left(\vec{H} - \vec{N}_1\right)}{\left|\vec{N}_2 - \vec{N}_1\right|\left|\vec{H} - \vec{N}_1\right|}. \qquad (3.12)$$

Недоліки запропонованого методу:

– якщо весь відблиск розміщений у трикутнику, то його не можна ідентифікувати;

– метод потребує великих обчислювальних затрат на здійснення векторних операцій.

Розробимо метод для ідентифікації дзеркальної складової, який базується на знаходженні параметрів t_1, t_2, t_3, при яких інтенсивність кольору досягає максимального значення на сторонах трикутника.

Для ідентифікації відблиску треба визначити максимальне значення дзеркальної складової кольору на сторонах трикутника (рисунок 3.7). Обирається метод і модель освітлення відповідно до дзеркальної складової кольору.

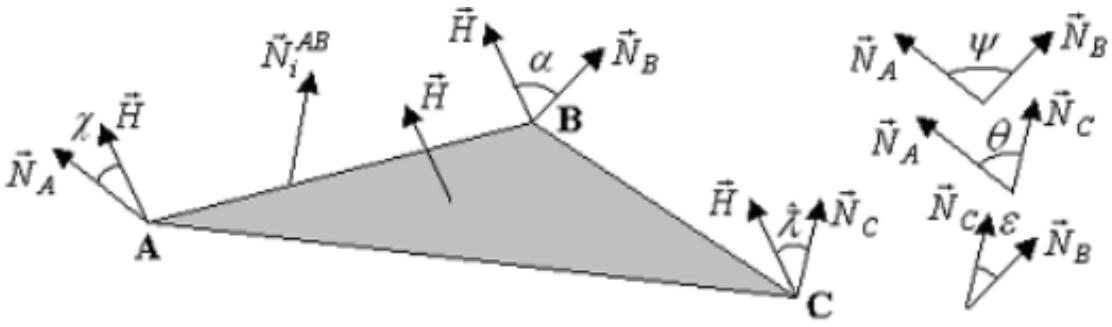


Рисунок 3.7 – Вектори нормалей трикутника ABC

$$t_1 = \frac{\cos\chi\cdot\cos\psi - \cos\alpha}{(\cos\psi - 1)(\cos\alpha + \cos\chi)}$$

$$t_2 = \frac{\cos\chi\cdot\cos\theta - \cos\lambda}{(\cos\theta - 1)(\cos\lambda + \cos\chi)} \qquad (3.13)$$

$$t_3 = \frac{\cos\alpha\cdot\cos\varepsilon - \cos\lambda}{(\cos\varepsilon - 1)(\cos\lambda + \cos\alpha)}$$

Проте даний метод не визначає можливість розміщення всього відблиску всередині трикутника.

Нормалі до точок $\vec{N}_A$, $\vec{N}_B$, $\vec{N}_C$ визначають кривизну поверхні, а $\vec{H}$ відображає розміщення джерела світла і спостерігача. Для знаходження косинусів кутів вектори нормалей нормалізуються. Кінцеві точки векторів розташовані на сфері одиничного радіуса з центром в початку координат за умови, що вектори суміщено в початок координат. Кінці нормалей точок трикутника $\vec{N}_A$, $\vec{N}_B$, $\vec{N}_C$ формують на площині побудованої сфери сферичний трикутник. Відомо, що кінцеві точки нормалей розташовані у межах побудованого сферичного трикутника.

Використаємо формулу $I = I_l k_s cos^n \gamma$. Дана формула призначена для обчислення дзеркальної складової інтенсивності кольору за моделлю Фонга. Коли вектори $\vec{N}$ і $\vec{H}$ колінеарні, спекулярна інтенсивність кольору приймає найбільше значення.

Існують такі можливі розміщення світлової плями відносно трикутника:

– за межами трикутника без перетину сторін трикутника;
– за межами трикутника з перетином сторін трикутника;
– у межах трикутника без перетину сторін трикутника;
– у межах трикутника з перетином сторін трикутника.

Вектори $\vec{N}_A$, $\vec{N}_B$, $\vec{N}_C$ і $\vec{H}$ визначаються координатами декартового простору. Проекція кінця $\vec{H}$ належить проекції сферичного трикутника на будь-яку з декартових площин, якщо вектор $\vec{H}$ розташовано між $\vec{N}_A$, $\vec{N}_B$, $\vec{N}_C$.

На рисунок 3.8 зображена фігура, обмежена дугами, перетином яких є проекції кінців векторів нормалей N'_a, N'_b, N'_c точок трикутника. Дана фігура побудована шляхом проведення проекції сферичного трикутника на декартову площину XOY.

Проаналізуємо трикутник *abc* (рисунок 3.8) для можливості спрощення розрахунків. Необхідно уникнути пропуск ідентифікації відблиску на трикутнику, але похибка не є критичною. Даний трикутник визначається прямими a, b та c, які паралельні відносно до $N'_b N'_c$, $N'_a N'_c$, $N'_a N'_b$.

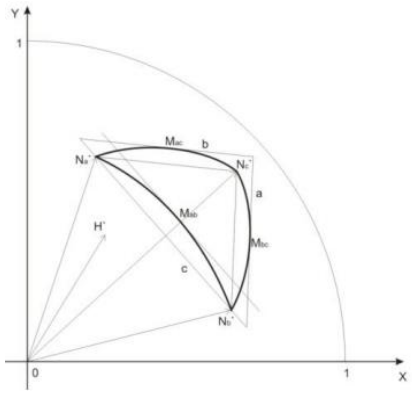


Рисунок 3.8 – Проекція сферичного трикутника

Обчисливши коефіцієнти k та b за координатами $(x_b y_b)$, $(x_c y_c)$, одержуємо рівняння прямої $N'_b N'_c$. Після підстановки відповідних координат у рівняння прямих, одержуємо систему

$$\begin{cases} y_b = k \cdot x_b + b \\ y_c = k \cdot x_c + b \end{cases} \tag{3.14}$$

Розв'язком даної системи є

$$k = \frac{y_b - y_c}{x_b - x_c},\ b = \frac{x_b y_c - x_c y_b}{x_b - x_c}, \text{ або } b = y_b - k \cdot x_b \tag{3.15}$$

Таким чином також розв'язується рівняння прямих $N'_a N'_c$, $N'_a N'_b$.

Для визначення рівняння прямої, яка розташована паралельно до заданої, необхідна лише одна точка. Коефіцієнт b можна обчислити за формулою (2), де y_b і x_b – координати будь-якої точки прямої, для якої потрібно побудувати рівняння. Відомо, що коефіцієнт k є однаковим у паралельних прямих.

Вектор $\vec{N}_{1/2}$, який розміщено між векторами $\vec{N}_a$ і $\vec{N}_b$, формує між ними кут $\psi/2$, який можна обчислити за формулою:

$$\vec{N}_{(1/2)} = \frac{\vec{N}_a + \vec{N}_b}{2\cos\frac{\psi}{2}} \tag{3.16}$$

Точка M_{ab} є проекцією кінцевої точки $\vec{N}_{1/2}$.

Тепер можемо дізнатись чи належить точка H' трикутнику abc, адже маємо рівнянням прямих $N'_b N'_c$, $N'_a N'_c$, $N'_a N'_b$ і координати M_{ac}, M_{ab}, M_{bc}.

Перевіримо такі умови – розташування точок H' і N'_a, H' і N'_b, H' і N'_c попарно в однакових напівплощинах, які відповідно розділено прямими a, b і c. При виконанні усіх трьох умов стає відомо, що точка H' розміщена у межах трикутника abc. При хибному значенні хоча б однієї з умов, визначається, що точка H' розміщена за межами abc. У такий спосіб визначається наявність центра відблиску на поверхні трикутника.

Розглянемо рисунок 3.9, на якому зображено граф-схему із узагальненим алгоритмом ідентифікації належності H' трикутнику abc.

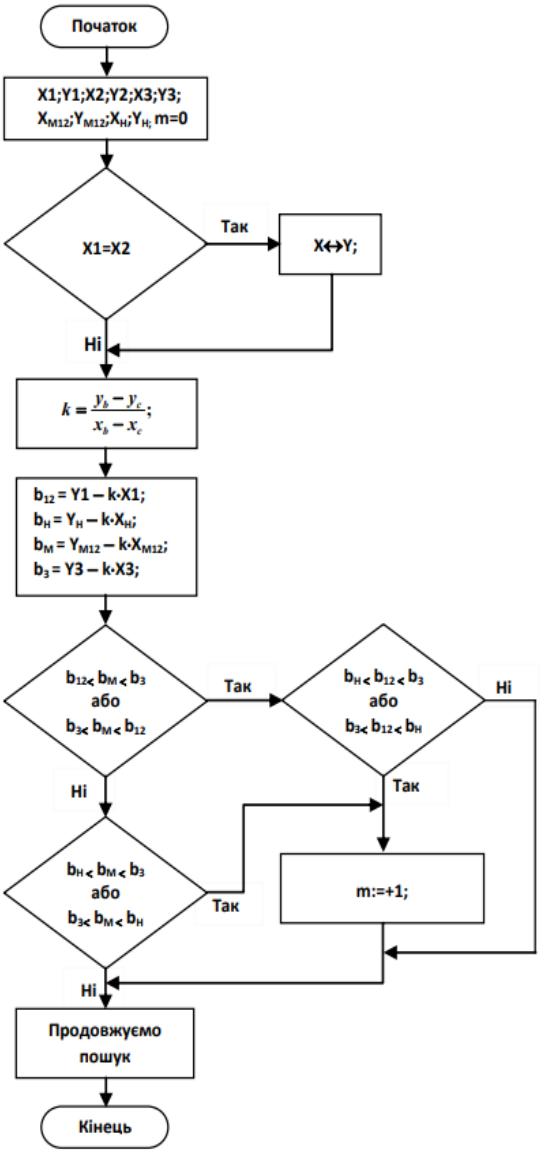


Рисунок 3.9 – Граф-схема алгоритму ідентифікації H' відносно одного з ребер трикутника abc

Даний алгоритм визначає процес визначення типу взаємного розміщення нормалей вершин трикутника $\vec{N}_A$, $\vec{N}_B$, $\vec{N}_C$ і вектора $\vec{H}$. Відповідно до алгоритму здійснюється аналіз розташування проекцій кінцевих точок векторів $\vec{N}_A$, $\vec{N}_B$, $\vec{N}_C$ і $\vec{H}$. Можна засвідчити, що розташування, яке відповідає випадку, зображеного на рисунку 3.7, є доречним, коли проаналізовано всі ребра трикутника, при $m=0$, $m=1$, $m=2$.

3.3. Особливості організації обчислювального процесу по визначенню вихідних параметрів для зафарбовування

При розробці методів і засобів зафарбовування важливо встановити такі функціональні взаємозв'язки між вихідними параметрами, які дозволять суттєво спростити реалізацію рендеригу як на програмному, так і апаратному рівнях.

Рядком растеризації трикутника (РРТ) називають проміжок між його ребрами в одному з ортогональних напрямів.

Твердження. При лінійному інтерполюванні інтенсивностей кольору вздовж ребер і рядків растеризації трикутника приріст інтенсивності кольору вздовж горизонтальних (вертикальних) рядків растеризації є постійною величиною.

Доведення. Нехай трикутник ABC задано координатами його вершин (X_A, Y_A), (X_B, Y_B), (X_C, Y_C) і відповідними їм інтенсивностями кольору: I_A, I_B, I_C. Проведемо через вершину B відрізок BK (рис. 3.10), який паралельний осі абсцис. Отримаємо два трикутники – ABK і BKC.

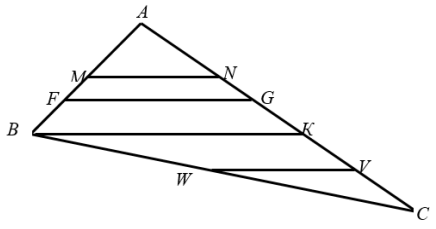


Рисунок 3.10 – Рядки растеризації трикутника

Знайдемо прирости інтенсивностей кольору для довільно вибраних рядків растеризації FG і MN :

$$\Delta I_{MN} = \frac{I_N - I_M}{MN}, \qquad \Delta I_{FG} = \frac{I_G - I_F}{FG}.$$

Виразимо інтенсивності кольору точок G і F через інтенсивності кольору вершин $A, B\ i\ C$: $I_G = \frac{I_C - I_A}{AC} \cdot AG + I_A$, $I_F = \frac{I_B - I_A}{AB} \cdot AF + I_A$.

Підставимо значення I_G, I_F у формулу для ΔI_{FG} 2.1. Отримаємо

$$\Delta I_{FG} = \frac{I_C - I_A}{AC} \cdot \frac{AG}{FG} - \frac{I_B - I_A}{AB} \cdot \frac{AF}{FG}.$$

Аналогічно для точок M і N запишемо

$$I_N = \frac{I_C - I_A}{AC} + I_M, \; I_M = \frac{I_B - I_A}{AB} \cdot AM + I_A.$$

Знайдемо приріст інтенсивності кольору вздовж рядка растеризації MN

$$\Delta I_{MN} = \frac{I_C - I_A}{AC} \cdot \frac{AN}{MN} - \frac{I_B - I_A}{AB} \cdot \frac{AM}{MN}. \qquad (3.17)$$

Трикутники AFG і AMN подібні, оскільки їх відповідні кути рівні. З подібності трикутників випливає, що

$$\frac{AG}{FG} = \frac{AN}{MN}, \qquad \frac{AF}{FG} = \frac{AM}{MN}. \qquad (3.18)$$

Ураховуючи співвідношення (3.18), можна констатувати, що праві частини виразів (3.17) і (3.18) збігаються, тобто $\Delta I_{FG} = \Delta I_{MN} = \Delta I_{Г}$.

Оскільки трикутник ABK подібний трикутнику AFG, то $\Delta I_{BÊ} = \Delta I_{FG}$.

Аналогічно можна показати, що для довільного РРТ WV трикутника BKC $\Delta I_{WV} = \Delta I_{BK}$. Оскільки трикутники ABK *і* BKC мають спільний рядок BK, то можна стверджувати, що отримана властивість має місце для усього трикутника.

Таким чином, приріст інтенсивності є постійною величиною для всіх горизонтальних рядків растеризації. Очевидно, що прирости інтенсивностей кольору мають стале значеннями й для вертикальних рядків растеризації.

Доведена властивість дозволяє суттєво зменшити обсяги обчислень при реалізації зафарбовування, оскільки приріст інтенсивності кольору

визначається для всього трикутника, а не для кожного рядка растеризації. Це дозволяє виконати ці обчислення в циклі підготування та виключити “ довгі “ операції безпосередньо із циклу зафарбовування.

У табл. додатку Г наведено кількість необхідних операцій для зафарбовування середньостатиччного трикутника з використанням доведеної властивості про сталість приросту інтенсивності. Аналіз показав, що порівняно з класичною реалізацією досягається зменшення часу зафарбовування в 1,7 раза.

Прирости інтенсивностей кольору для суміжних трикутників мають різне значення, тому через дискретний характер формування крокових траєкторій ребер полігону можливе артефактне порушення монотонності зміни інтенсивності кольору на межах сусідніх полігонів. Автором запропоновано підхід до його зменшення за рахунок використання при формуванні крокової траєкторії роздільних кроків в ортогональних напрямках і виключення центрування крокових переміщень.

Знайдемо вирази для розрахунку приростів інтенсивностей уздовж горизонтальних і вертикальних рядків растеризації. Для цього складемо систему рівнянь

$$\begin{cases} I_B = I_A + \Delta I_v \cdot \Delta Y_{BA} - \Delta I_g \cdot \Delta X_{AB}, \\ I_C = I_A + \Delta I_v \cdot \Delta Y_{CA} - \Delta I_g \cdot \Delta X_{CA}. \end{cases}$$

Розв’язавши систему рівнянь відносно I_g, I_v, отримуємо

$$\Delta I_g = \frac{\Delta X_{AB} \cdot (I_A - I_C) + \Delta X_{CA} \cdot (I_A - I_B)}{\Delta X_{AB} \cdot \Delta Y_{CA} - \Delta X_{BA} \cdot \Delta X_{CA}},\ \Delta I_v = \frac{\Delta Y_{CA} \cdot (I_B - I_A) + \Delta Y_{BA} \cdot (I_A - I_C)}{\Delta X_{AB} \cdot \Delta Y_{CA} - \Delta X_{BA} \cdot \Delta X_{CA}}.$$

Діагональний крок можна розглядати як одночасне виконання вертикального та горизонтального крокового переміщення, тому для ребер із додатними приростами координат

$$\Delta I_{dg} = \frac{(\Delta X_{AB} + \Delta Y_{BA}) \cdot (I_A - I_C) + (\Delta X_{CA} - \Delta Y_{CA}) \cdot (I_A - I_B)}{\Delta X_{AB} \cdot \Delta Y_{CA} - \Delta X_{BA} \cdot \Delta X_{CA}}.$$

В інших випадках необхідно враховувати знаки приростів координат.

Доведену властивість про сталість приросту інтенсивності можна використати й для векторів, замінивши інтенсивність кольору на координатну складову. Можна стверджувати, що прирости ортогональних координатних складових векторів нормалей уздовж горизонтальних (вертикальних) РРТ мають сталі значення. Тобто, для векторів нормалей уздовж горизонтальних (вертикальних) рядків $\Delta x = const, \Delta y = const, \Delta z = const$, де $\Delta x, \Delta y, \Delta z -$ різниці між відповідними координатними складовими двох сусідніх у рядку растеризації векторів.

Для нормалізації вектора нормалі в початковій точці рядка растеризації трикутника використовують формулу

$$\vec{N} = \frac{x}{\sqrt{x^2 + y^2 + z^2}}\vec{i} + \frac{y}{\sqrt{x^2 + y^2 + z^2}}\vec{j} + \frac{z}{\sqrt{x^2 + y^2 + z^2}}\vec{k}.$$

Для наступного за ним вектора нормалі підкореневий вираз буде мати такий вигляд

$$F_i = (x + \Delta x)^2 + (y + \Delta y)^2 + (z + \Delta z)^2 =$$
$$= (x^2 + y^2 + z^2) + 2 \cdot (x \cdot \Delta x + y \cdot \Delta y + z \cdot \Delta z) + ((\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2).$$

Уведемо позначення

$$A = x^2 + y^2 + z^2,\ B = (x \cdot \Delta x + y \cdot \Delta y + z \cdot \Delta z),\ C = (\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2.$$

Для i-того відносно першого вектора у рядку растеризації $F_i = A + 2 \cdot i \cdot B + i^2 \cdot D$. Використаємо для останнього члена метод кінцевих різниць, для чого знайдемо

$$\Delta_1 = (i+1)^2 \cdot D - i^2 \cdot D = 2 \cdot i \cdot D + D,$$
$$\Delta_2 = 2 \cdot (i+1) \cdot D + D - 2 \cdot i \cdot D - D = 2 \cdot D.$$

З урахування отриманих виразів можна записати

$F_{i+1} = F_i + 2 \cdot B + C_{i+1}$, де $C_{i+1} = C_i + 2 \cdot D$. $C_1 = D,\ F_o = A,\ C_0 = 1$.

Отримані формули для обчислення підкореневого виразу не мають операцій множення.

Доведену властивість про сталість приросту інтенсивності кольору вздовж РРТ можна використати для розпаралелення зафарбовування.

Перший метод базується на розподілі обчислювальних дій процедури зафарбовування на нижньому рівні, коли геометричному процесору задані параметри зіставних трикутників, а також інтенсивності кольору в їх вершинах. Зафарбовування області, обмеженої трикутником, проводиться з використанням тріангуляції Серпінського . За методом трикутник розбивається на чотири зіставних шляхом з'єднання середин його сторін. У результаті отримують чотири трикутники, які рівні за площею. Зіставні трикутники зафарбовують паралельно. При цьому використовується тільки один пристрій зафарбовування, який тонує один із трикутників, а результати його роботи переносяться на всі інші трикутники з певним зміщенням за координатами та відповідними трансформаціями інтенсивностей кольору точок. За рахунок розпаралелення обчислювального процесу вихідний трикутник буде зафарбовуватись в 4 рази швидше.

При зафарбовуванні виконують растеризацію верхнього трикутника і пов–торюють всі елементарні крокові прирости для інших трикутників з урахуванням того, що трикутники 1, 2, 3 мають однакові напрямки зафар-бовування, а трикутник 4 зафарбовується в зустрічному напрямку (рис. 3.11). Перед растеризацією знаходять координати вершин всіх зіставних трикутників. Аналогічно при визначенні інтенсивностей кольору внутрішніх точок верхнього (лівого) зіставного трикутника дублюють усі дії кодової інтерполяції й для інших зіставних трикутників, але вже по відношенню до їх вершин.

Оскільки тріангуляцію вихідного трикутника здійснюють шляхом ділення його ребер навпіл, то виникає задача такого розбиття трикутника на зіставні, при якому будуть відсутні точки "просікання", оскільки прирости сторін вихідного трикутника не завжди є парними. Доведено, що артефактів можна уникнути при перекритті співстаних трикутників.

Один із методів розпаралелення полягає в незалежному визначенні інтенсивностей кольору в парних і непарних точках рядка растеризації (рис. 3.12), що забезпечує підвищення продуктивності зафарбовування до двох разів. У циклі підготування знаходять інтенсивності I_1 , I_2 кольорів відповідно

першої та наступної за нею точок у рядку растеризації. Інтенсивності кольору наступних точок у РРТ знаходять за формулами:

$$I_4 = I_2 + 2\Delta I_g,\ I_6 = I_4 + 2\Delta I_g,\ ...,\ I_{2w} = I_{2w-2} + 2\Delta I_g,\ ...;$$

$$I_3 = I_1 + 2\Delta I_g,\ I_5 = I_3 + 2\Delta I_g,\ ...,\ I_{2w+1} = I_{2w-1} + 2\Delta I_g,\ ....$$

Оскільки приріст інтенсивності кольору вздовж рядка растеризації сталий, то інтенсивність кольору в його i- ій точці можна визначити за формулою $I_i = I_1 + i \cdot \Delta I_\Gamma$, що дозволяє незалежно визначити інтенсивність кольору цілого сегменту точок (рис. 3.13) у рядку растеризації. Найбільш проста апратна реалізація методу має місце при розмірах сегментів, які кратні степені двійки.

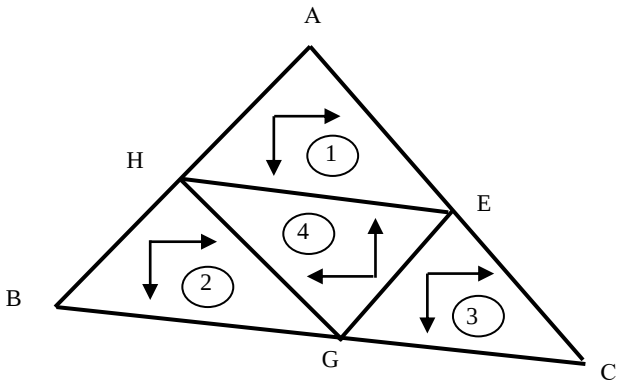


Рисунок 3.11 – Напрямки зафарбовування складових трикутників

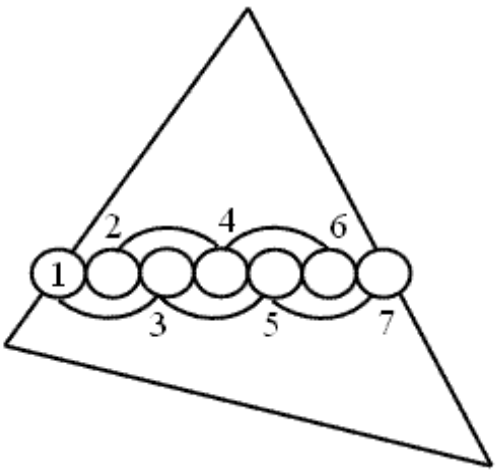


Рисунок 3.12 – Зафарбовування парних і непарних точок рядка растеризації трикутника

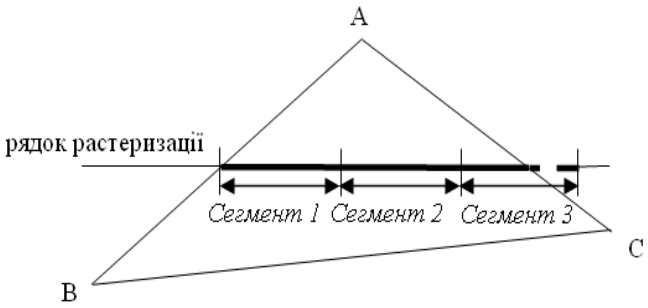


Рисунок 3.13 – Розбиття рядка растеризації на сегменти

При сегментній реалізації визначають прирости і нтенсивностей кольору для всіх точок сегмента та інтенсивність кольору першої точки наступного сегмента. При формуванні останнього сегмента за рахунок маскування блокують запис у відеопам’ять тих точок, які виходять за межі рядка растеризації.

3.4 Висновки

У даному розділі розроблено теоретичні засади високопродуктивного рендерингу тривимірних зображень у системах комп’ютерної графіки.

1. Запропоновано для знаходження векторів нормалей використати квадратичну інтерполяцію між початковим та кінцевим векторами нормалей рядка растеризації трикутника, для чого довизначають нормалізований вектор у середній точці рядка растеризації трикутника. Отримано рекурентні співвідношення для обчислення поточного вектора рядка растеризації. При програмній реалізації запропонованого методу досягається підвищення продуктивності більш, ніж в 2,5 раза порівняно з класичною нормалізацією.

2. Запропоновано метод визначення векторів нормалей із використанням сферично-кутової інтерполяції. Отримано рекурентні відношення для розрахунку поточного вектора нормалі в рядку растеризації трикутника через два попередніх вектора. Розглянуто реалізацію методу як за умови визначення кута між початковим і кінцевим векторами рядка

растеризації трикутника, так і їх скалярного добутку. В останьому випадку з обчислювального процесу вилучається трудомістка процедура визначення кутів між векторами нормалей до поверхні.

3. Доведено властивість про сталість приросту інтенсивності кольору вздовж горизонтальних (вертикальних) рядків растеризації трикутника, що дозволило підвищити продуктивність зафарбовування за рахунок того, що приріст інтенсивності кольору розраховується для трикутника, а не для кожного його рядка растеризації. Це дало можливість відділити цикл підготовки від циклу зафарбовування, у середньому в 1,7 раза зменшити час зафарбовування середньостатистичного трикутника та спростити апаратну реалізацію.

4. На основі доведеної властивості про сталість приросту інтенсивності кольору запропоновано методи розпаралелення обчислювального процесу, які забезпечують підвищення продуктивності зафарбовування при збалансованому завантаженні рендерів.

5. Розроблено метод підвищення продуктивності формування графічних об’єктів за методом Гуро.

6. Розроблено метод ідентифікації відблиску на ділянці поверхні, обмеженої трикутником, який дозволяє підвищити швидкість формування тривимірних графічних зображень.

4 АНАЛІЗ ТРУДОМІСТКОСТІ ПРОЦЕДУР КІНЦЕВОЇ ВІЗУАЛІЗАЦІЇ

4.1 Склад і кількість інструкцій для визначення векторів нормалей для середньостатистичного трикутника

Ділянка поверхні, обмежена середньостатистичним трикутником, включає 100 точок і для блискучих поверхонь має коефіцієнт спекулярності – 256.

Використано трикутник АВС з $\Delta x_{AB} = 8,\ \Delta y_{AB} = 8$, $\Delta x_{AC} = 9,\ \Delta y_{AC} = 12,$ $\Delta x_{BC} = 16,\ \Delta y_{BC} = 4$.

Для оцінки трудомісткості виконання процедур кінцевої візуалізації використано кількість тактів процесора Pentium M для виконання складових інструкцій. Інструкція задає одну окрему операцію процесора, визначену системою його команд.

Результати досліджень зведено в таблицю.

Таблиця 4.1 – Склад і кількість інструкцій для середньостатистичного трикутника

Процедура	Кількість інструкцій			
	+	x	$\sqrt{\ }$	:
Розрахунок векторів уздовж ребер трикутника	108	–	–	9
Розрахунок векторів нормалей уздовж РРТ	288	–	–	33
Нормалізація векторів нормалей	200	300	100	300
Разом	596	300	100	342

Виконання наведених інструкцій при використанні процесора Pentium M складає 26110 тактів.

4.2 Визначення векторів нормалей для середньостатистичного трикутника з використанням сферично-кутової інтерполяції

Результати досліджень зведено в таблицю.

Таблиця 4.2 – Склад і кількість інструкцій для середньостатистичного трикутника з використанням сферично-кутової інтерполяції згідно з формулою $\vec{N}(t+1)=2\vec{N}(t)\cdot\cos\varphi-\vec{N}(t-1)$

Процедура	Кількість інструкцій						
	+	x	$\sqrt{}$	:	arcos	cos	Зсув
Розрахунок нормалізованих векторів нормалей до початкових і кінцевих точок РРТ	292	168	33	168	–	–	–
Розрахунок скалярних добутків векторів до кінцевих точок РРТ	36	36	–	–	–	–	–
Розрахунок $\tilde{n}os(\psi / m)$	–	–	–	12	12	12	–
Розрахунок векторів до внутрішніх точок поверхні, обмеженої трикутником	132	132	–	–	–	–	132
Разом	460	336	33	180	12	12	132

Виконання наведених інструкцій при використанні процесора Pentium M складає 17298 тактів. Порівняно з класичною реалізацією визначення нормалізованих векторів досягнуто підвищення продуктивності в 1,5 раза.

Таблиця 4.3 – Склад і кількість інструкцій для визначення векторів нормалей для середньостатистичного трикутника з використанням для сферично-кутової інтерполяції виразу $cos(\frac{\psi}{m}) \approx \frac{cos\psi - 1}{m^2} + 1$

Процедура	Кількість інструкцій					
	+	x	$\sqrt{}$	/	Зсув	± 1
Розрахунок нормалізованих векторів нормалей до кінцевих точок PPT	292	168	33	168	–	–
Розрахунок скалярних добутків векторів до кінцевих точок PPT	36	36	–	–	–	–
Розрахунок $ños(\psi - 1) / m^2 + 1$	–	12	–	12	–	24
Розрахунок векторів нормалей до внутрішніх точок поверхні, обмеженої трикутником	132	132	–	–	132	–
Разом	460	348	33	180	132	24

Виконання наведених інструкцій при використанні процесора Pentium M складає 15056 тактів. Порівняно з класичною реалізацією визначення нормалізованих векторів досягнуто підвищення продуктивності в 1,73 раза.

4.3 Визначення векторів нормалей до точок поверхні, обмеженої середньостатистичним трикутником, з використанням квадратичної інтерполяції

Результати досліджень зведено в таблицю.

Таблиця 4.4 – Склад і кількість інструкцій для визначення векторів нормалей до точок поверхні, обмеженої середньостатистичним трикутником, з використанням квадратичної інтерполяції

Процедура	Кількість інструкцій				
	«+»	«x»	«$\sqrt{\ }$»	«÷»	Зсув
Розрахунок векторів нормалей до кінцевих точок РРТ	152	66	22	75	–
Розрахунок векторів нормалей до середніх точок РРТ	60	36	12	12	24
Розрахунок $\vec{G}_i, \vec{P}_i, \vec{Q}_i$	180	–	–	–	48
Розрахунок векторів нормалей до точок РРТ	450	450	–	12	–
Разом	842	552	34	99	72

При визначенні векторів у середніх точках РРТ використано формулу (2.16). Виконання наведених в табл. В.4. інструкцій при використанні процесора Pentium M складає 15109 тактів. Порівняно з класичною реалізацією визначення нормалізованих векторів досягнуто підвищення продуктивності в 1,72 раза. При використанні формули $\vec{N}_{1/2} \approx (\vec{N}_i + \vec{N}_j)(0{,}103 \cdot \cos^2\psi - 0{,}306 \cdot \cos\psi + 0{,}705)$ час визначення векторів нормалей для середньостатистичного трикутника зменшується на 7,2%

порівняно з попередньою реалізацією за рахунок вилучення з обчислювального процесу операцій ділення і знаходження квадратного кореня.

Таблиця 4.5 – Склад і кількість інструкцій для визначення векторів нормалей до точок поверхні, обмеженої середньостатистичним трикутником, з використанням квадратичної інтерполяції та методу кінцевих різниць

Процедура	Кількість інструкцій				
	«+»	«x»	«$\sqrt{\ }$»	«÷»	Зсув
Розрахунок векторів нормалей до кінцевих точок РРТ	152	66	22	75	–
Розрахунок векторів нормалей до середніх точок РРТ	60	36	12	12	24
Розрахунок $\vec{G}_i, \vec{P}_i, \vec{Q}_i$	180	–	–	–	48
Розрахунок векторів нормалей до точок РРТ	453	9		12	
Разом	845	102	34	99	72

Для розрахунку векторів необхідно виконати 10168 тактів. Порівняно з класичною реалізацією визначення нормалізованих векторів досягнуто підвищення продуктивності в 2,56 раза за рахунок зменшення операцій множення.

4.4 Склад і кількість інструкцій для зафарбовування середньостатистичного трикутника за методом Гуро

Таблиця 4.6 – Склад і кількість інструкцій для зафарбовування середньостатистичного трикутника за методом Гуро

Процедура	Кількість інструкцій	
	«+»	«÷»
Розрахунок приростів складових інтенсивностей кольору вздовж ребер	9	9
Розрахунок інтенсивностей складових кольору точок ребер	99	–
Розрахунок приростів інтенсивностей складових кольору рядків растеризації	33	33
Розрахунок інтенсивностей складових кольору точок рядків растеризації	225	–
Разом	366	42

Виконання наведених операцій при використанні процесора Pentium M складає 2820 тактів.

Таблиця 4.7 – Склад і кількість інструкцій для зафарбовування середньостатистичного трикутника за модифікованим методом Гуро

Процедура	Кількість інструкцій		
	«+»	«÷»	«x»
Розрахунок приростів інтенсивностей складових кольору вздовж ребер	9	9	–
Розрахунок інтенсивностей складових кольору точок ребер	99	–	–

Продовження таблиці 4.7

Розрахунок приросту інтенсивностей складових	9	3	12
Розрахунок інтенсивностей складових кольору точок рядків растеризації	225	–	–
Разом	342	12	12

Зафарбовування середньостатистичного трикутника за модифікованим методом Гуро передбачає виконання 1650 тактів. Порівняно з класичною реалізацією зафарбовування середньостатистичного трикутника досягнуто підвищення продуктивності в 1,7 раза.

4.5 Висновки

Результати аналізу трудомісткості виконання процедур рендерингу підтвердили їх ефективність.

5 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ПРИСКОРЕНОГО ЗАФАРБОВУВАННЯ ТРИВИМІРНИХ ОБ'ЄКТІВ

5.1. Розробка програмного модулю для зафарбовування тривимірних графічних об'єктів з використання розроблених методів

Визначимо базові параметри для оцінки ефективності розроблених методів і засобів. Продуктивність формування графічних сцен можна оцінити за часом її формування.

При цьому необхідно розрізняти статичний та динамічний режими. Для динамічного режиму необхідна розробка відповідного програмного модулю.

В роботі розроблено програмний модуль для порівняння сформованих зображень з тестовими.

При цьому використовують формулу:

$$NMSE = \frac{\sum_i \left(R_1(i) - R_2(i)\right)^2 + \left(G_1(i) - G_2(i)\right)^2 + \left(B_1(i) - B_2(i)\right)^2}{\sum_i R_1(i)^2 + G_1(i)^2 + B_1(i)^2} \qquad (5.1)$$

де $NMSE$ – нормована середньоквадратична похибка (NMSE), i – кількість пікселів зображення, $R_1(i), G_1(i), B_1(i)$ – інтенсивності кольору відповідних кольорових каналів.

Якщо $NMSE \le 0{,}0001$, то зображення мають візуальну ідентичність.

Якщо $NMSE \in [0{,}0001 \div 0{,}00025]$, то два зображення мають несуттєві візуальні відмінності.

Якщо $[0{,}00025 < NMSE \le 0{,}001]$, то мають місце відмінності зображення, які користувач відрізняє.

Якщо $NMSE > 0{,}001$, то зображення відрізняються одне від одного.

На рисунку 5.1 наведено модель розробленого програмного забезпечення.

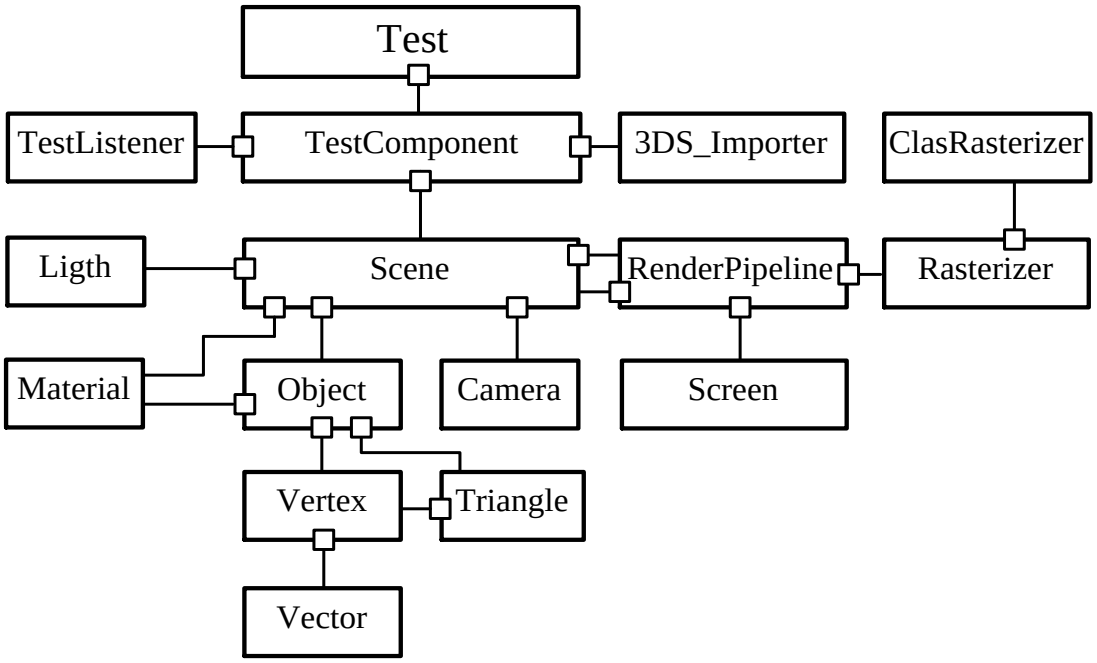


Рисунок 5.1 – Об’єктно-орієнтована модель програмного модуля

Графічний двигун включає:

– стадію геометричної підготовки, де виконуються перетворення афінного характеру та видові перетворення;

– стадію рендерингу, на якій виконується тонування відповідно до вибраного методу.

Для програми розроблено інтерфейс [17], вигляд якого зображено на рисунку 5.2.

Він включає чотири функціональні ділянки.

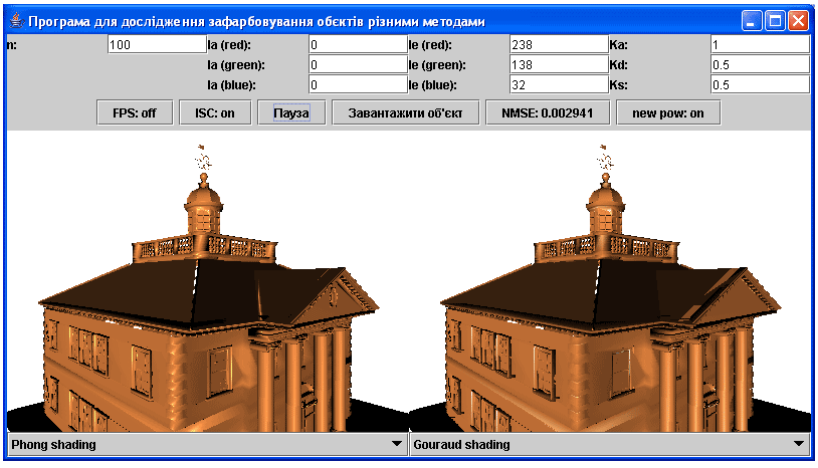


Рисунок 5.2 – Вигляд графічного інтерфейсу

Перша ділянка інтерфейсу включає поля редагування *JEdit*, що забезпечують користувачу можливість зміни параметрів тонування (рис. 5.3).

Рисунок 5.3 – Поле редагування *JEdit*

У поле редагування *JEdit* має можливість змінювати коефіцієнт n; інтенсивності амбітної складової кольору I_a та інтенсивність I_e джерел світла; коефіцієнти K_a, K_d, K_S складової інтегрального світла.

Ділянка інтерфейсу (рис. 5.4) дає можливість вибору ДФОС для визначення за допомогою кнопки "*new pow*". Для вибору об'єктів для тонування використовується кнопки "Вибрати об'єкт". При натисненні кнопки *NMSE* у відповідному вікні виводиться значення похибки.

При натисненні кнопка "Пауза" призупиняється процес тонування. У вікні "*render time*" виводиться значення час, який необхідний для тонування зображення об'єкту

У полі "*FPS*" виводиться кількість кадрів у секунду при тонування сцени.

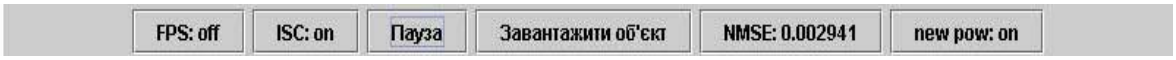


Рисунок 5.4 – Набір кнопок для управляння тонуванням

Базова частина інтерфейсу (рис. 5.5) відведена для демонстрації сформованих зображень.

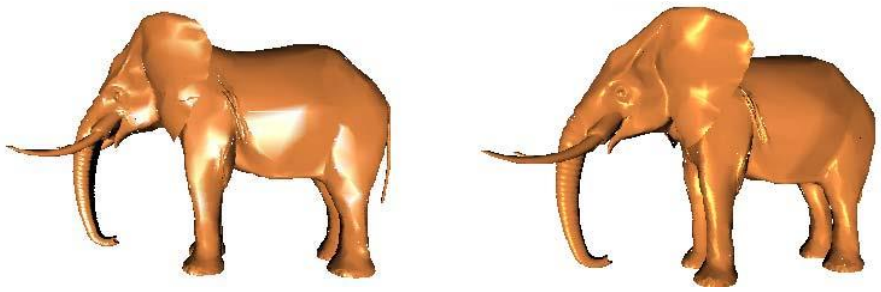


Рисунок 5.5 – Частина інтерфейсу призначена для виводу згенерованих тривимірних об’єктів

Для вибору методу зафарбовування (рисунок 5.6) виділено відповідне поле.

Рисунок 5.6 – Поле для вибору методу тонування

Лістинг основних програмних модулів та класів наведено в додатках.

5.2 Керівництво програмісту

Програму розроблено з використанням мови Java. Це передбачає встановлення на ПК *JDK* 1.4 або більш нової версії.

Програма включає такі основні класи:

Test – розроблено інтерфейс користувача.

Реалізовано вибір методу тонування (в масиві *rasterizers* розміщено класи растеризації.

За необхідності додавання нового методу тонування треба розробити клас растеризації та внести його в масив.

TestComponent – відповідає за візуалізацію на екрані сформованого об’єкта. Цей клас дає можливість занести в програму файли із об’єктами;

– додавання джерел освітлення;

– виконання афінних перетворень;

– призупинення формування графічної сцени;

– задання інтенсивностей кольорів;

– визначення $NMSE$;

Реалізовано відслідковування часу формування об’єктів.

У таблиці наведено інші класи, які використовуються як складові до перерахованих.

Таблиця 5.1 – Використані класи

idx3d_3ds_Exporter	Реалізує збереження об’єкта в файл
idx3d_3ds_Importer	Завантаження об’єкта з файлу (він повинен бу розміщений в тій ж директорії, що й програма мати розширення *.3ds)
idx3d_3ds_Camera	Реалізує розташування камери в сцені
idx3d_3ds_Color	Реалізуються різні операції з кольором
idx3d_3ds_CoreObject	Забезпечує реалізацію основних методів роботи об’єктом, та основні властивості того ж об’єкта
idx3d_Environment	Установлення фону зображення
idx3d_Light	Включає властивості джерела світла: напрям яскравість
idx3d_Lightmap	Реалізує керування картою освітлення
idx3d_Material	Забезпечує роботу із матеріалом
idx3d_Math	Реалізує деякі математичні функції, яких немає стандартному класі Math
idx3d_Matrix	Включає роботу з матрицями афінних перетворень
idx3d_Object	Характеристики об’єкта, та методи роботи з ними.
idx3d_ObjectFactory	Реалізація простих геометричних об’єктів
idx3d_Rasterizer	Реалізуються основні методи зафарбовуван об’єкта: розбиття об’єкта на трикутники, стрічкове зафарбовування трикутника
idx3d_Scene	Включає в себе всі об’єкти що є в сцені(об’єкти, зафарбовуються, джерела світла)
idx3d_Screen	Забезпечує роботу з зображенням, на екрані
idx3d_Triangle	Містить в собі всі характеристи трикутника(координати точок у вершин трикутника, координати векторів нормалей вершинах), та методи роботи з ними

Продовження таблиці 5.1 – Використані класи

idx3d_Vector	Містить в собі характеристики вектора(три координати), методи роботи з ними(сума векторів, різниця векторів, векторний та скалярний добуток векторів, визначення кута між векторами)
idx3d_Vertex	Містить у собі координати вектора та координати точки початку вектора
PhongRasterizer	Клас у якому реалізовано зафарбовування за методом Фонга
GouraudRasterizer	Реалізація зафарбовування методом Гуро
AdaptivGourauAndFong	Клас у якому реалізовано зафарбовування адаптивним методом(для трикутника вибирається, яким методом зафарбовувати: Гуро чи Фонгом)
Approx_cos_line	Реалізована растеризація трикутника методом Фонга з використанням квадратичної BRDF з лінійною апроксимацією коефіцієнтів
Approximation_cos_n	Реалізована растеризація трикутника методом Фонга з використанням експоненціальної BRDF
log2_n	Растеризація з використанням BRDF піднесення до степеня двійки
Phong_Shlik	Реалізація методу Фонга з BRDF запропонованою Шліком
Phong_New_Shlik	Реалізація методу Фонга з покращеною формулою BRDF запропонованою Шліком

Для реалізації методу тонування формується свій клас і наслідковується клас idx3d_Rasterizer.

У власному класі перевантажити наступні функції:

initRender – в цій функції необхідно задати епапи, які виконуються перед тонуванням;

prerenderFirstPart – визначаються дії для першої ділянки трикутника;

prerenderSecondPart – визначаються дії для другої ділянки трикутника;

renderLine – визначаються інтенсивності кожного пікселя РР.

У функції *renderLine* є цикл $for\left(int\ x = xL;\ x < xR;\ x++\right)$, в якому розраховуються інтенсивності кольору пікселів відрізка прямої.

Так для того щоб задати ДФВЗ, потрібно у зазначеному циклі замінити розрахунок $angle = (float)Math.pow(angle, n);$ на свій варіант обчислення дистрибутивної функції відбивної здатності поверхонь.

У даному циклі розрахована інтенсивність кольору заноситься в масив p, який є властивістю об'єкта $screen.$ Для вилучення невидимих поверхонь розраховується координата глибини z для кожного пікселя поверхонь і заноситься в масив $zBuffer$.

5.3 Керівництво оператору

Розроблений програмний модуль, програма призначена для високопродуктивного формування реалістичних зображення.

Передбачені можливості вибору параметрів та керування процесом формування графічних сцен.

Системні вимоги для роботи програми:

Мінімальні:

- процесор Pentium 400, або вище;
- розмір ОЗП 64Mb;
- операційна система MS Windows 95;
- 1 Mb на жорсткому диску для розміщення самої програми;
- наявність Java SDK або RE 1.3.

Оптимальні:

- процесор Intel Pentium 1000;
- розмір ОЗП 128Mb;
- операційна система MS Windows 98;
- 1 Mb на жорсткому диску для розміщення самої програми.

Для запуску програмного модуля використовується файл "start.bat" з файлового менеджера. Це призводить до відтворення вікна, яке зображено на рисунку 5.7.

Колір має три компоненти: фонову, дифузну та спекулярну. Інтенсивності цих складових задаються у відповідних полях (Ia і Ie).

Оскільки в програмі використовується R,G,B система, то для кожного з каналів задається значення інтенсивності.

Для зафарбовування задаються коефіцієнти *Ka, Kd, Ks,* значення від цих коефіцієнтів змінюється від нуля до одиниці

Задається також коефіцієнт спекулярності n поверхні.

Його вибирають з діапазону від 0 до 1000 залежно від шороховатості поверхні.

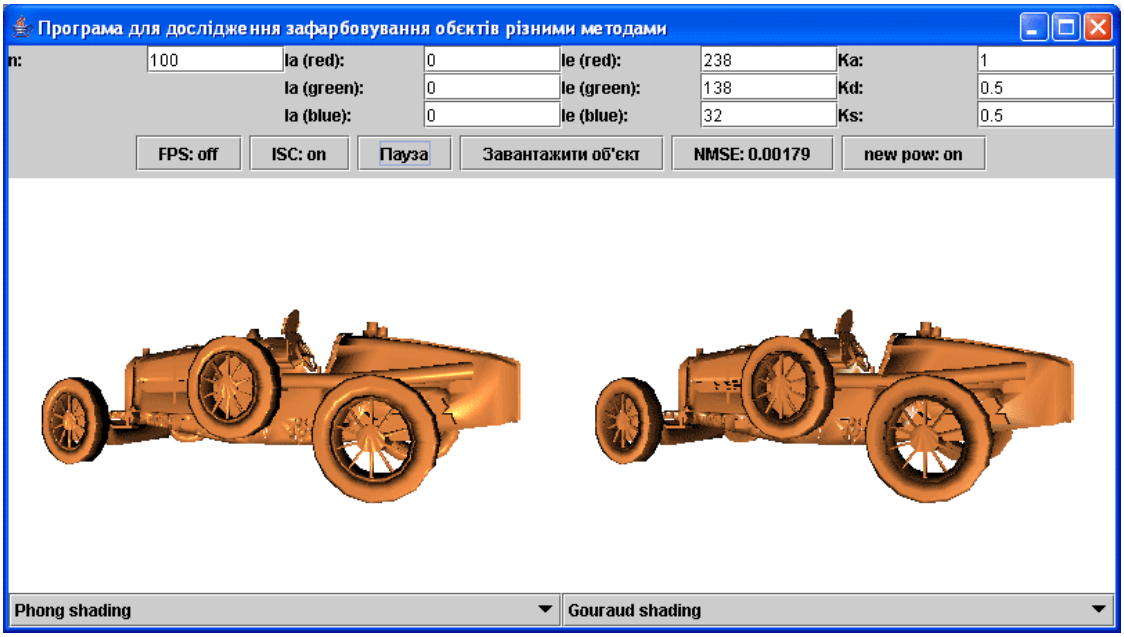


Рисунок 5.7 – Головне вікно програми idx3d

В програмі передбачено вибір графічного об’єкту з архіву. Для цього треба натиснути мишкою на кнопку “Завантажити об’єкт” (рисунок 5.8) і в вікні „*Open*”, вибрати потрібний об’єкт.

Він має розширення *.3DS.

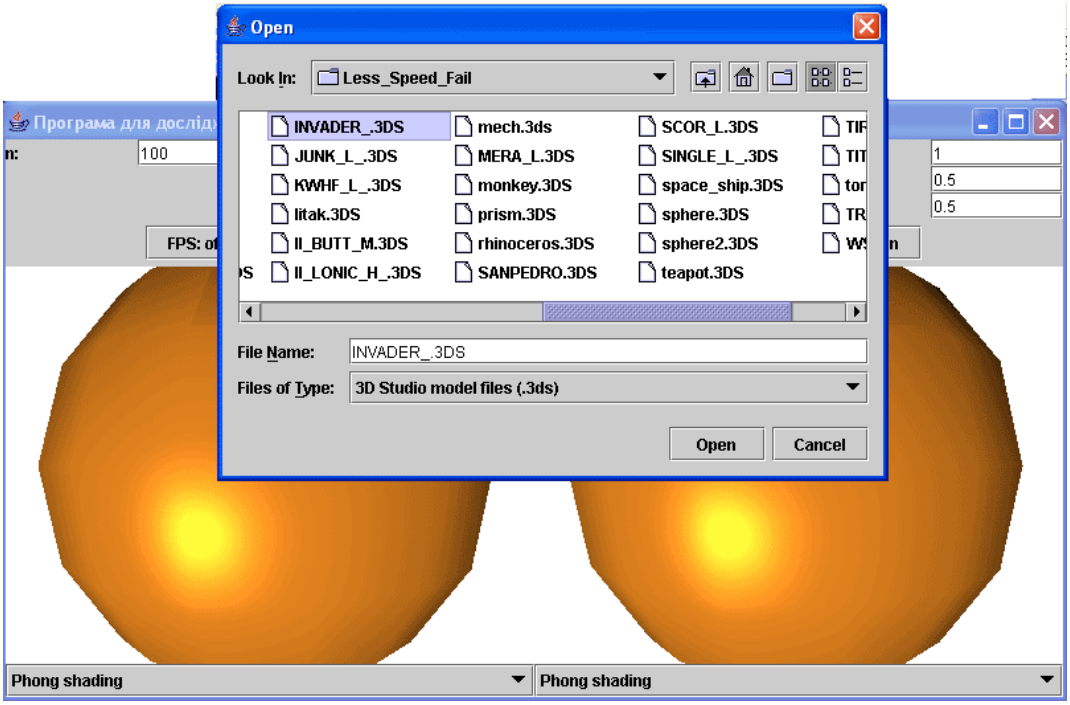


Рисунок 5.8 – Вікно для вибору об’єкту

Для вибору методу тонування достатньо натиснути мишкою на випадаючий список та вибрати потрібний метод.

Після виконання перекислених дій у вікні програмного модуля появиться сформоване зображення (рис. 5.9).

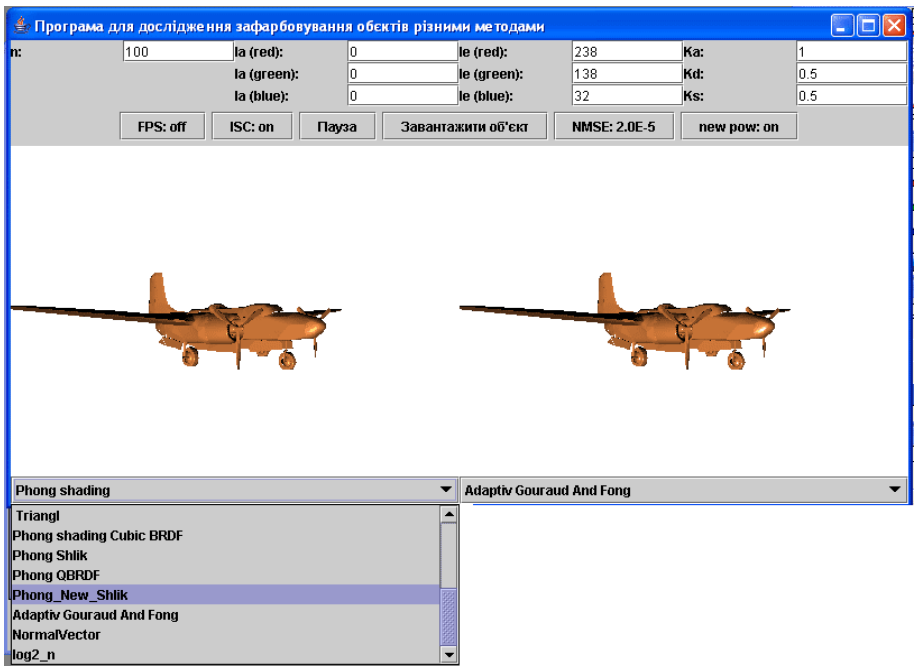


Рисунок 5.9 – Головне вікно програми idx3d

У розробленому програмному модулі є вікно “FPS” для виведення кількості кадрів за секунду та часу рендеренгу.

Кнопка „Пауза” дає можливість призупинити на деякий час формування графічного зображення.

При натисненні кнопки $NMSE$ у відповідному вікні виводиться значення нормованої середньоквадратичної похибки.

Кнопка *newpow* використовується для вибору ДФВЗ.

5.4 Висновки

Розроблено програмні засоби для реалізації розроблених в роботі високопродуктивних методів зафарбовування тривимірних об’єктів.

Програми розроблено з використанням мови Java.

Розроблено засоби для порівняння сформованих тривимірних об’єктів з тестовими фігурами.

6 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для кожної науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота «Методи та засоби високопродуктивного формування тривимірних графічних зображень» відноситься до науково-технічних робіт, які з самого початку орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку мають бути виконані такі етапи робіт:

1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;

2) розраховано витрати на здійснення науково-технічної розробки;

3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

6.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 6.1 [26].

Таблиця 6.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатацій-них витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів

Продовження таблиці 6.1

Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використо-вуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використо-вуються у виробництві

Продовження таблиці 6.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 6.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	3	3	3
2. Ринкові переваги (наявність аналогів)	3	3	4
3. Ринкові переваги (ціна продукту)	3	2	3
4. Ринкові переваги (технічні властивості)	3	4	2
5. Ринкові переваги (експлуатаційні витрати)	3	3	2
6. Ринкові перспективи (розмір ринку)	3	4	2
7. Ринкові перспективи (конкуренція)	3	2	4
8. Практична здійсненність (наявність фахівців)	4	4	3
9. Практична здійсненність (наявність фінансів)	3	3	3

Продовження таблиці 6.2

10. Практична здійсненність (необхідність нових матеріалів)	3	3	3
11. Практична здійсненність (термін реалізації)	2	3	4
12. Практична здійсненність (розробка документів)	3	3	3
Сума балів	36	37	36
Середньоарифметична сума балів $СБ_c$	36,3		

За результатами розрахунків, наведених в таблиці 6.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 6.3 [26].

Таблиця 6.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ ,розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вище середнього
21…30	Середній
11…20	Нижче середнього
0…10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Методи та засоби високопродуктивного формування тривимірних графічних зображень» становить 36,3 бала, що, відповідно до таблиці 6.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

6.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення розрахуємо за формулою [27]:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i \,, \tag{6.1}$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу *і-го* технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом і при цьому має виконуватись умова $\sum_{i=1}^{k} \alpha_i = 1$;

β_i – відносне значення *і-го* технічного показника якості нової розробки.

Відносні значення β_i для різних випадків розраховуємо за такими формулами:

– для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{нi}}{I_{ai}} \,, \tag{6.2}$$

де $I_{нi}$ та $I_{на}$ – чисельні значення конкретного *і*-го технічного показника якості відповідно для нової розробки та аналога;

– для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{нi}} \,; \tag{6.3}$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 6.4.

Таблиця 6.4 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірю-вання	Аналог	Проектоване програмне забезпечення	Відношення параметрів нової розробки до аналога	Питома вага показника
Підтримувані ОС	-	1	2	2	0,1
Зручність інтерфейсу	бал	6	8	1,33	0,15
Навантаження на процесор ЕОМ	%	16	12	1,33	0,3
Швидкість обробки зображення	бал	6	8	1,33	0,25
Кількість задіяних модулів обробки (функції підтримки)	шт.	3	5	1,67	0,2

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення складе:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i = 2 \cdot 0{,}1 + 1{,}33 \cdot 0{,}15 + 1{,}33 \cdot 0{,}3 + 1{,}33 \cdot 0{,}25 + 1{,}67 \cdot 0{,}2 = 1{,}47.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,47 рази.

В ході дослідження було обрано 3 методи для аналізу, а саме технологія TruForm, технологія Phong Tessellation та метод додаткової тріангуляції, який оперує з інтенсивностями кольору.

Технологія TruForm здійснює додаткову тріангуляцію, застосовуючи кубічні патчі Бєзьє. Дана технологія дозволяє підвищити рівень реалістичності 3D-зображень і не потребує модифікації підсистеми рендерингу відеокарти. TruForm містить етап передачі полігональної інформації зі сцени на 3D-акселератор, етап внутрішнього перетворення трикутників у криві поверхні, а також створення нових трикутників за допомогою визначення контрольних точок.

Технологія Phong Tessellation передбачає додаткову теселяцію для формування вигнутих контурів, обраховуючи тільки позиції вершин полігональної мережі та нормалі в них. У розрахунках використовуються барицентричні координати та квадратичні патчі для відновлення поверхні.

Метод додаткової тріангуляції, який оперує з інтенсивностями кольору, полягає у визначенні точок, що мають найбільше значення інтенсивності кольору на кожному з ребер трикутника. Такий метод додаткової тріангуляції працює з інтенсивностями кольору, саме тому він є найбільш прийнятним для використання зафарбовування за методом Гуро. Зафарбовування за Фонгом, у даному випадку, потребує визначення нормалей у контрольних точках.

У Phong Tessellation якість результуючого зображення на багато покращується у порівнянні з технологією TruForm, однак технології Phong Tessellation притаманні деякі інші недоліки. По-перше, для реалізації потребується виконати на багато більший обсяг обчислень, у порівнянні з технологією Phong Shading, що вважається високотрудомісткою. По-друге, подібно до технології TruForm, технологія Phong Tessellation використовується для кожного об’єкта графічної сцени, що передбачає додаткові розрахунки для об’єктів, що займають невелику площу на екрані користувача.

Попередньо проведений аналіз довів, що розробка методів та програмних засобів є доцільною. В результаті отримаємо продукт, що покриває недоліки існуючих рішень і забезпечує високопродуктивне формування тривимірних графічних зображень.

6.3 Розрахунок витрат на здійснення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему «Методи та засоби високопродуктивного формування тривимірних графічних зображень», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

До витрат на оплату праці належать витрати на виплату основної та

додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Витрати на основну заробітну плату дослідників ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою [26]:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \quad (6.4)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, T_p=21 дні.

$$З_о = 12100{,}00 \cdot 21 / 21 = 12100{,}00 \; грн.$$

Проведені розрахунки зведемо до таблиці.

Таблиця 6.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	12100,00	576,19	21	12100,00
Ст. науковий співробітник	11200,00	533,33	21	11200,00
Інженер-програміст	10850,00	516,67	21	10850,00
Аналітик систем обробки графічних даних	10900,00	519,05	9	4671,43
Технік	6800,00	323,81	15	4857,14
Консультант-аналітик обробки цифрових зображень	11100,00	528,57	11	5814,29
Всього				49492,86

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Методи та засоби високопродуктивного формування тривимірних графічних зображень» розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i , \qquad (6.5)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}} , \qquad (6.6)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо $Мм$=2379,00 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [26];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об’єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно T_p = 21 дн;

$t_{зм}$ – тривалість зміни, год.

$C_1 = 2379{,}00 \cdot 1{,}10 \cdot 1{,}65 / (21 \cdot 8) = 25{,}70$ *грн.*

$З_{р1} = 25{,}70 \cdot 10{,}00 = 257{,}02$ *грн.*

Таблиця 6.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Встановлення допоміжного обладнання	10,00	2	1,10	25,70	257,02
Інсталяція програмного забезпечення	7,50	4	1,50	35,05	262,86
Встановлення цифрових обчислювальних систем	4,00	5	1,70	39,72	158,88
Відлагодження інтерполяційних модулів графічної системи	2,00	5	1,70	39,72	79,44
Підготовка цифрової експериментальної моделі геометричного інтерполятора	6,00	4	1,50	35,05	210,29
Формування бази даних підсистеми рейдерингу	12,00	3	1,35	31,54	378,52
Формування бази даних для підсистеми кадрового буфера	12,00	3	1,35	31,54	378,52
Узгодження параметрів системи комп’ютерної графіки для формування реалістичних зображень	4,00	5	1,70	39,72	158,88
Тренування системи	3,50	4	1,50	35,05	122,67
Всього					2007,07

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (6.7)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 10%.

$З_{дод}$ = (49492,86 + 2007,07) · 10 / 100% = 5149,99 грн.

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (6.8)$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

Зн = (49492,86 + 2007,07 + 5149,99) · 22 / 100% = 12462,98 грн.

До сировини та матеріалів належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Методи та засоби високопродуктивного формування тривимірних графічних зображень».

Витрати на матеріали (*М*), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$М = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} В_j \cdot Ц_{вj}, \quad (6.9)$$

де $Н_j$ – норма витрат матеріалу *j*-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу *j*-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ = *1,1 ... 1,15*);

$В_j$ – маса відходів *j*-го найменування, кг;

$Ц_{вj}$ – вартість відходів *j*-го найменування, грн/кг.

$М_1$ = 3,00 · 95,00 · 1,1 - 0,000 · 0,00 = 313,50 грн.

Проведені розрахунки зведемо до таблиці.

Таблиця 6.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Папір канцелярський офісний ECONOMIC (A4-500)	95,00	3,00	0,000	0,00	313,50
Папір для заміток ECONOMIC (A5)-60	48,00	3,00	0,000	0,00	158,40
Начиння канцелярське DATUM FX	165,00	4,00	0,000	0,00	726,00
Органайзер офісний DATUM Office	102,00	4,00	0,000	0,00	448,80
Картридж для принтера HP-210A	920,00	2,00	0,000	0,00	2024,00
Диск оптичний VEKO-10 (CD-R)	12,50	3,00	0,000	0,00	41,25
Диск оптичний VEKO-W (CD-RW)	15,00	3,00	0,000	0,00	49,50
FLASH-пам'ять Kingstar (32 ГБ) Class 10	210,00	1,00	0,000	0,00	231,00
FLASH-пам'ять Kingstar (64 ГБ) Class 10 A	345,00	1,00	0,000	0,00	379,50
Всього					4371,95

Витрати на комплектуючі *($К_в$)*, які використовують при проведенні НДР на тему «Методи та засоби високопродуктивного формування тривимірних графічних зображень», відсутні

До спецустаткування для наукових (експериментальних) робіт належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{спец} = \sum_{i=1}^{k} Ц_i \cdot C_{пр.i} \cdot K_i \ , \qquad (6.10)$$

де Ці – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{пр.i}$ –кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1,10…1,12$);

k – кількість найменувань устаткування.

$B_{спец} = 18960,00 \cdot 1 \cdot 1,1 = 20856,00$ *грн.*

Отримані результати зведемо до таблиці:

Таблиця 6.8 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Мультимедійна проекційна система	1	18960,00	20856,00
Всього			20856,00

До програмного забезпечення для наукових (експериментальних) робіт належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість ПЗ розраховуємо за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (6.11)$$

де $Ц_{іпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$С_{прг.і}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$К_і$ – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($К_і = 1,10…1,12$);

k – кількість найменувань програмних засобів.

$В_{прг} = 5420,00 \cdot 1 \cdot 1,1 = 5962,00$ *грн.*

Отримані результати зведемо до таблиці:

Таблиця 6.9 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
ОС Windows	1	5420,00	5962,00
Прикладний пакет Microsoft Office	1	5230,00	5753,00
Прикладний пакет розробника Java	1	4100,00	4510,00
Комп'ютерна програма для рендерингу поверхонь тривимірних фігур	1	3150,00	3465,00
Комп'ютерна програма для зафарбовування поверхонь тривимірних об'єктів за модифікованим методом Гуро	1	3150,00	3465,00
Комп'ютерна програма для адаптивної нормалізації векторів у рядку растеризації	1	3150,00	3465,00
Комп'ютерна програма для зафарбовування поверхонь тривимірних об'єктів за модифікованим методом Фонга	1	3150,00	3465,00
Комп'ютерна програма для розрахунку дистрибутивної функції відбивної здатності поверхні для задач рендерингу	1	3150,00	3465,00
Всього			33550,00

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (6.12)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$A_{обл} = (21570{,}00 \cdot 1) / (2 \cdot 12) = 898{,}75$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 6.10 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Програмно-аналітичний комплекс	21570,00	2	1	898,75
Графічно-обчислювальний комплекс обробки даних	25250,00	2	1	1052,08
Програмні продукти обробки графічних даних	15600,00	3	1	433,33
Обладнання виводу графічної інформації	9200,00	4	1	191,67
Місце оператора спеціалізоване	8620,00	5	1	143,67

Продовження таблиці 6.10

Офісна оргтехніка	11400,00	4	1	237,50
Дослідницька лабораторія	225000,00	25	1	750,00
Всього				3707,00

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{вni}}{\eta_i}, \quad (6.13)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_e$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), приймемо $Ц_e$ = 4,25 грн;

$K_{вni}$ – коефіцієнт, що враховує використання потужності, $K_{вni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$B_e = 0,32 \cdot 160,0 \cdot 4,25 \cdot 0,95 / 0,97 = 217,60$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 6.11 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Програмно-аналітичний комплекс	0,32	160,0	217,60
Графічно-обчислювальний комплекс обробки даних	0,56	160,0	380,80
Мультимедійна проекційна система	0,11	45,0	21,04
Обладнання виводу графічної інформації	0,18	20,0	15,30
Місце оператора спеціалізоване	0,15	160,0	102,00
Офісна оргтехніка	0,75	20,0	63,75
Всього			800,49

До службових відряджень дослідної роботи на тему «Методи та засоби високопродуктивного формування тривимірних графічних зображень» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати розраховуємо як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (З_о + З_р) \cdot \frac{Н_{св}}{100\%}, \quad (6.14)$$

де $Н_{св}$ – норма нарахування за статтею «Службові відрядження», приймемо $Н_{св}$ = 20%.

$B_{св}$ = (49492,86 + 2007,07) · 20 / 100% = 10299,99 грн.

Витрати на роботи, які виконують сторонні підприємства, установи і організації розраховуємо як 30…45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{сп} = (З_о + З_р) \cdot \frac{Н_{сп}}{100\%}, \quad (6.15)$$

де $Н_{сп}$ – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», приймемо $Н_{сп}$= 30%.

$B_{сп}$ = (49492,86 + 2007,07) · 30 / 100% = 15449,98 грн.

До інших витрат належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Інші витрати розраховуємо як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (6.16)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів}$ = 50%.

$І_в$ = (49492,86 + 2007,07) · 50 / 100% = 25749,96 грн.

До накладних (загальновиробничих) витрат належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Накладні (загальновиробничі) витрати розраховуємо як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$В_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}\ , \qquad (6.17)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв}$ = 100%.

$В_{нзв}$ = (49492,86 + 2007,07) · 100 / 100% = 51499,93 грн.

Витрати на проведення науково-дослідної роботи на тему «Методи та засоби високопродуктивного формування тривимірних графічних зображень» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$В_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + В_{спец} + В_{прг} + А_{обл} + В_е + В_{св} + В_{сп} + І_в + В_{нзв} \quad (4.18)$$

$В_{заг}$= 49492,86 +2007,07 +5149,99 +12462,98209 +4371,95 +0,00 +20856,00 + 33550,00 + 3707,00 +800,49 +10299,99 +15449,98 +25749,96 +51499,93 = 235398,19 грн.

Загальні витрати *ЗВ* на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{В_{заг}}{\eta}, \qquad (6.19)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,95.

ЗВ = 235398,19 / 0,95 = 247787,57 грн.

6.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Методи та засоби високопродуктивного формування тривимірних графічних зображень» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

Таблиця 6.12 – Кількість споживачів

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	1500	2100	2500	1000

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

ΔN – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, приймемо 15000 осіб;

$Ц_б$ – вартість програмного продукту у році до впровадження результатів розробки, приймемо 960,00 грн;

$\pm\Delta Ц_o$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, приймемо 50,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [26]:

$$\Delta П_i = (\pm \Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (6.20)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2021 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо ρ =20%;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2021 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta П_1 = (50,00 \cdot 15000,00 + 1010,00 \cdot 1500) \cdot 0,83 \cdot 0,2 \cdot (1 - 0,18/100\%) = 308311,80$$

грн.

Збільшення чистого прибутку 2-го року:

$$\Delta П_2 = (50,00 \cdot 15000,00 + 1010,00 \cdot 3600) \cdot 0,83 \cdot 0,2 \cdot (1 - 0,18/100\%) = 597022,32$$

грн.

Збільшення чистого прибутку 3-го року:

$$\Delta П_3 = (50,00 \cdot 15000,00 + 1010,00 \cdot 6100) \cdot 0,83 \cdot 0,2 \cdot (1 - 0,18/100\%) = 940725,32$$

грн.

Збільшення чистого прибутку 4-го року:

$$\Delta П_4 = (50,00 \cdot 15000,00 + 1010,00 \cdot 7100) \cdot 0,83 \cdot 0,2 \cdot (1 - 0,18/100\%) = 1078206,52$$

грн.

Приведена вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \quad (6.21)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ=0,12;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$ПП$ =308311,80/(1+0,12)^1+597022,32/(1+0,12)^2+940725,32/(1+0,12)^3+

+1078206,52/(1+0,12)^4=275278,39+475942,54+669589,70+685219,74=2106030,37

Величина початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{інв} \cdot ЗВ, \tag{6.22}$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=2;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 247787,57 грн.

$PV = k_{інв} \cdot ЗВ$ = 2 · 247787,57 = 495575,14 грн.

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \tag{6.23}$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 2106030,37 грн;

PV – теперішня вартість початкових інвестицій, 495575,14 грн.

$E_{абс} = ПП - PV$ = 2106030,37 - 495575,14 = 1610455,23 грн.

Внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1, \quad (6.24)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 1610455,23 грн;

PV – теперішня вартість початкових інвестицій, 495575,14 грн;

$T_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1$ = (1+1610455,23/495575,14)$^{1/4}$= 0,44.

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \quad (6.25)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2021 році в Україні d =0,1;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,2.

$\tau_{мін}$ = 0,1+0,2 = 0,3 < 0,44 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Методи та засоби високопродуктивного формування тривимірних графічних зображень» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}, \tag{6.26}$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$ = 1 / 0,44 = 2,29 р.

$T_{ок}$ < 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

6.5 Висновки

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Методи та засоби високопродуктивного формування тривимірних графічних зображень» становить 36,3 бала, що свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,47 рази.

$T_{ок} <$ 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення НДДКР за темою «Методи та засоби високопродуктивного формування тривимірних графічних зображень».

ВИСНОВКИ

Проведено аналіз методів і засобів зафарбовування тривимірних об’єктів для задач високореалістичної графіки. Обґрунтовано вибір для модифікації методів Фонга та Гуро.

Запропоновано високопродуктивний метод визначення векторів для задач рендерингу, особливість якого полягає в використанні для апроксимації поліному другого степеня, що дозволило зменшити час формування зображень графічних сцен. Отримані вирази для визначення векторів нормалей мають порівняно з існуючими меншу обчислювальну складність.

Розроблено новий метод визначення векторів у рядку растеризації, особливість якого полягає у використанні рекурентних співвідношень, що дозволяє суттєво спростити нормалізацію векторів.

Розроблено новий метод підвищення реалістичності рендерингу Гуро, особливість якого полягає у визначені перетину відблиску сторонами трикутника з метою додаткової тріангуляції, що дозволило підвищити реалістичність формування графічних сцен за рахунок урахування спекулярної складової кольору.

Розроблено метод ідентифікації відблиску на поверхнях об’єктів, особливість якого полягає у аналізі розміщення векторів нормалей до вершин трикутника і серединного вектора, що дає можливість виключити з процесу рендерингу визначення спекулярної складової кольору і, як наслідок, підвищити продуктивність формування графічних сцен.

Розроблено алгоритми та програмні засоби для підвищення продуктивності формування графічних сцен.

Отримані в магістерській роботі наукові та практичні результати можна використати для побудови високопродуктивних засобів рендерингу в комп’ютерних системах візуалізації тривимірних зображень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Романюк О. Н., Чорний А. В. Високопродуктивні методи та засоби зафарбовування тривимірних графічних об’єктів : Монографія. Вінниця : УНІВЕСУМ-Вінниця, 2006. 190 с.

2) Романюк О. Н. Комп'ютерна графіка : Навч. посіб. Вінниця : ВНТУ, 1999. 130 с.

3) Херн Д., Бейкер М. Комп'ютерна графіка та стандарт OpenGL. 3-тє вид. Вільямс, 2005. 1168 с.

4) Шейдер. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Шейдер (дата звернення: 28.11.2021).

5) Хілл Ф. OpenGL. Програмування комп'ютерної графіки. Санкт-Петербург, 2002. 1088 с.

6) Романюк О. Н. Дослідження дистрибутивних функцій відбивної здатності поверхонь. Оптико-електронні інформаційно-енергетичні технології. 2007. Т. 13, № 1. С. 45–50.

7) Флемінг Б. Фотореалізм. Професійні прийоми роботи. ДМК, 2000. 384 с.

8) Романюк О. Н. Новий підхід до підвищення реалістичності зафарбовування тривимірних об’єктів за методом Гуро. Інформаційні технології та комп'ютерна інженерія. 2005. № 2. С. 106.

9) Романюк О. Н., Сторчак О. Алгоритми тріангуляції. Коміздат. 2004. URL: http://citforum.ck.ua/programming/theory/alg_triangl/ (дата звернення: 28.11.2021).

10) Роджерс Д. Ф. Алгоритмічні основи машинної графіки. Москва : Мир, 1989. 503 с.

11) Романюк О. Н. Класифікація дистрибутивних функцій відбивної здатності поверхні. Інформатика, кібернетика і обчислювальна техніка. 2008. Т. 132, № 9. С. 145–151.

12) Романюк О. Н., Яковенко О.О., Ціхановська О.М., Дудник О.О.,

Чехместрук Р.Ю. Обзор пакетів прикладних програм для тривимірної графіки. Електронні інформаційні ресурси: створення, використання, доступ. Пам'яті Олексія Петровича Стахова : зб. матеріалів міжнар. наук.-практ. інтернет-конф., м. Суми/Вінниця, 9-10 листоп. 2021 р. Суми, 2021. С. 182-189.

13) Романюк О.Н., Яковенко О.О., Романюк О.В., Котлик С.В. Аналіз кросплатформового програмного інтерфейсу OpenGL і його нововведень. Інформаційні технології і автоматизація – 2021: матеріали XIV міжнар. наук.-практ. конф., м. Одеса, 21-22 жовт. 2021 р. Одеса, 2021. С. 255-260.

14) Романюк О. Н., Романюк О.В., Яковенко О.О. Підвищення продуктивності рендерингу Гуро. Молодь у світі сучасних технологій за тематикою: Використання інформаційних та комунікаційних технологій в сучасному цифровому суспільстві : матеріали IX міжнар. наук.-практ. конф. студентів, аспірантів та молодих вчен., м. Херсон, 4-5 черв. 2020 р. Херсон, 2020. С. 190-193.

15) Романюк О. Н., Романюк О.В., Яковенко О.О. Метод прискореного зафарбовування поверхонь 3D-об'єктів. Молодь у світі сучасних технологій за тематикою: Використання інформаційних та комунікаційних технологій в сучасному цифровому суспільстві : матеріали IX міжнар. наук.-практ. конф. студентів, аспірантів та молодих вчен., м. Херсон, 4-5 черв. 2020 р. Херсон, 2020. С. 187-190.

16) Романюк О. Н., Романюк О. В., Яковенко О. О. Метод спрощеного визначення векторів для задач рендерингу. Молодь в науці: дослідження, проблеми, перспективи : Всеукр. науково-практ. інтернет-конф., м. Вінниця, 19 трав. 2020 р. Вінниця, 2020.

17) Яковенко О. О., Войтко В. В. Розробка інтерфейсу користувача в Unity. Матеріали XLVI науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2017) : зб. доп., м. Вінниця, 22-24 берез. 2017 р. Вінниця, 2017.

18) Вяткин С. І., Романюк О. Н., Рейда О. М., Яковенко О. О.

Візуалізація тривимірних об'єктів з використанням аналітичного завдання освітлення навколишнього середовища. Вісник Херсонського національного технічного університету. 2019. № 2(69). С. 106-111.

19) Комп’ютерна програма для рендерингу поверхонь тривимірних фігур : пат. 91843 Україна. Опубл. 28.08.2019.

20) Комп`ютерна програма для зафарбовування поверхонь тривимірних об`єктів за модифікованим методом Гуро : пат. 91846 Україна. Опубл. 28.08.2019.

21) Комп’ютерна програма для адаптивної нормалізації векторів у рядку растеризації : пат. 91847 Україна. Опубл. 28.08.2019.

22) Комп`ютерна програма для зафарбовування поверхонь тривимірних об`єктів за модифікованим методом Фонга : пат. 91848 Україна. Опубл. 28.08.2019.

23) Комп`ютерна програма для розрахунку дистрибутивної функції відбивної здатності поверхні для задач рендерингу : пат. 91849 Україна. Опубл. 28.08.2019.

24) Комп’ютерна програма для розрахунку спекулярної складової кольору з використанням нової моделі відбивної здатності поверхні : пат. 91850 Україна. Опубл. 28.08.2019.

25) Комп`ютерна програма для текстурування з урахуванням перехресного співвідношення чотирьох колінеарних точок : пат. 91957 Україна. Опубл. 28.08.2019.

26) Козловський В. О., Лесько О. Й., Кавецький В. В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. Вінниця : ВНТУ, 2021. 42 с.

27) Кавецький В. В., Козловський В. О., Причепа І. В. Економічне обґрунтування інноваційних рішень. Вінниця : ВНТУ, 2016. 113 с.

28) Walsh P. Advanced 3D game programming with DirectX 10.0. Plano, Tex : Wordware Pub., 2008. 529 c.

29) Phong B. T. Illumination for computer generated

pictures. Communications of the ACM. 1975. Т. 18, № 6. С. 311–317. URL: https://doi.org/10.1145/360825.360839 (дата звернення: 28.11.2021).

30) Illumination Models. Glasnost. URL: https://cutt.ly/EyNkxyY (дата звернення: 28.11.2021).

31) Microfacet BRDF. Simon's Tech Blog. URL: https://cutt.ly/vyNkxk0 (дата звернення: 28.11.2021).

32) Cook R. L., Torrance K. E. A reflectance model for computer graphics. ACM SIGGRAPH Computer Graphics. 1981. Т. 15, № 3. С. 307–316. URL: https://doi.org/10.1145/965161.806819 (дата звернення: 28.11.2021).

33) Cook R. L., Torrance K. E. A Reflectance Model for Computer Graphics. ACM Transactions on Graphics. 1982. Т. 1, № 1. С. 7–24. URL: https://doi.org/10.1145/357290.357293 (дата звернення: 28.11.2021).

34) Schlick C. An Inexpensive BRDF Model for Physically-based Rendering. Computer Graphics Forum. 1994. Т. 13, № 3. С. 233–246. URL: https://doi.org/10.1111/1467-8659.1330233 (дата звернення: 28.11.2021).

35) Schlick C. A Fast Alternative to Phong's Specular Model. Graphics Gems. 1994. С. 385–387. URL: https://doi.org/10.1016/b978-0-12-336156-1.50048-3 (дата звернення: 28.11.2021).

36) Gouraud H. Continuous Shading of Curved Surfaces. IEEE Transactions on Computers. 1971. С-20, № 6. С. 623–629. URL: https://doi.org/10.1109/t-c.1971.223313 (дата звернення: 28.11.2021).

ДОДАТОК А

Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ
д.т.н., проф. О. Н. Романюк
"13" вересня 2021 р.

Технічне завдання
на магістерську кваліфікаційну роботу «Методи та засоби високопродуктивного формування тривимірних графічних зображень»
за спеціальністю
121 – Інженерія програмного забезпечення

Керівник магістерської кваліфікаційної роботи:
д.т.н., проф. О. Н. Романюк
"____"_____2021 р.
Виконала:
студентка гр.2ПІ-20м О. О. Яковенко
"____"_____2021 р.

Вінниця – 2021 року

1. Найменування та галузь застосування

Магістерська кваліфікаційна робота: «Методи та засоби високопродуктивного формування тривимірних графічних зображень».

Галузь застосування – системи комп’ютерної графіки.

2. Підстава для розробки.

Підставою для виконання магістерської кваліфікаційної роботи (МКР) є індивідуальне завдання на МКР та наказ № 277 від «24» вересня 2021 року ректора по ВНТУ про закріплення тем МКР.

3. Мета та призначення розробки.

Метою роботи є підвищення продуктивності та реалістичності формування тривимірних графічних зображень за рахунок розробки нових методів і засобів.

Призначення роботи – розробка методів і засобів зафарбовування тривимірних графічних об’єктів.

3 Вихідні дані для проведення МКР

Перелік основних літературних джерел, на основі яких буде виконуватись МКР.

1. Романюк О. Н., Чорний А. В. Високопродуктивні методи та засоби зафарбовування тривимірних графічних об’єктів : Монографія. Вінниця : УНІВЕСУМ-Вінниця, 2006. 190 с.

2. Романюк О. Н. Комп'ютерна графіка : Навч. посіб. Вінниця : ВНТУ, 1999. 130 с.

3. Романюк О. Н. Дослідження дистрибутивних функцій відбивної здатності поверхонь. Оптико-електронні інформаційно-енергетичні технології. 2007. Т. 13, № 1. С. 45–50.

4. Романюк О. Н., Сторчак О. Алгоритми тріангуляції. Коміздат. 2004. URL: http://citforum.ck.ua/programming/theory/alg_triangl/ (дата звернення: 28.11.2021).

5. Романюк О. Н. Класифікація дистрибутивних функцій відбивної здатності поверхні. Інформатика, кібернетика і обчислювальна техніка. 2008. Т. 132, № 9. С. 145–151.

4. Технічні вимоги

Базові методи зафарбовування – методи Гуро, Фонга; режим – TrueColor; вихідні дані для зафарбовування – вектори нормалей до вершин трикутників; серединний вектор $\vec{H}$; дані про розташування джерела світла та спостерігача; коефіцієнт спекулярності поверхі; координати вершин трикутників; максимальна розрядність для задання адрес вершин трикутників – 12; вихідні дані – кінцеве зображення, зафарбоване згідно методів Фонга та Гуро

5. Конструктивні вимоги.

Конструкція пристрою повинна відповідати естетичним та ергономічним вимогам, повинна бути зручною в обслуговуванні та керуванні.

Графічна та текстова документація повинна відповідати діючим стандартам України.

6. Перелік технічної документації, що пред'являється по закінченню робіт:

– пояснювальна записка до МКР;
– технічне завдання;
– лістинги програми.

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

9. Стадії та етапи розробки:

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи
1	Аналіз методів і засобів зафарбовування	14.09.21 – 28.09.21
2	Методи прискореного визначення нормалізованих векторів для задач рендерингу	29.09.21 – 12.10.21
3	Розробка методів прискореного зафарбовування тривимірних об’єктів	13.10.21 – 26.10.21
4	Аналіз трудомісткості процедур кінцевої візуалізації	27.10.21 – 09.11.21
5	Розробка програмних засобів для прискореного зафарбовування тривимірних об’єктів	10.11.21 – 22.11.21
6	Економічна частина	23.11.21 – 01.12.21

10. Порядок контролю та прийняття.

Виконання етапів магістерської кваліфікаційної роботи контролюється керівником згідно з графіком виконання роботи.

Прийняття магістерської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком.

ДОДАТОК Б

Протокол перевірки роботи

ДОДАТОК В

Лістинг коду

Лістинг модуля AdaptivGourauAndFong.java

```
public class AdaptivGourauAndFong extends idx3d_Rasterizer
{
    private boolean bTypeRasterizer=false; //true - Guoro, false - Fong.
    private double m12, m13, m23;
    private double q_1, q_2, q_3;
    private double p_1, p_2, p_3;
    private double opt_m=0.9;
    private double opt_q=0.004;
    private double opt_p=0.008;
    private idx3d_Vector h;
    int i1, i2, i3;
    float ii12r, ii12g, ii12b;
    float ii13r, ii13g, ii13b;
    float ii23r, ii23g, ii23b;
    float liir, liig, liib;
    float riir, riig, riib;
    float lir, lig, lib;
    float rir, rig, rib;
    float i4r, i4g, i4b;
    //End for Gouraund
    //for Fong
    private idx3d_Vector v4;
    private idx3d_Vector vi13, vi12, vi23;
    private idx3d_Vector lvi, rvi;
    private idx3d_Vector lv, rv;
    //End for Fong
```

```
public boolean selectMetod()
{
   m12=p1.n.x*p2.n.x+p1.n.y*p2.n.y+p1.n.z*p2.n.z;
   m13=p1.n.x*p3.n.x+p1.n.y*p3.n.y+p1.n.z*p3.n.z;
   m23=p2.n.x*p3.n.x+p2.n.y*p3.n.y+p2.n.z*p3.n.z;
   if(Math.abs(m12)>opt_m ||
         Math.abs(m13)>opt_m ||
         Math.abs(m23)>opt_m) return false;
   else
   {
      h = idx3d_Vector.add(scene.light[0].v, scene.defaultCamera.lookat);
      q_1=p1.n.x*h.x+p1.n.y*h.y+p1.n.z*h.z;
      q_2=p2.n.x*h.x+p2.n.y*h.y+p2.n.z*h.z;
      q_3=p3.n.x*h.x+p3.n.y*h.y+p3.n.z*h.z;
      if(Math.abs(q_1-q_2)>opt_q ||
            Math.abs(q_1-q_3)>opt_q ||
            Math.abs(q_2-q_3)>opt_q) return false;
      else
      {
p_1=p1.n.x*scene.light[0].v.x+p1.n.y*scene.light[0].v.y+p1.n.z*scene.light[0].v.z;
p_2=p2.n.x*scene.light[0].v.x+p2.n.y*scene.light[0].v.y+p2.n.z*scene.light[0].v.z;
p_3=p3.n.x*scene.light[0].v.x+p3.n.y*scene.light[0].v.y+p3.n.z*scene.light[0].v.z;
            if(Math.abs(p_1-p_2)>opt_p ||
               Math.abs(p_1-p_3)>opt_p ||
               Math.abs(p_2-p_3)>opt_p) return false;
            else return true;}}
   //return true;}
public AdaptivGourauAndFong()
protected void initRender()
{
```

```
      bTypeRasterizer=selectMetod();
    if(bTypeRasterizer)
    {
     //for Gouraund
      h = idx3d_Vector.add(scene.light[0].v, scene.defaultCamera.lookat);
      int dy12 = Math.abs(p1.y-p2.y);
      int dy13 = Math.abs(p3.y-p1.y);
      int dy23 = Math.abs(p3.y-p2.y);
      i1 = calcColor(p1.n2);
      i2 = calcColor(p2.n2);
      i3 = calcColor(p3.n2);
      if (dy12 != 0)
      {
         ii12r = ((float)(idx3d_Color.getRed(i2) - idx3d_Color.getRed(i1))) /
(float)dy12;
         ii12g = ((float)(idx3d_Color.getGreen(i2) - idx3d_Color.getGreen(i1))) /
(float)dy12;
         ii12b = ((float)(idx3d_Color.getBlue(i2) - idx3d_Color.getBlue(i1))) /
(float)dy12;}
      if (dy13 != 0)
      {
         ii13r = ((float)(idx3d_Color.getRed(i3) - idx3d_Color.getRed(i1))) /
(float)dy13;
         ii13g = ((float)(idx3d_Color.getGreen(i3) - idx3d_Color.getGreen(i1))) /
(float)dy13;
         ii13b = ((float)(idx3d_Color.getBlue(i3) - idx3d_Color.getBlue(i1))) /
(float)dy13;}
      if (dy23 != 0)
      {
```

```
        ii23r = ((float)(idx3d_Color.getRed(i3) - idx3d_Color.getRed(i2))) /
(float)dy23;
        ii23g = ((float)(idx3d_Color.getGreen(i3) - idx3d_Color.getGreen(i2))) /
(float)dy23;
        ii23b = ((float)(idx3d_Color.getBlue(i3) - idx3d_Color.getBlue(i2))) /
(float)dy23;
      }
      i4r = (dy12*ii13r) + idx3d_Color.getRed(i1);
      i4g = (dy12*ii13g) + idx3d_Color.getGreen(i1);
      i4b = (dy12*ii13b) + idx3d_Color.getBlue(i1);
      //End for Gouraund}
    else
    {
      //for Fong
       h = idx3d_Vector.add(scene.light[0].v, scene.defaultCamera.lookat);
      int dx12 = Math.abs(p1.x-p2.x);
      int dx13 = Math.abs(p3.x-p1.x);
      int dx23 = Math.abs(p3.x-p2.x);
      int dy12 = Math.abs(p1.y - p2.y);
      int dy13 = Math.abs(p3.y - p1.y);
      int dy23 = Math.abs(p3.y - p2.y);
      vi13 = new idx3d_Vector((p3.n2.x - p1.n2.x) / dy13,
                              (p3.n2.y - p1.n2.y) / dy13,
                              (p3.n2.z - p1.n2.z) / dy13);
      vi12 = new idx3d_Vector((p2.n2.x - p1.n2.x) / dy12,
                              (p2.n2.y - p1.n2.y) / dy12,
                              (p2.n2.z - p1.n2.z) / dy12);
      vi23 = new idx3d_Vector((p3.n2.x - p2.n2.x) / dy23,
                              (p3.n2.y - p2.n2.y) / dy23,
                              (p3.n2.z - p2.n2.z) / dy23);
```

```
    v4 = idx3d_Vector.add(idx3d_Vector.scale(dy12, vi13), p1.n2);
    //End for Fong}}
protected void prerenderFirstPart()
{
 if(bTypeRasterizer)
 {
    //for Gouraund
    lir = idx3d_Color.getRed(i1);
    lig = idx3d_Color.getGreen(i1);
    lib = idx3d_Color.getBlue(i1);
    rir = idx3d_Color.getRed(i1);
    rig = idx3d_Color.getGreen(i1);
    rib = idx3d_Color.getBlue(i1);
    if (dx > 0)
    {
        liir = ii12r;
        liig = ii12g;
        liib = ii12b;
        riir = ii13r;
        riig = ii13g;
        riib = ii13b;}
    else
    {
        riir = ii12r;
        riig = ii12g;
        riib = ii12b;
        liir = ii13r;
        liig = ii13g;
        liib = ii13b;}
    //End for Gouraund}
```

```
   else
 {
   //for Fong
   lv = p1.n2;
   rv = p1.n2;
   if (dx > 0)
   {
      lvi = vi12;
      rvi = vi13;}
   else
   {
      lvi = vi13;
      rvi = vi12;}
   //End for Fong}}
protected void prerenderSecondPart()
{
 if(bTypeRasterizer)
 {
   //for Gouraund
   if (dx > 0)
   {
      lir = idx3d_Color.getRed(i2);
      lig = idx3d_Color.getGreen(i2);
      lib = idx3d_Color.getBlue(i2);
      rir = i4r;
      rig = i4g;
      rib = i4b;
      liir = ii23r;
      liig = ii23g;
      liib = ii23b;
```

```
        riir = ii13r;
        riig = ii13g;
        riib = ii13b;}
    else
    {
        rir = idx3d_Color.getRed(i2);
        rig = idx3d_Color.getGreen(i2);
        rib = idx3d_Color.getBlue(i2);
        lir = i4r;
        lig = i4g;
        lib = i4b;
        riir = ii23r;
        riig = ii23g;
        riib = ii23b;
        liir = ii13r;
        liig = ii13g;
        liib = ii13b;}
    //End for Gouraund}
else
{
    //for Fong
    if (dx > 0)
    {
        lv = p2.n2;
        rv = v4;
        lvi = vi23;
        rvi = vi13;}
    else
    {
        lv = v4;
```

```
        rv = p2.n2;
        lvi = vi13;
        rvi = vi23;}
    //End for Fong}}
protected void prerenderLine()
{
 if(bTypeRasterizer)
 {
    //for Gouraund
    renderLine();
    lir += liir;
    lig += liig;
    lib += liib;
    rir += riir;
    rig += riig;
    rib += riib;
    //End for Gouraund}
 else
 {
    //for Fong
    renderLine();
    lv = idx3d_Vector.add(lv, lvi);
    rv = idx3d_Vector.add(rv, rvi);
    //End for Fong}}
public String getRasterizerName()
{
    return "Adaptiv Gouraud And Fong";}
protected void renderLine()
{
 if(bTypeRasterizer)
```

```
{
  //for Gouraund
  int cdx = xR - xL;
  if (cdx == 0) return;
  float iir = 0;
  float iig = 0;
  float iib = 0;
  float cir = 0;
  float cig = 0;
  float cib = 0;
  float lastr = lir;
  float lastg = lig;
  float lastb = lib;
  boolean bReset = true;
  for (x=xL;x<xR;x++)
  {
     pos=x+offset;
     if (z<zBuffer[pos])
     {
        if (rir==255 && rig==255 && rib==255 )
        {
           return;
        }
        if (bReset)
        {
           cdx = xR - x;
           if ((xR - x)>1)
           {
              iir = (rir - lir) / cdx;
              iig = (rig - lig) / cdx;
```

```
                iib = (rib - lib) / cdx;
                bReset = false;}
            cir = lir;
            cig = lig;
            cib = lib;}
        screen.p[pos]=0xFF000000 | idx3d_Color.getColor(Math.round(cir > 255 ?
255 : cir), Math.round(cig > 255 ? 255 : cig), Math.round(cib > 255 ? 255: cib));
//          screen.p[pos]=0xFF000000 | idx3d_Color.getColor(255, 255, 255);
        zBuffer[pos]=z;
        if (useIdBuffer) idBuffer[pos]=currentId;
        cir += iir;
        cig += iig;
        cib += iib;}
      else
        bReset = true;}
    z += dz;
    //End for Gouraund}
  else
  {
    //for Fong
    int cdx = xR - xL;
    if (cdx==0)
      return;
    for (int x=xL; x<xR; x++)
    {
      pos=x+offset;
      if (z<zBuffer[pos])
      {
        idx3d_Vector cv = idx3d_Vector.GetVectorOnPos(lv, rv, xL, xR, x);
        float angle = idx3d_Vector.angle(scene.light[0].v, cv);
```

```
                if (angle < 0) angle = 0;
                c = idx3d_Color.scale(color, (int)(angle*255));
                angle = idx3d_Vector.angle(h, cv);
                if (angle < 0) angle = 0;
                angle = (float)Math.pow(angle, n);
                int c2 = idx3d_Color.scale(0xFFFFFF, (int)(angle*255));
                c = idx3d_Color.add(c, c2);
                screen.p[pos]=0xFF000000|c;
                zBuffer[pos]=z;
                if (useIdBuffer) idBuffer[pos]=currentId;}
            z+=dz;
            nx+=dnx;
            ny+=dny;}
        //End for Fong}}
    //for Gouraund
    private int calcColor(idx3d_Vector vector)
    {
        float angle = idx3d_Vector.angle(scene.light[0].v, vector);
        if (angle < 0) angle = 0;
        int c = idx3d_Color.scale(color, (int)(angle*255));
        angle = idx3d_Vector.angle(h, vector);
        if (angle < 0) angle = 0;
        angle = (float)Math.pow(angle, n);
        int c2 = idx3d_Color.scale(0xFFFFFF, (int)(angle*255));
        c = idx3d_Color.add(c, c2);
        return c;}
    //End for Gouraund
    protected void beforeRenderTriangle()
}
```

Лістинг модуля NormalVector.java

```
public class NormalVector extends idx3d_Rasterizer
{
   private idx3d_Vector h, v4;
   private idx3d_Vector vi13, vi12, vi23;
   private idx3d_Vector lvi, rvi;
   private idx3d_Vector lv, rv;
   int n_2;
   private idx3d_Vector N_2 ;
   int Nl,Nr, count_n;
   private idx3d_Vector dN_i,Na,Nb;
   private RET Ret[];
   public NormalVector()
   {
   }
   protected void initRender()
   {
      h = idx3d_Vector.add(scene.light[0].v, scene.defaultCamera.lookat);
      int dx12 = Math.abs(p1.x-p2.x);
      int dx13 = Math.abs(p3.x-p1.x);
      int dx23 = Math.abs(p3.x-p2.x);
      int dy12 = Math.abs(p1.y-p2.y);
      int dy13 = Math.abs(p3.y-p1.y);
      int dy23 = Math.abs(p3.y-p2.y);
      vi13 = new idx3d_Vector((p3.n2.x - p1.n2.x) / dy13,
                              (p3.n2.y - p1.n2.y) / dy13,
                              (p3.n2.z - p1.n2.z) / dy13);
      vi12 = new idx3d_Vector((p2.n2.x - p1.n2.x) / dy12,
                              (p2.n2.y - p1.n2.y) / dy12,
```

```
                    (p2.n2.z - p1.n2.z) / dy12);
    vi23 = new idx3d_Vector((p3.n2.x - p2.n2.x) / dy23,
                    (p3.n2.y - p2.n2.y) / dy23,
                    (p3.n2.z - p2.n2.z) / dy23);
    v4 = idx3d_Vector.add(idx3d_Vector.scale(dy12, vi13), p1.n2);
}
protected void prerenderFirstPart()
{
    lv = p1.n2;
    rv = p1.n2;
    if (dx > 0)
    {
        lvi = vi12;
        rvi = vi13;
    }
    else
    {
        lvi = vi13;
        rvi = vi12;
    }
}
protected void prerenderSecondPart()
{
    if (dx > 0)
    {
        lv = p2.n2;
        rv = v4;
        lvi = vi23;
        rvi = vi13;
    }
```

```
        else
        {
            lv = v4;
            rv = p2.n2;
            lvi = vi13;
            rvi = vi23;
        }
    }
    protected void prerenderLine()
    {
        norm();
        renderLine();
        lv = idx3d_Vector.add(lv, lvi); // приращение вектора по горизонтали в лево
        rv = idx3d_Vector.add(rv, rvi); // приращение вектора по горизонтали в
право
    }
     protected idx3d_Vector dN(int n, int k,idx3d_Vector N_p,idx3d_Vector N_g
)//для промежних ненормованых векторів
     {
      int n2=1, k2=1;
      n2<<=n;
      k2<<=k;
      idx3d_Vector Ngp= idx3d_Vector.sub(N_g,N_p);
      return new idx3d_Vector((n2*Ngp.x/k2),(n2*Ngp.y/k2),(n2*Ngp.z/k2));
     }
      protected void norm()
    {
        idx3d_Vector Nbuf12,N12;
        Na=lv; Nb=rv;
        int st=Math.abs(xR - xL);
```

```
for(n_2=1;n_2<st;)      // пошук числа 2 в степені к
{
  n_2<<=1;
}
if(st!=n_2)
{               //знаходження кінцевого вектора
  idx3d_Vector buf= idx3d_Vector.sub(Nb,Na);
  N_2=new idx3d_Vector((buf.x*n_2/st),(buf.y*n_2/st),(buf.z*n_2/st));
  N_2.normalize();
  Nb=N_2;
}
double cos_W = idx3d_Vector.angle(Na,Nb);
double z_12=Math.pow((2*(1+cos_W)),0.5);
Nbuf12 = idx3d_Vector.add(Na,Nb);
N12= new
idx3d_Vector((float)(Nbuf12.x/z_12),(float)(Nbuf12.y/z_12),(float)(Nbuf12.z/z_12)
);       N12.normalize();//середина
 double di=0;
double zi=z_12;
idx3d_Vector Ni=N12;
Nl=xL;
count_n=0;
Ret=new RET[n_2];
RET Wet= GetSegment( Na, Ni,Nb ,n_2/2,zi,n_2/2); //поиск интервалов
Ret[count_n]=Wet;
for(int k=0;k<count_n;k++)
{
  int min=n_2,point=0;
  for(int i=k;i<=count_n;i++)
  {
```

```
        int retx=Ret[i].GetX();
        if(min>retx)
        {
           min=retx;
           point=i;
        }
       }
       RET buf= Ret[k];
       Ret[k]=Ret[point];
       Ret[point]=buf;
     }
     if(count_n>2)
      Ret[count_n+1]= new RET(Nb,n_2);
   }
   public RET GetSegment(idx3d_Vector NA,idx3d_Vector Ni,idx3d_Vector NB ,
int n_2,double zi,int R)
   {
        double di=1-(Math.pow((2+zi),0.5)/2); //максимальная ошибка
        if(di<0.02 )
        {
         return new RET(Ni,R);
        }
        double zi1 = Math.pow((2+zi),0.5);
        idx3d_Vector  NLi= idx3d_Vector.add(NA,Ni);
        idx3d_Vector  Li=new
idx3d_Vector((float)(NLi.x/zi1),(float)(NLi.y/zi1),(float)(NLi.z/zi1));
        idx3d_Vector  NRi= idx3d_Vector.add(NB,Ni);
        idx3d_Vector  Ri=new
idx3d_Vector((float)(NRi.x/zi1),(float)(NRi.y/zi1),(float)(NRi.z/zi1));
        int dN=n_2/2;
```

```
     if(dN>=2)
      {
       Ret[count_n++]= GetSegment(NA, Li,Ni , dN, zi1,R-dN);//влево
       Ret[count_n++]=GetSegment(Ni, Ri,NB , dN, zi1,R+dN);//вправо}
  return new RET(Ni,R);}
public String getRasterizerName()
{
    return "NormalVector";
}
protected void renderLine()
{
    int cdx = Math.abs( xR - xL);
   if (cdx==0)
        return;
    int i=0,k=0,xi=0;
    idx3d_Vector Np=Na,Ng;
    idx3d_Vector dN,Ni;
    Ng=Ret[k].GetVector();
    dN= idx3d_Vector.sub(Ng,Np);
    dN.x*=n_2/cdx;
    dN.y*=n_2/cdx;
    dN.z*=n_2/cdx;
    for (int x=xL; x<xR; x++)
    {
     if(count_n>2)
       if(Ret[k].GetX()<xi )
       {
         Np=Ret[k].GetVector();
         Ng=Ret[k+1].GetVector();
         i=0;
```

```
 k++;
 dN= idx3d_Vector.sub(Ng,Np);
 dN.x*=n_2/cdx;
 dN.y*=n_2/cdx;
 dN.z*=n_2/cdx;
}
 dN.x*=i;
 dN.y*=i;
 dN.x*=i;
Ni=idx3d_Vector.add(Np,dN);
i++;
xi++;
pos=x+offset;
if (z<zBuffer[pos])
{
    idx3d_Vector cv=Ni;
   float angle = idx3d_Vector.angle(scene.light[0].v, cv);
   if (angle < 0) angle = 0;
   c = idx3d_Color.scale(color, (int)(angle*255));
   angle = idx3d_Vector.angle(h, cv);
   if (angle < 0) angle = 0;
   angle = (float)Math.pow(angle, n);
   int c2 = idx3d_Color.scale(0xFFFFFF, (int)(angle*255));
   c = idx3d_Color.add(c, c2);
   screen.p[pos]=0xFF000000|c;
   zBuffer[pos]=z;
   if (useIdBuffer) idBuffer[pos]=currentId;}
z+=dz;
nx+=dnx;
ny+=dny;}}protected void beforeRenderTriangle()}
```

Лістинг модуля GouraudRasterizer4.java

```
public class GouraudRasterizer4 extends idx3d_Rasterizer
{
   private idx3d_Vector h;
   int i1, i2, i3;
   float ii12r, ii12g, ii12b;
   float ii13r, ii13g, ii13b;
   float ii23r, ii23g, ii23b;
   float liir, liig, liib;
   float riir, riig, riib;
   float lir, lig, lib;
   float rir, rig, rib;
   float i4r, i4g, i4b;
   double Pr[];
   double Pg[];
   double Pb[];
   private idx3d_Vector vA, vB, vC,nA, nB, nC ;
   double X1,X2,X3;
   double Y1,Y2,Y3;
   double X12,X23,X13;
   double Y12,Y23,Y13;
   public GouraudRasterizer4()
   protected void initRender()
   {
      h = idx3d_Vector.add(scene.light[0].v, scene.defaultCamera.lookat);
      vA = new idx3d_Vector(p1.n2);
      vB = new idx3d_Vector(p2.n2);
      vC = new idx3d_Vector(p3.n2);
      double cos_1 = idx3d_Vector.angle(vA, h);//  cos(Na H)
```

```
double cos_2 = idx3d_Vector.angle(vB, h);//  cos(Nb H)
double cos_3 = idx3d_Vector.angle(vC, h);//  cos(Nc H)
double cos_4 = idx3d_Vector.angle(vA, vB);//  cos(Na Nb)
double cos_5 = idx3d_Vector.angle(vA, vC);//  cos(Na Nc)
double cos_6 = idx3d_Vector.angle(vC, vB);//  cos(Nc Nb)
double t1=(cos_1*cos_4 - cos_2)/((cos_4 - 1)*(cos_2 + cos_1)) ;
double t2=(cos_1*cos_5 - cos_3)/((cos_5 - 1)*(cos_3 + cos_1)) ;
double t3=(cos_2*cos_6 - cos_3)/((cos_6 - 1)*(cos_3 + cos_2)) ;
X1=Math.abs(p1.x+t1*(p2.x-p1.x));
X2=Math.abs(p1.x+t2*(p3.x-p1.x));
X3=Math.abs(p2.x+t3*(p3.x-p2.x));
Y1=Math.abs(p1.y+t1*(p2.y-p1.y));
Y2=Math.abs(p1.y+t2*(p3.y-p1.y));
Y3=Math.abs(p2.y+t3*(p3.y-p2.y));
 ////////////////////////////
 int   v1 = calcColor(p1.n2);
 int   v2 = calcColor(p2.n2);
 int   v3 = calcColor(p3.n2);
 ////////////////////////////
double Fr[][]={{0,0,0,0,0,1,idx3d_Color.getRed(v1)},
          {X2*X2,X2*Y2,Y2*Y2,X2,Y2,1,idx3d_Color.getRed(v2)},
{X2*X2/4,X2*Y2/4,Y2*Y2/4,X2/2,Y2/2,1,(idx3d_Color.getRed(v1)+idx3d_Co
lor.getRed(v2))/2},{X3*X3,X3*Y3,Y3*Y3,X3,Y3,1,idx3d_Color.getRed(v3)},
X3*X3/4,X3*Y3/4,Y3*Y3/4,X3/2,Y3/2,1,(idx3d_Color.getRed(v1)+idx3d_Col
or.getRed(v3))/2},{(X2+X3)*(X2+X3)/4,(X2+X3)*(Y2+Y3)/4,(Y2+Y3)*(Y2+
Y3)/4,(X2+X3)/2,(Y2+Y3)/2,1,(idx3d_Color.getRed(v2)+idx3d_Color.getRed(v
3))/2}};
Kramer sist= new Kramer();
Pr=new double[6];
Pr= sist.GetKramer(Fr,6,7);
```

```
        double Fg[][]={{0,0,0,0,0,1,idx3d_Color.getGreen(v1)},
                {X2*X2,X2*Y2,Y2*Y2,X2,Y2,1,idx3d_Color.getGreen(v2)},
        {X2*X2/4,X2*Y2/4,Y2*Y2/4,X2/2,Y2/2,1,(idx3d_Color.getGreen(v1)+idx3d_
        Color.getGreen(v2))/2},{X3*X3,X3*Y3,Y3*Y3,X3,Y3,1,idx3d_Color.getGree
        n(v3)},
        {X3*X3/4,X3*Y3/4,Y3*Y3/4,X3/2,Y3/2,1,(idx3d_Color.getGreen(v1)+idx3d_
        Color.getGreen(v3))/2},
        {(X2+X3)*(X2+X3)/4,(X2+X3)*(Y2+Y3)/4,(Y2+Y3)*(Y2+Y3)/4,(X2+X3)/2,(
        Y2+Y3)/2,1,(idx3d_Color.getGreen(v2)+idx3d_Color.getGreen(v3))/2}};
        Pg=new double[6];
        Pg= sist.GetKramer(Fg,6,7);
        double Fb[][]={{0,0,0,0,0,1,idx3d_Color.getBlue(v1)},
                {X2*X2,X2*Y2,Y2*Y2,X2,Y2,1,idx3d_Color.getBlue(v2)},
        {X2*X2/4,X2*Y2/4,Y2*Y2/4,X2/2,Y2/2,1,(idx3d_Color.getBlue(v1)+idx3d_C
        olor.getBlue(v2))/2},
                {X3*X3,X3*Y3,Y3*Y3,X3,Y3,1,idx3d_Color.getBlue(v3)},
{X3*X3/4,X3*Y3/4,Y3*Y3/4,X3/2,Y3/2,1,(idx3d_Color.getBlue(v1)+idx3d_Color.
        getBlue(v3))/2},
        {(X2+X3)*(X2+X3)/4,(X2+X3)*(Y2+Y3)/4,(Y2+Y3)*(Y2+Y3)/4,(X2+X3)/2,(
        Y2+Y3)/2,1,(idx3d_Color.getBlue(v2)+idx3d_Color.getBlue(v3))/2}};
        Pb=new double[6];
        Pb= sist.GetKramer(Fb,6,7);
    protected void prerenderFirstPart()
    protected void prerenderSecondPart()
    protected void prerenderLine(){
        renderLine();}
    public String getRasterizerName()
    {
        return " --------- ";}
    protected void renderLine()
```

```
{
   int cdx = xR - xL;
   if (cdx == 0) return;
   for (x=xL;x<xR;x++)
   {
      double Ixyr=Pr[5]*x*x+Pr[4]*x*y+Pr[3]*y*y+Pr[2]*x+Pr[1]*y+Pr[0];
      double Ixyg=Pg[5]*x*x+Pg[4]*x*y+Pg[3]*y*y+Pg[2]*x+Pg[1]*x+Pg[0];
      double Ixyb=Pb[5]*x*x+Pb[4]*x*y+Pb[3]*y*y+Pb[2]*x+Pb[1]*y+Pb[0];
    float   cir = (float)Ixyr;
    float   cig = (float)Ixyg;
    float   cib = (float)Ixyb;
      if(cir<0){ cir=0;}
      if(cig<0){ cig=0;}
      if(cib<0){ cib=0;}
      pos=x+offset;
      if (z<zBuffer[pos]){
         screen.p[pos]=0xFF000000 | idx3d_Color.getColor(Math.round(cir > 255 ?
   255 : cir), Math.round(cig > 255 ? 255 : cig), Math.round(cib > 255 ? 255: cib));
         zBuffer[pos]=z;
         if (useIdBuffer) idBuffer[pos]=currentId;}}}
private int calcColor(idx3d_Vector vector)
{float angle = idx3d_Vector.angle(scene.light[0].v, vector);
   if (angle < 0) angle = 0;
   int c = idx3d_Color.scale(color, (int)(angle*255));
   angle = idx3d_Vector.angle(h, vector);
   if (angle < 0) angle = 0;
   angle = (float)Math.pow(angle, n);
   int c2 = idx3d_Color.scale(0xFFFFFF, (int)(angle*255));
   c = idx3d_Color.add(c, c2);
   return c;}protected void beforeRenderTriangle()}
```

ДОДАТОК Г.

Ілюстративна частина

МЕТОДИ ТА ЗАСОБИ ВИСОКОПРОДУКТИВНОГО ФОРМУВАННЯ ТРИВИМІРНИХ ГРАФІЧНИХ ЗОБРАЖЕНЬ

Методи та засоби високопродуктивного формування тривимірних графічних зображень

Студентка гр. 2ПІ-20м Яковенко О. О.

Науковий керівник - д.т.н. проф.

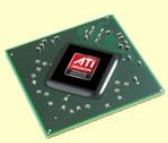
Романюк О. Н.

Вінниця - 2021

1

Рисунок Г.1 – Слайд презентації №1

Галузі застосування та перспективи розвитку графічних засобів



У 2019 році реалізовано понад 450 млн. графічних процесорів. У 2020 планується 544 млн.

У 2020 року світовий ринок смартфонів зросте до 759 млн. шт. У найближчі 5 років сумарний продаж ігор для мобільних телефонів збільшиться у 18 разів, а прибуток від їх реалізації досягне 17,5 млрд. дол.

У минулому році продано 35 млн. кишенькових комп'ютерів. До 2021 р. щорічний продаж нетбуків зросте до 139 млн. одиниць

У 2019 році реалізовано понад 3 млн. графічних станцій

Оборот у галузі ігрових консолей склав у 2019 році 17,94 млрд. дол

Відеокарти

У 2019 році у світі реалізовано 302 млн. ноутбуків, зокрема, в Україні – 720 тис. шт.

Архітектури AMD Fusion, Larrabee

У бортових системах прогнозується широке використання комп'ютерної графіки

2

Рисунок Г.2 – Слайд презентації №2

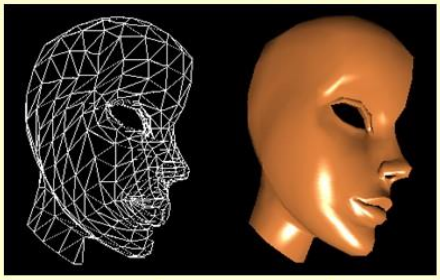
Мета і завдання дослідження

- **Мета і завдання дослідження.** Метою роботи є підвищення продуктивності та реалістичності формування тривимірних графічних зображень за рахунок розробки нових методів і засобів.
- **Об'єкт дослідження –** процес зафарбовування тривимірних графічних об'єктів.
- **Предмет дослідження –** методи і засоби зафарбовування тривимірних графічних об'єктів.
- **Задачі дослідження:**
- Проаналізувати методи та засоби зафарбовування тривимірних об'єктів.
- Розробити методи прискореного визначення нормалізованих векторів для задач рендерингу
- Розробити методи прискореного зафарбовування тривимірних об'єктів.
- Розробити програмні засоби для прискореного зафарбовування тривимірних об'єктів.
- Провести тестування розроблених методів з метою отримання порівняльних оцінок.

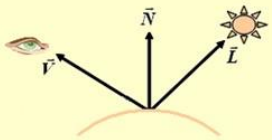
3

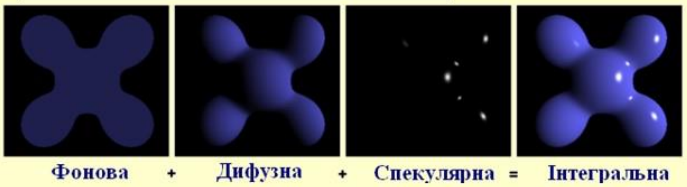
Рисунок Г.3 – Слайд презентації №3

Складові інтенсивності кольору

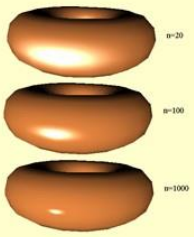


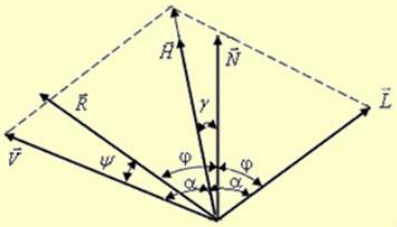
$$I = k_a I_a + I_i (k_d (\vec{N} \cdot \vec{L}) + k_s (\vec{N} \cdot \vec{H})^n)$$





Фонова + Дифузна + Спекулярна = Інтегральна





Вектори до точки поверхні

$(\vec{V} \cdot \vec{R})^n = \cos^n \psi$ - *ДФВЗ Фонга,*

$(\vec{N} \cdot \vec{H})^n = \cos^n \gamma$ - *ДФВЗ Бліна,*

$$\vec{H} = \frac{\vec{L} + \vec{V}}{|\vec{L} + \vec{V}|}, \quad n \in [1,1000].$$

$$1 - (\vec{N} \cdot \vec{V})^2 - (\vec{L} \cdot \vec{N})^2 - (\vec{L} \cdot \vec{V})^2 + 2(\vec{L} \cdot \vec{N}) \cdot (\vec{N} \cdot \vec{V}) \cdot (\vec{L} \cdot \vec{V}) = (\vec{L} \times \vec{N} \cdot \vec{V})^2$$

4

Рисунок Г.4 – Слайд презентації №4

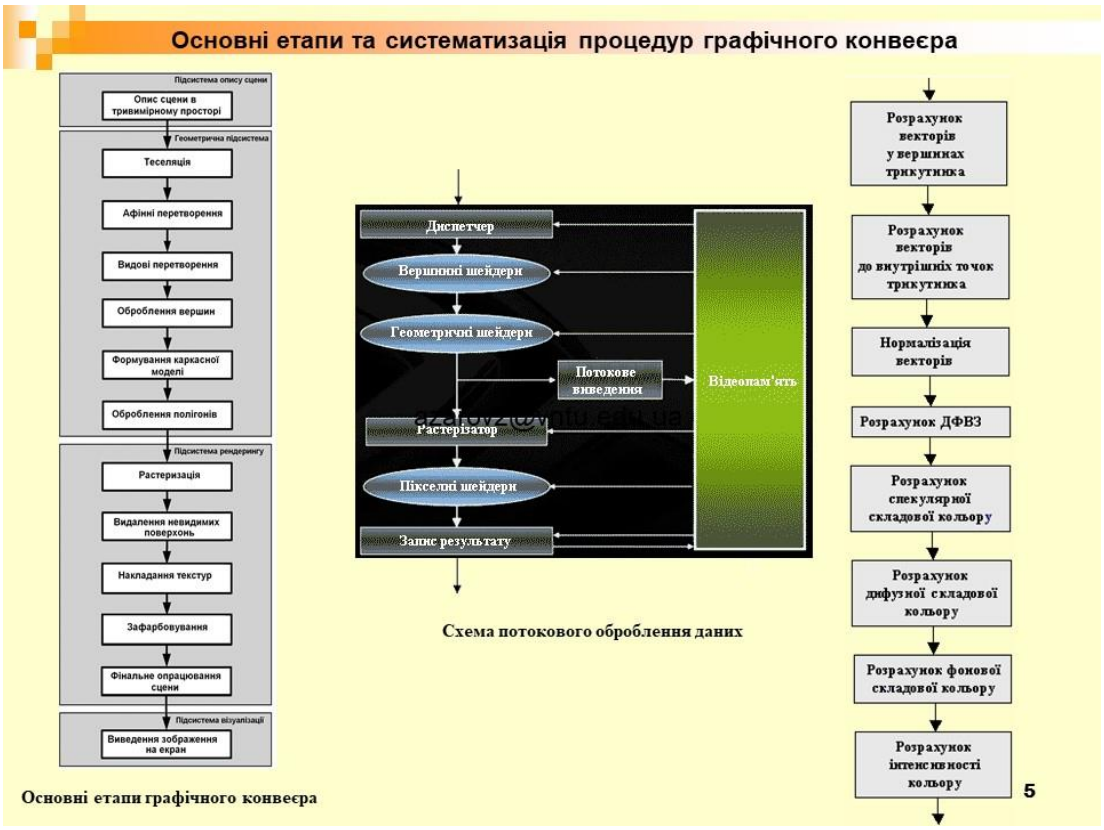


Рисунок Г.5 – Слайд презентації №5

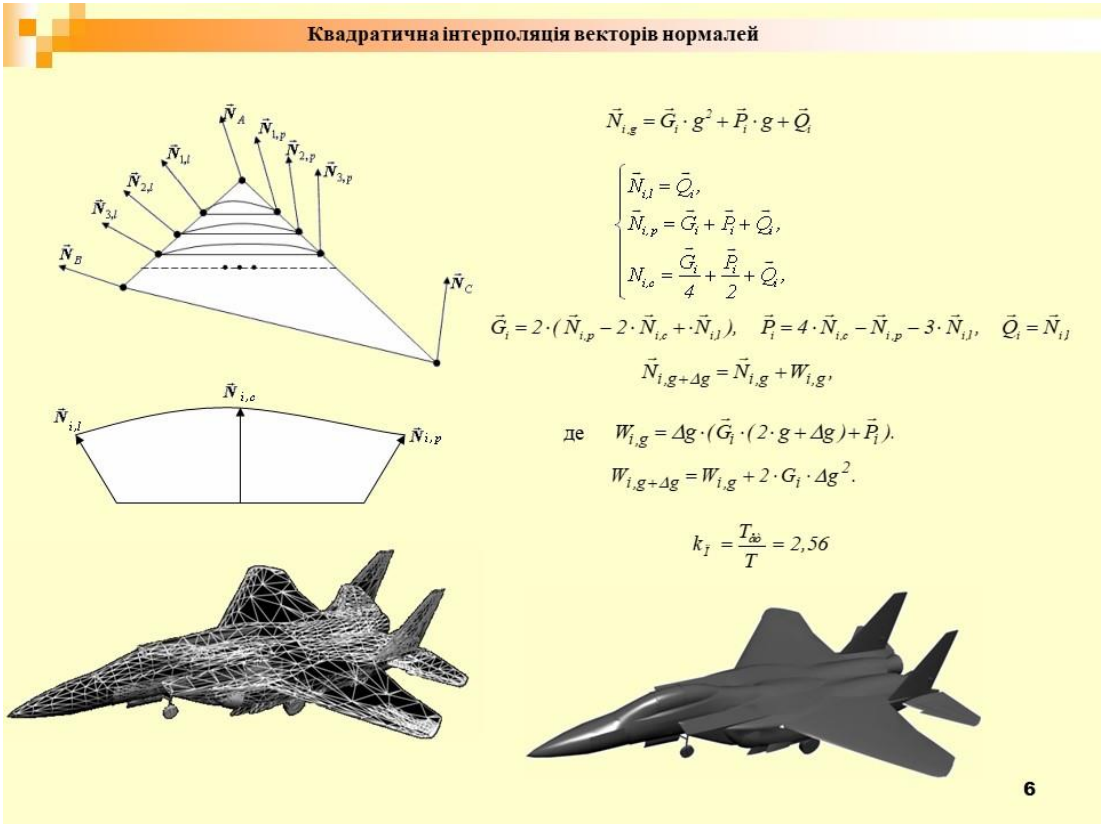
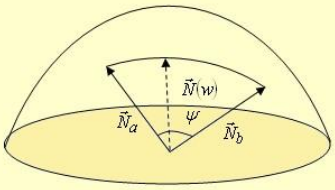


Рисунок Г.6 – Слайд презентації №6

Метод підвищення продуктивності зафарбовування з використанням сферично-кутової інтерполяції векторів нормалей



$$\vec{N}(w) = \vec{N}_a \frac{\sin((1-w)\psi)}{\sin\psi} + \vec{N}_b \frac{\sin(w\psi)}{\sin\psi},$$

$$\vec{N}(w) = \vec{N}_a \frac{\sin\psi\cos(w\psi) - \cos\psi\sin(w\psi)}{\sin\psi} + \vec{N}_b \frac{\sin(w\psi)}{\sin\psi}.$$

$$\vec{N}(w) = \vec{N}_a \cos(w\psi) + \frac{\vec{N}_b - \vec{N}_a \cdot (\vec{N}_b \cdot \vec{N}_a)}{\sqrt{1-(\vec{N}_b \cdot \vec{N}_a)^2}} \sin(w\psi),$$

$$\vec{N}(w) = \vec{N}_a \cos(w\psi) + \vec{N}_k \sin(w\psi). \qquad \varphi = \frac{\psi}{m}.$$

$$\vec{N}(t) = \vec{N}_a \cos(t\varphi) + \vec{N}_k \sin(t\varphi),$$

де $t \in [0, m]$ - позиція піксела в рядку растеризації, $\psi = \arccos(\vec{N}_a \cdot \vec{N}_b)$.

Рекурентне визначення вектору нормалі до потокової точки

$\vec{N}(t+1) = 2\vec{N}(t)\cos\varphi - \vec{N}(t-1).$ $\quad \vec{N}(t+2) = \vec{N}(t)\cdot\left(4\cos^2\varphi - 1\right) - 2\vec{N}(t-1)\cdot\cos\varphi.$

$\vec{N}(t+3) = \cos 2\varphi \cdot \left(4\vec{N}(t)\cdot\cos\varphi - \vec{N}(t-1)\right) - 3\vec{N}(t-1).$

$\vec{N}(t+4) = 2\vec{N}(t)\cdot\cos 2\varphi\left(2\cos 2\varphi + 3\right) - 2\vec{N}(t-1)\cos\varphi\left(\cos 2\varphi + 4\right) + \vec{N}(t)$

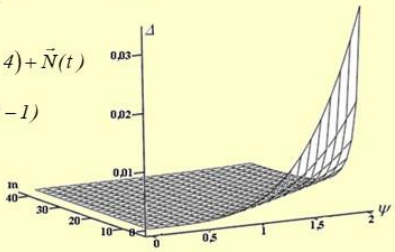
$\cos\left(\frac{\psi}{m}\right) \approx \frac{\cos\psi - 1}{m^2} + 1.$ $\quad \vec{N}(t+1) = 2\vec{N}(t)\cdot\left(\frac{\cos\psi - 1}{m^2} + 1\right) - \vec{N}(t-1)$

$$I(t+1)_d = 2I(t)_d \cos\varphi - I(t-1)_d,$$

де $I(t)_d = \vec{L}_d \cdot \vec{N}(t)$, $\vec{L}_d = I_l k_d \vec{L}$,

$I(t-1)_d = \vec{L}_d \cdot \vec{N}(t-1).$

$$\vec{N}(t+1)\cdot\vec{H} = 2\left(\vec{N}(t)\cdot\vec{H}\right)\cos\varphi - \vec{N}(t-1)\cdot\vec{H}.$$

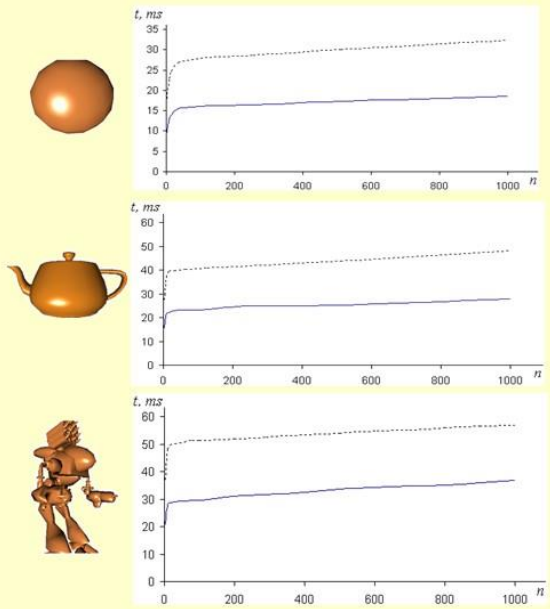


Залежність абсолютної похибки апроксимації від ψ і m

7

Рисунок Г.7 – Слайд презентації №7

Метод підвищення продуктивності зафарбовування з використанням сферично-кутової інтерполяції векторів нормалей



Час зафарбовування тестових фігур з використанням сферично-кутової інтерполяції векторів

8

Рисунок Г.8 – Слайд презентації №8

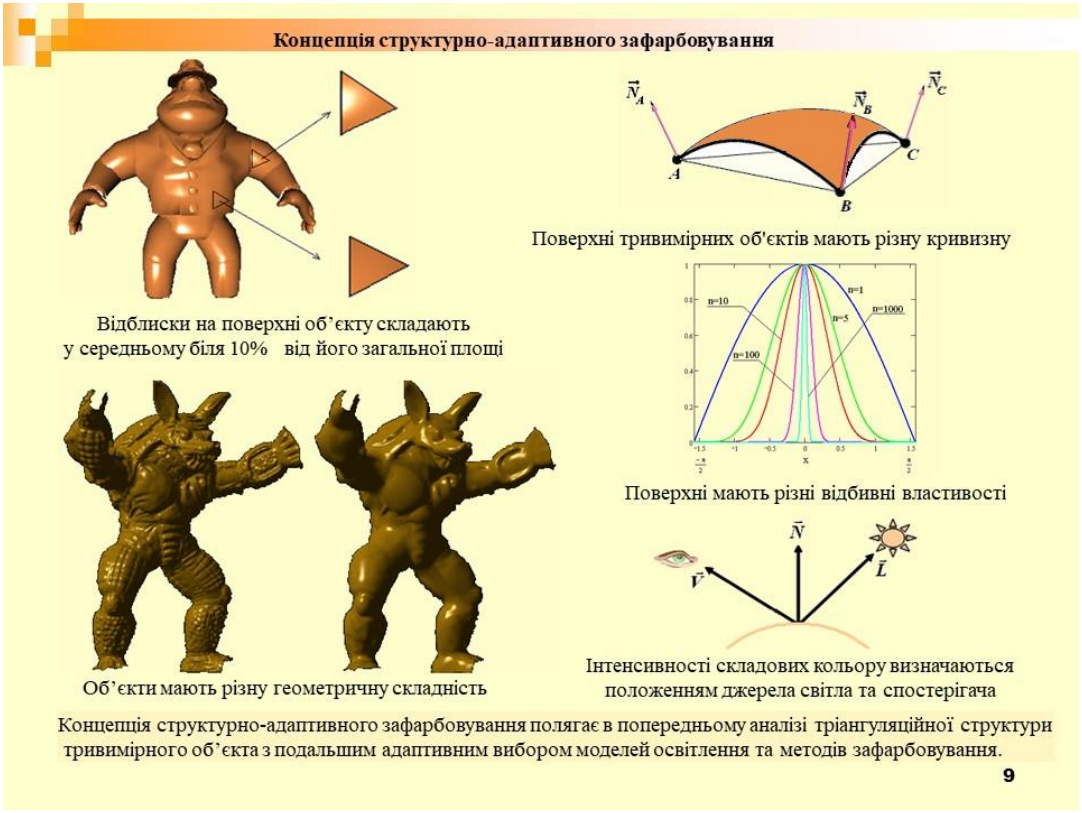


Рисунок Г.9 – Слайд презентації №9

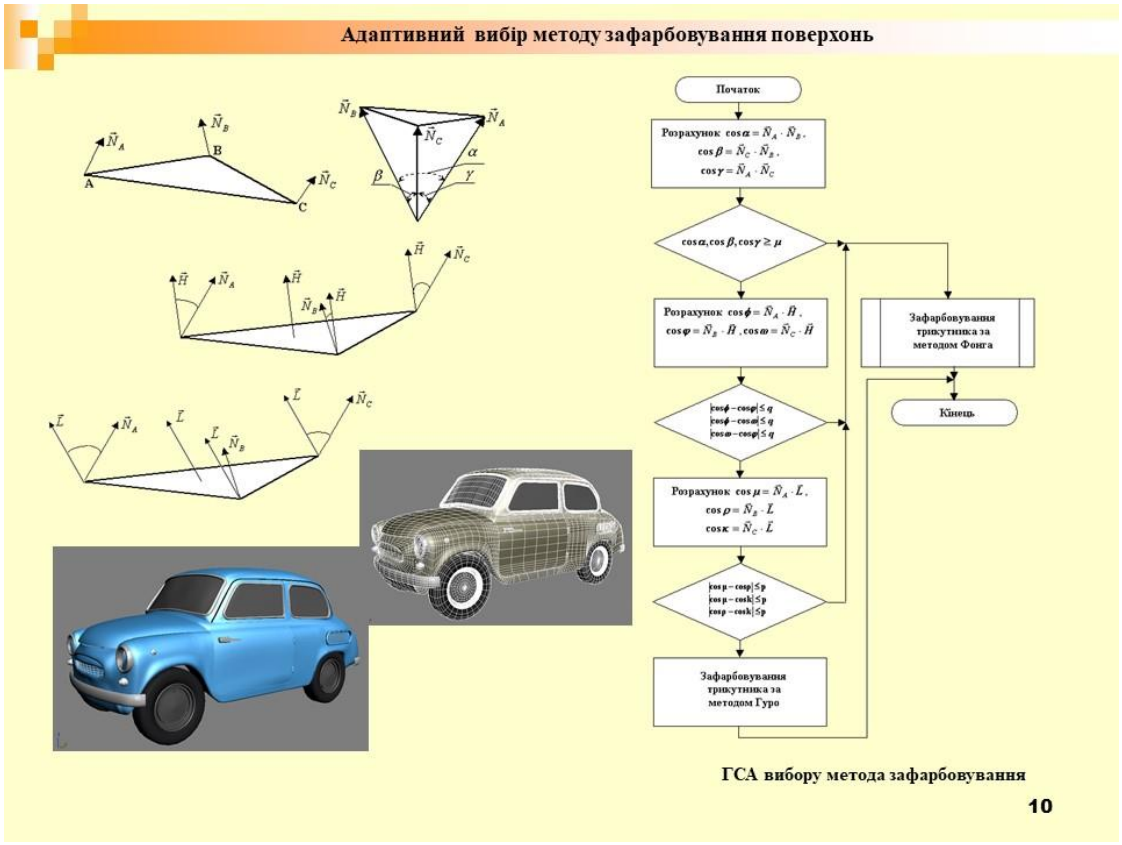


Рисунок Г.10 – Слайд презентації №10

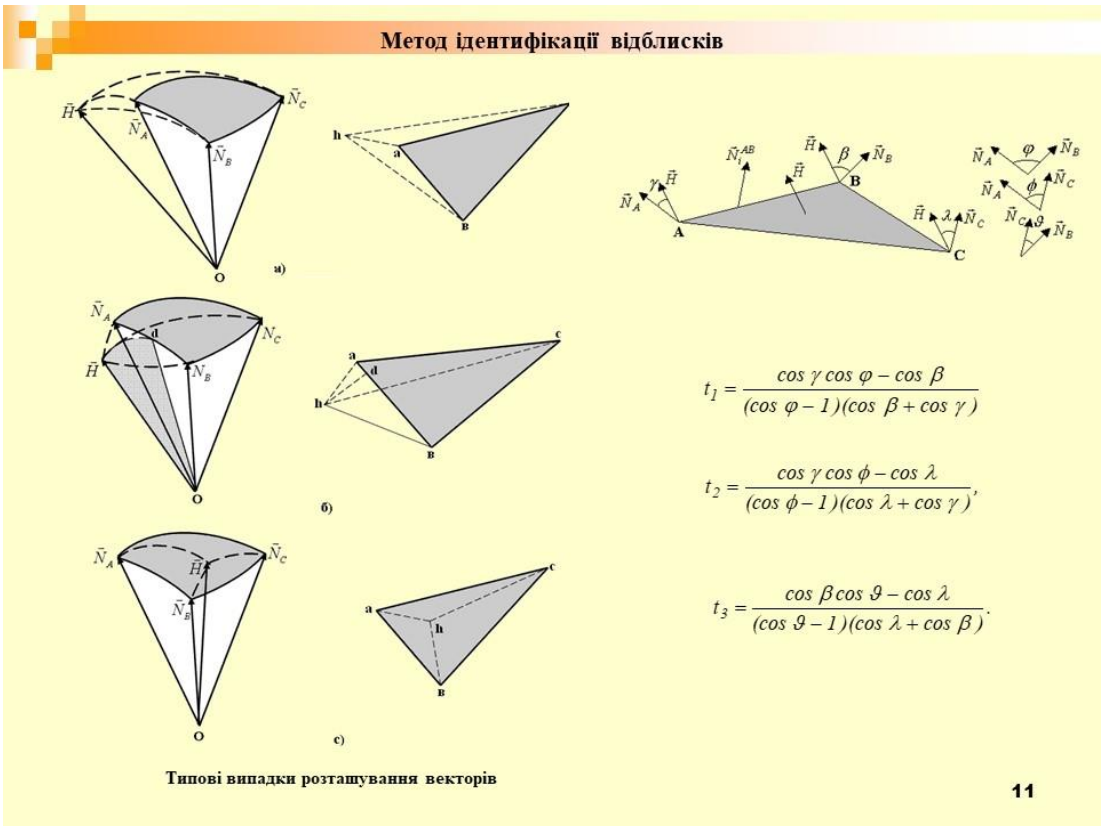


Рисунок Г.11 – Слайд презентації №11

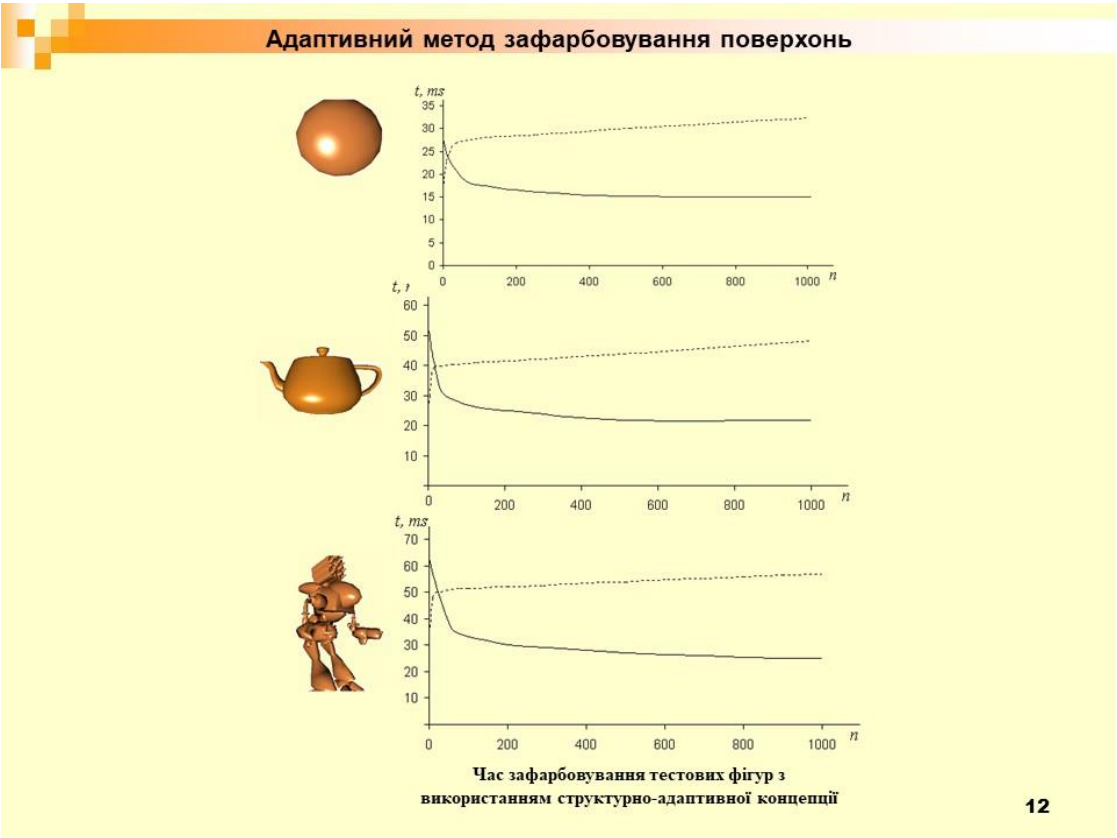


Рисунок Г.12 – Слайд презентації №12

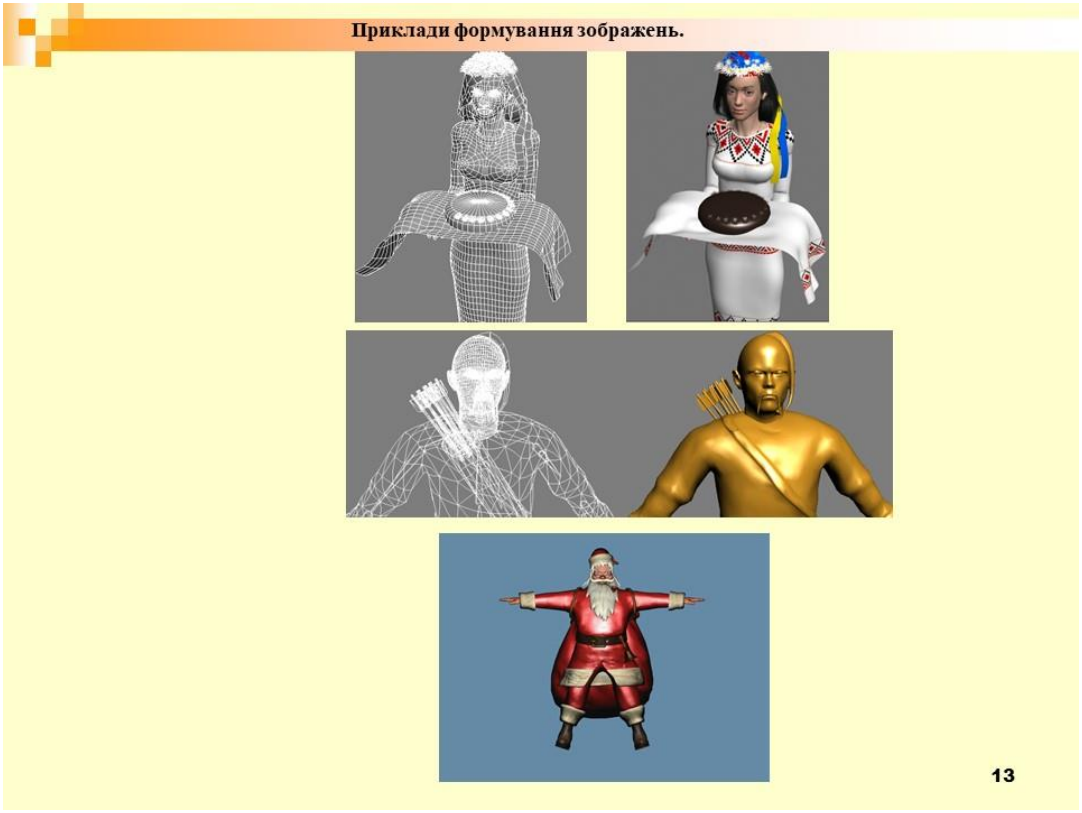


Рисунок Г.13 – Слайд презентації №13

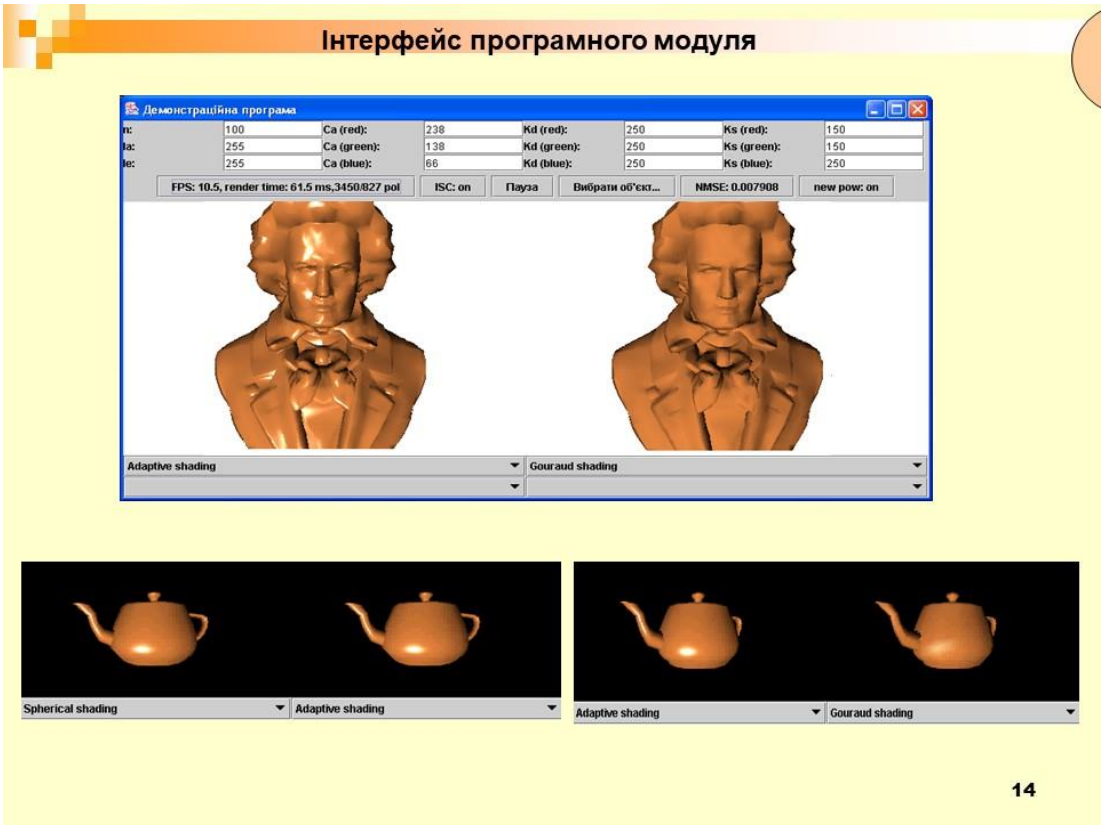


Рисунок Г.14 – Слайд презентації №14

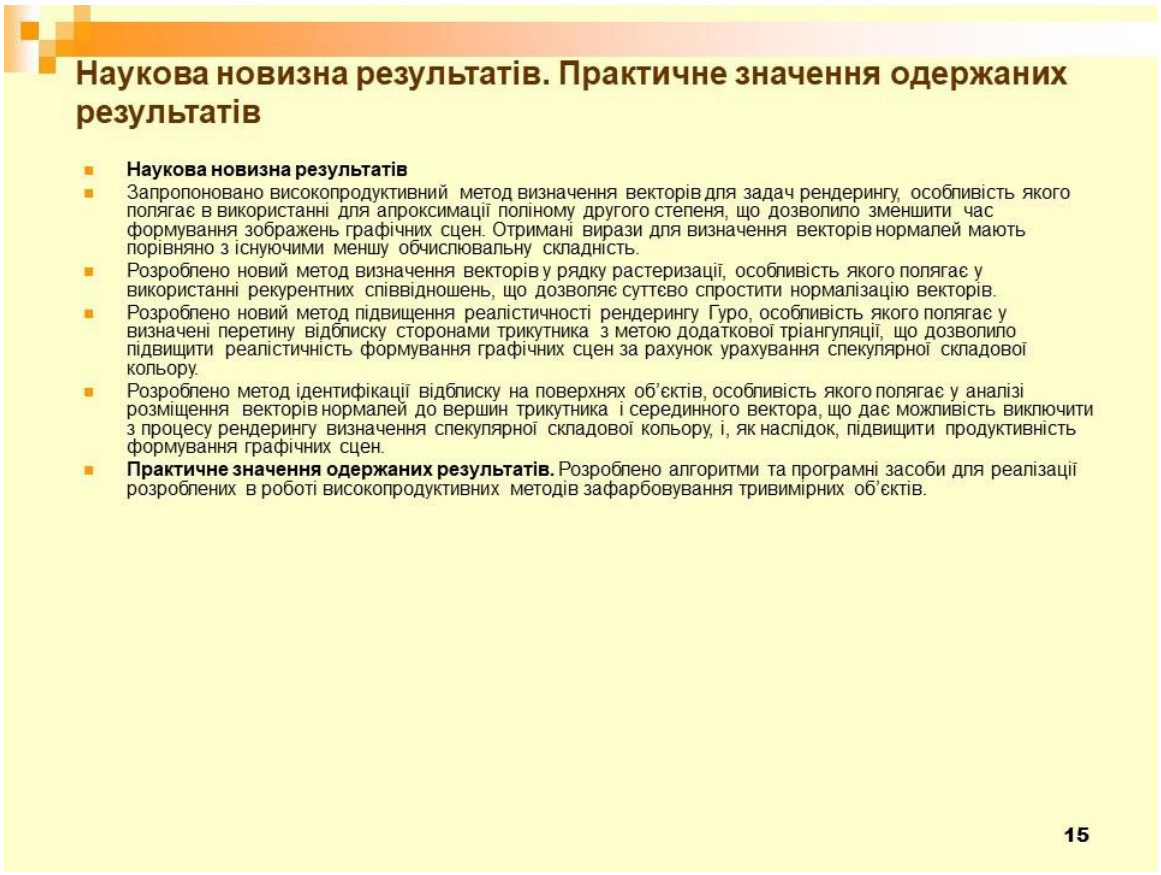


Рисунок Г.15 – Слайд презентації №15

ДОДАТОК Ґ.

Свідоцтва про реєстрацію авторського права


УКРАЇНА

СВІДОЦТВО

про реєстрацію авторського права на твір

№ 91843

Комп'ютерна програма "Комп'ютерна програма для рендерингу поверхонь тривимірних фігур"

(вид, назва твору)

Автор(и) Романюк Олександр Никифорович, Дудник Олег Олександрович, Романюк Оксана Володимирівна, Яковенко Олеся Олегівна

(повне ім'я, псевдонім (за наявності))

Дата реєстрації 28.08.2019

МІНІСТЕРСТВО ЕКОНОМІЧНОГО РОЗВИТКУ І ТОРГІВЛІ УКРАЇНИ ІДЕНТИФІКАЦІЙНИЙ КОД 37508596

Державний секретар Міністерства економічного розвитку і торгівлі України О. Ю. Перевезенцев

Рисунок Ґ.1 – Свідоцтво про реєстрацію авторського права на твір №91843

УКРАЇНА

СВІДОЦТВО

про реєстрацію авторського права на твір

№ 91846

Комп'ютерна програма "Комп'ютерна програма для зафарбовування поверхонь тривимірних об'єктів за модифікованим методом Гуро"

(вид, назва твору)

Автор(и) **Романюк Олександр Никифорович, Мальований Олександр Григорович, Романюк Оксана Володимирівна, Яковенко Олеся Олегівна**

(повне ім'я, псевдонім (за наявності))

Дата реєстрації 28.08.2019

МІНІСТЕРСТВО ЕКОНОМІЧНОГО РОЗВИТКУ І ТОРГІВЛІ УКРАЇНИ ІДЕНТИФІКАЦІЙНИЙ КОД 37508596

М.П.

Державний секретар Міністерства економічного розвитку і торгівлі України О. Ю. Перевезенцев

Рисунок Ґ.2 – Свідоцтво про реєстрацію авторського права на твір №91846

Рисунок Ґ.3 – Свідоцтво про реєстрацію авторського права на твір №91847

УКРАЇНА

СВІДОЦТВО

про реєстрацію авторського права на твір

№ 91848

Комп'ютерна програма "Комп'ютерна програма для зафарбовування поверхонь тривимірних об'єктів за модифікованим методом Фонга"

(вид, назва твору)

Автор(и) **Романюк Олександр Никифорович, Мальований Олександр Григорович, Романюк Оксана Володимирівна, Яковенко Олеся Олегівна**

(повне ім'я, псевдонім (за наявності))

Дата реєстрації 28.08.2019

МІНІСТЕРСТВО ЕКОНОМІЧНОГО РОЗВИТКУ І ТОРГІВЛІ УКРАЇНИ ІДЕНТИФІКАЦІЙНИЙ КОД 37508596

М.П.

Державний секретар Міністерства економічного розвитку і торгівлі України О. Ю. Перевезенцев

Рисунок Ґ.4 – Свідоцтво про реєстрацію авторського права на твір №91848

УКРАЇНА

СВІДОЦТВО

про реєстрацію авторського права на твір

№ 91849

Комп'ютерна програма "Комп'ютерна програма для розрахунку дистрибутивної функції відбивної здатності поверхні для задач рендерингу"

(вид, назва твору)

Автор(и) **Романюк Олександр Никифорович, Савицька Людмила Анатоліївна, Романюк Оксана Володимирівна, Яковенко Олеся Олегівна**

(повне ім'я, псевдонім (за наявності))

Дата реєстрації 28.08.2019

МІНІСТЕРСТВО ЕКОНОМІЧНОГО РОЗВИТКУ І ТОРГІВЛІ УКРАЇНИ ІДЕНТИФІКАЦІЙНИЙ КОД 37508536

М.П.

Державний секретар Міністерства економічного розвитку і торгівлі України О. Ю. Перевезенцев

Рисунок Г.5 – Свідоцтво про реєстрацію авторського права на твір №91849

УКРАЇНА

СВІДОЦТВО

про реєстрацію авторського права на твір

№ 91850

Комп'ютерна програма "Комп'ютерна програма для розрахунку спекулярної складової кольору з використанням нової моделі відбивної здатності поверхні"

(вид, назва твору)

Автор(и) **Романюк Олександр Никифорович, Савицька Людмила Анатоліївна, Романюк Оксана Володимирівна, Яковенко Олеся Олегівна**

(повне ім'я, псевдонім (за наявності))

Дата реєстрації 28.08.2019

МІНІСТЕРСТВО ЕКОНОМІЧНОГО РОЗВИТКУ І ТОРГІВЛІ УКРАЇНИ ІДЕНТИФІКАЦІЙНИЙ КОД 37505896

М.П.

Державний секретар Міністерства економічного розвитку і торгівлі України О. Ю. Перевезенцев

Рисунок Ґ.6 – Свідоцтво про реєстрацію авторського права на твір №91850

УКРАЇНА

СВІДОЦТВО

про реєстрацію авторського права на твір

№ 91957

Комп'ютерна програма "Комп'ютерна програма для текстурування з урахуванням перехресного співвідношення чотирьох колінеарних точок"

(вид, назва твору)

Автор(и) Романюк Олександр Никифорович, Денисюк Алла Василівна, Кирилащук Світлана Анатоліївна, Яковенко Олеся Олегівна

(повне ім'я, псевдонім (за наявності))

Дата реєстрації 28.08.2019

МІНІСТЕРСТВО ЕКОНОМІЧНОГО РОЗВИТКУ І ТОРГІВЛІ УКРАЇНИ ІДЕНТИФІКАЦІЙНИЙ КОД 37505396

Державний секретар Міністерства економічного розвитку і торгівлі України О. Ю. Перевезенцев

ПК «Україна». Зам. 19-2001. 2019 р. 1 кн.

Рисунок Ґ.7 – Свідоцтво про реєстрацію авторського права на твір №91957