

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра програмного забезпечення

Пояснювальна записка

до магістерської кваліфікаційної роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему: Розробка методу та програмних засобів для розпаралелення

рендерингу Гуро

Виконав: студент II курсу

групи 2ПІ-18м

спеціальності

121 – Інженерія програмного забезпечення

(шифр і назва напрямку підготовки, спеціальності)

Мальований Олександр Григорович

(прізвище та ініціали)

Керівник: завідувач кафедри ПЗ, д.т.н.,

проф. Романюк Олександр Никифорович

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Освітньо-кваліфікаційний рівень – магістр
Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
Романюк О. Н.
“ ___ ” _____ 2019 року

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

1. Тема роботи – Розробка методу та програмних засобів для розпаралелення рендерингу Гуро.

Керівник роботи: Романюк Олександр Никифорович, д.т.н., завідувач кафедри ПЗ, затверджені наказом вищого навчального закладу від “ ___ ” _____ 2019 року № ___

2. Строк подання студентом роботи – 3 січня 2019 р.

3. Вихідні дані до роботи: базовий метод зафарбовування – метод Гуро; модель освітлення – модель Бліна; графічний режим – TrueColor; вихідні дані для зафарбовування – координати вершин трикутників, інтенсивності кольору у вершинах трикутника; максимальна розрядність для задання адрес вершин трикутників – 12; вихідні дані – кінцеве зображення, зафарбоване згідно методу Гуро в поєднанні з моделлю освітлення Бліна.

4. Зміст розрахунково-пояснювальної записки: аналіз методів зафарбовування, визначення параметрів кодової інтерполяції при рендерингу Гуро, визначення максимальної похибки розрахунку інтенсивності внутрішніх точок трикутника, розробка методів розпаралелення рендерингу Гуро; розробка програмних засобів для зафарбовування; економічний розділ.

5. Перелік графічного матеріалу: галузі застосування графічних засобів; основні етапи графічного конвеєру; метод розпаралелення рендерингу Гуро; програмні засоби для рендерингу Гуро; інтерфейс програмного модуля.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Романюк О. Н., д.т.н., завідувач кафедри ПЗ		
5	Бальзан М.В., к.е.н., доцент кафедри ЕПВМ		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної Роботи	Строк виконання етапів роботи	Примітка
1	Аналіз моделей рендерингу	07.10.2019 – 27.10.2019	Вик.
2	Виведення математичних залежностей	28.10.2019 – 4.11.2019	Вик.
3	Розробка методів розпаралелення рендерингу Гуро	5.11.2019 – 15.11.2019	Вик.
4	Розробка програмних засобів	16.11.2019 – 30.11.2019	Вик.
5	Економічна частина	4.11.2019 – 14.11.2019	Вик.

Студент _____

(підпис)

Мальований О. Г.

(прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи _____

(підпис)

Романюк О. Н.

(прізвище та ініціали)

АНОТАЦІЯ

У магістерській кваліфікаційній роботі проведено детальний аналіз методів і засобів рендерингу графічних сцен. Сформульовано мету досліджень - підвищення продуктивності зафарбовування тривимірних зображень за рахунок розпаралелення рендерингу Гуро.

Уперше встановлено залежність приросту інтенсивностей кольору для точок поверхні, обмеженої трикутником, що дає можливість підвищити продуктивність зафарбовування та спростити апаратну реалізацію пристроїв зафарбовування. Подальшого розвитку отримав метод зафарбовування Гуро, який відрізняється від відомого визначення в кожному такті не однієї, а відразу кількох інтенсивностей точок трикутника, що дозволяє підвищити продуктивність формування графічних сцен.

На основі отриманих в магістерській кваліфікаційній роботі теоретичних положень запропоновано алгоритми та розроблено програмні засоби зафарбовування у комп'ютерних системах візуалізації тривимірних зображень. Розроблено структурні схеми рендерів.

Отримані в магістерській кваліфікаційній роботі наукові та практичні результати можна використати для побудови високопродуктивних систем рендерингу.

ABSTRACT

In the master's qualification work, a detailed analysis of methods and means of rendering graphic scenes was carried out. The purpose of the research was formulated to improve the productivity of coloring of three-dimensional images by paralleling the rendering of Gouraud.

For the first time, the dependence of the increase of color intensities for surface points bounded by a triangle is established, which allows to increase the coloring performance and to simplify the hardware implementation of the coloring devices. Further development was the method of painting Gouraud., which differs from the known definition in each cycle not one but several intensities of points of a triangle, which allows to increase the productivity of forming graphic scenes.

Based on the theoretical qualifications obtained in the master's thesis, algorithms have been proposed and software for painting in computer-aided 3D imaging systems has been developed. Structural schemes of renders have been developed.

The scientific and practical results obtained in the master's qualification work can be used to build high-performance rendering systems.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ МЕТОДІВ ФОРМУВАННЯ ТРИВИМІРНИХ ЗОБРАЖЕНЬ.....	11
1.1 Аналіз методів рендерингу.....	11
1.2 Особливості реалізації рендерингу Гуро.....	15
1.3 Висновки.....	18
2 ДОСЛІДЖЕННЯ РЕНДЕРИНГУ ГУРО.....	19
2.1 Визначення параметрів кодової інтерполяції при рендерингу Гуро.....	19
2.2 Визначення максимальної похибки розрахунку інтенсивності внутрішніх точок трикутника.....	23
2.3 Висновки.....	31
3 РОЗРОБКА МЕТОДІВ РОЗПАРАЛЕЛЕННЯ РЕНДЕРИНГУ ГУРО.....	32
3.1 Розробка методів розпаралелення.....	32
3.2 Розробка апаратних засобів розпаралелення.....	34
3.3 Паралельний рендеринг з використанням додаткової тріангуляції.....	39
3.4 Висновки.....	44
4 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ЗАФАРБОВУВАННЯ.....	45
4.1 Інтерфейс програмного модуля для високопродуктивного зафарбовування Гуро.....	45
4.2 Програми для високопродуктивного зафарбовування.....	47
4.3 Розробка загального алгоритму.....	56
4.4 Висновки.....	58
5 ЕКОНОМІЧНА ЧАСТИНА.....	59
5.1 Технологічний аудит розроблених методу та програмних засобів для розпаралелення рендерингу Гуро.....	59
5.2 Розрахунок витрат на виконання роботи.....	65
5.3 Розрахунок економічного ефекту від можливої комерціалізації.....	68
5.4 Висновки.....	72
ВИСНОВКИ.....	73
ПЕРЕЛІК ПОСИЛАНЬ.....	74
Додаток А. Технічне завдання.....	78
Додаток Б. Ілюстративний матеріал.....	82
Додаток В. Модель взаємозв'язку програмних модулів.....	83
Додаток Г. Час формування тестових фігур з використання методу Гуро.....	84
Додаток Д. Модуль “Gouraud Rasterizer”. <i>java</i>	85
Додаток Е. Лістинг головного модуля програми.....	92
Додаток Є. Графічний матеріал.....	101

ВСТУП

Розвиток обчислювальної техніки обумовлює широке використання програмних та апаратних засобів машинної графіки [1-3,5,6,12,15] графічний спосіб подання результатів одержав найбільше поширення завдяки тому, що він найбільш інформативний і та легкий у сприйнятті даних людиною. Засоби машинної графіки використовуються практично у всіх галузях народного господарства. Для узагальнення інформації, яка поступає у великих обсягах і з великою швидкістю, графічні зображення просто необхідні, тому що дозволяють стежити за змінами в стані об'єктів у масштабі реального часу.

Тривимірні зображення [22-24] на відміну від двовимірних мають значно більшу реалістичність, бо вони створюють ефект об'ємності простору, дозволяють оцінити конструктивні та естетичні переваги об'єктів.

Роль ефективних засобів наочного представлення об'ємних предметів на екранах ЕОМ безупинно зростає. Це пояснюється тим, що тривимірні (3D) зображення є більш ємними, компактними і наочними засобами представлення інформації.

Серед методів формування тривимірних зображень найбільше поширення одержали каркасний, контурний і напівтоновий. Каркасні (дротові) зображення складаються з ліній відтвореного об'єкта, які є видимі і невидимі спостерігачу. Об'ємність зображення основана на ефекті перспективи, що створює ілюзію глибини зображення. Недоліком каркасних зображень є прозорість об'єктів, тому що відтворюються всі лінії предметів і це призводить до втрати наочності.

Контурні зображення складаються тільки з видимих спостерігачу ліній і поверхонь. Видалення невидимих ліній потребує збільшення обсягу обчислень, що зростають по мірі збільшення складності і кількості відтворених об'єктів.

Найбільше реалістичні зображення одержують при використанні напівтонових методів. З його допомогою також відтворюють тільки видимі поверхні, використовуючи градацію їхньої яскравості. Напівтонові зображення мають найвищу наочність, проте характеризуються суттєвими (найвищими

серед розглянутих методів) обчислювальними затратами. Об'ємність зображень досягається за рахунок інтерполяції інтенсивності кольору в кожній точці зображення. Такий процес зафарбовування тривимірних областей одержав назву рендерінга. Для одержання реалістичних зображень використовують такі методи рендерінга: однотонний, метод Гуро і метод Фонга. Найбільше поширення одержав метод Гуро, тому що однотонне зафарбовування не забезпечує необхідної якості, а більш реалістичне зафарбовування Фонга вимагає занадто великих обчислювальних затрат.

Рендерінг - це багатетапний процес, який включає в себе ряд складних процедур, зокрема, розрахунок нормалізованих векторів, визначення кривизни поверхні, розрахунок фонові, дифузної та спекулярної складових кольору, накладання текстур. Перераховані процедури потребують великих обчислювальних витрат, що визначає актуальність розробки ефективних методів візуалізації тривимірних сцен у реальному часі.

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

Метою магістерської кваліфікаційної роботи є підвищення продуктивності формування графічних сцен за рахунок розпаралелення рендерингу за методом Гуро.

Основними задачами дослідження є:

- провести аналіз існуючих методів і засобів зафарбовування поверхонь тривимірних об'єктів;
- виведення аналітичної залежності приросту інтенсивностей кольору для визначення інтенсивностей кольору ділянки поверхні, обмеженої трикутником;
- розробити методи підвищення продуктивності рендерингу Гуро за рахунок розпаралелення ;
- розробити програмні та апаратні компоненти на основі запропонованих методів;

- провести експериментальні дослідження розроблених засобів зафарбовування.

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

Предмет дослідження – методи та засоби формування тривимірних графічних об'єктів.

Методи дослідження. У процесі досліджень використовувались: теорія чисел та чисельних методів, методи аналітичної геометрії для розробки методів рендерингу тривимірних об'єктів; комп'ютерне моделювання для аналізу та перевірки отриманих теоретичних положень.

Наукова новизна отриманих результатів.

1. Уперше встановлено залежність приросту інтенсивностей кольору для точок поверхні, обмеженої трикутником, що дає можливість підвищити продуктивність зафарбовування та спростити апаратну реалізацію пристроїв зафарбовування.

2. Подальшого розвитку отримав метод зафарбовування Гуро, який відрізняється від відомого визначення в кожному такті не однієї, а відразу кількох інтенсивностей точок трикутника, що дозволяє підвищити продуктивність формування графічних сцен.

Практична цінність отриманих результатів. Практична цінність одержаних результатів полягає в тому, що на основі отриманих в магістерській кваліфікаційній роботі теоретичних положень запропоновано алгоритми та розроблено програмні засоби зафарбовування у комп'ютерних системах візуалізації тривимірних зображень. Розроблено структурні схеми рендерів.

Особистий внесок здобувача. Усі наукові результати, викладені у магістерській кваліфікаційній роботі, отримані автором особисто. У друкованих працях, опублікованих у співавторстві, автору належать такі результати алгоритм розбиття трикутника на складові [8]; формули для нормалізації векторів [9]; програмний модуль для визначень інтенсивностей кольору [10], методи розпаралелення [20].

Апробація матеріалів магістерської кваліфікаційної роботи. Основні положення магістерської кваліфікаційної роботи доповідалися та обговорювалися на міжнародних та всеукраїнських конференціях: Міжнародна науково-практична конференція «Електронні інформаційні ресурси: створення, використання, доступ» (Вінниця,2019).

1 АНАЛІЗ МЕТОДІВ ФОРМУВАННЯ ТРИВИМІРНИХ ЗОБРАЖЕНЬ

1.1 Аналіз методів рендерингу

У наші дні комп'ютерна графіка, як наука формування реалістичних образів природних об'єктів, стає усе більш популярною. Вона використовується практично в усіх галузях для образного відтворення інформації і дозволяє оперативно прийняти рішення. Вона владно вторгається в бізнес, медицину, рекламу, індустрію розваг і т.д.

3D графіка, як складова частина машинної графіки, являє собою графічне представлення місця дії або об'єкта у трьох осях виміру: висота, ширина і глибина, щоб одержати більш реалістичне його зображення. Ця методика, що обманює користувача ПК, примушуючи спостерігати тривимірне зображення на плоскому екрані, є усе більш популярною на ринку комп'ютерних розваг, додаючи більше реалістичних і інтерактивних аспектів в графічні прикладні програми. Новітні досягнення в області 3D - технології відразу ж дають реальну віддачу в програмах моделювання і підвищують реалістичність комп'ютерних ігор. Більш того, тривимірна графіка дозволяє забезпечити унікальний інтерфейс користувача, що працює зі значними потоками даних.

Для відображення 3D об'єкта, його необхідно спочатку подати у вигляді набору точок і вершин у 3D системі координат (осі x , y , z). Вершини цього об'єкта зберігаються в системній оперативній пам'яті, і цілком визначають об'єкт. Для того, щоб відобразити цей об'єкт на плоскому 2D моніторі, об'єкт повинен бути відтворений.

Процес кінцевої візуалізації називають рендерингом (rendering).

У результаті рендерингу обчислення для кожного пікселя зображення кольору та адрес, що забезпечує ілюзію об'ємності тривимірного об'єкту двовимірному екрані. В результаті рендерингу растеризуються всі точки на поверхні об'єкта, охоплені полігглальною мережею.

У сучасних пакетах прикладних програм найбільшого поширення отримало завдання об'єкту сукупністю трикутників, оскільки в цьому випадку

растаризація є найпростішою. Можливо використання і теселяції для ескізної наглядності, однак на етапі рендерингу використовують найпростіший полігон - трикутник. Високополігональні моделі включають велику кількість трикутників. Такий обсяг інформації істотно впливає на швидкість відображення сцени на екрані комп'ютера і на час, необхідний для одержання остаточного зображення, тому на сьогоднішній день актуальними є питання підвищення продуктивності існуючих алгоритмів для процедур рендеринга.

Серед методів формування тривимірних зображень найбільше поширення одержали каркасний, контурний і напівтоновий. Каркасні (дротові) зображення складаються з ліній відтвореного об'єкта, які є видимі і невидимі спостерігачу. Об'ємність зображення основана на ефекті перспективи, що створює ілюзію глибини зображення. Недоліком каркасних зображень є прозорість об'єктів, тому що відтворюються всі лінії предметів і це призводить до втрати наочності.

Контурні зображення складаються тільки з видимих спостерігачу ліній і поверхонь. Видалення невидимих ліній потребує збільшення обсягу обчислень, що зростають приблизно в квадратичній залежності по мірі збільшення складності і кількості відтворених об'єктів.

Найбільше реалістичні зображення одержують при використанні напівтонових методів. З його допомогою також відтворюють тільки видимі поверхні, використовуючи градацію їхньої яскравості. Напівтонові зображення мають найвищу наочність, проте характеризуються суттєвими (найвищими серед розглянутих методів) обчислювальними затратами. Об'ємність зображень досягається за рахунок інтерполяції інтенсивності кольору в кожній точці зображення. Такий процес зафарбовування тривимірних областей одержав назву рендерінга [15,17,26-28]. Для одержання реалістичних зображень використовують такі методи рендерінга: однотонний, метод Гуро і метод Фонга [15,24,25]. Метод Гуро при високій щільності полігональної мережі[25] забезпечує високу реалістичність формування тривимірних сцен. Однотонне зафарбовування використовується, як правило, для ескізного проектування.

Високу реалістичність забезпечує зафарбовування Фонга, але має суттєвий недолік – високу трудомісткість.

Для реалізації тонування методом Гуро виконують триангуляцію об'єкта з метою спрощення подальших розрахунків. Якщо при формуванні полігональної мережі для кожної грані використовується по одній нормалі, то формується зображення, що включає однотонні ділянки (Error: Reference source not found.а). За методом Гуро можна отримати більш реалістичне зображення (Error: Reference source not found.б). Нормалі визначаються у вершинах полігональної мережі. Згідно з методом у вершинах трикутника визначаються інтенсивності кольору, які використовуються для визначення кольору вздовж ребер. Для цього використовується кодове лінійне інтерполювання. Наступним етапом є визначення кольору вздовж рядків растеризації .

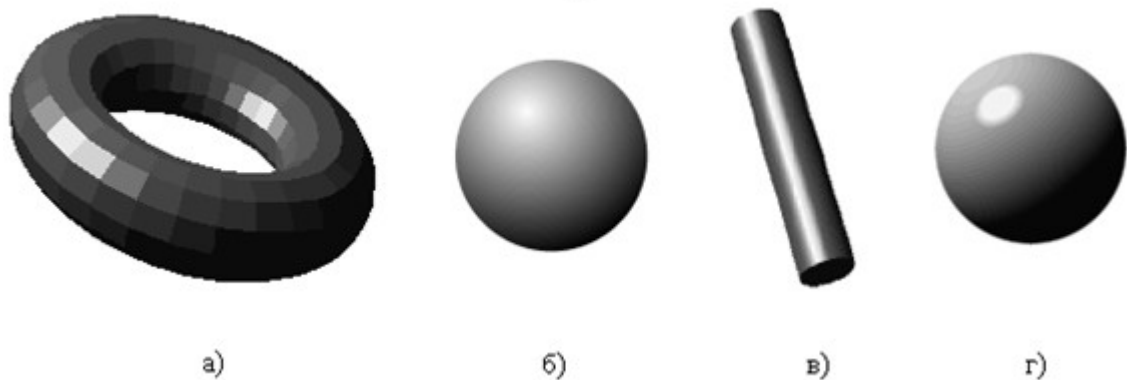


Рисунок 1.1 – Приклади синтезу об'ємних фігур

- а) однотонне заповнення,
- б) зафарбовування методом Гуро,
- в, г) зафарбовування методом Фонга

Недоліком методу Гуро є прояв ефекту смуг Маха [23], який полягає у візуальному збільшенні інтенсивності кольору на межах граней.

Смуги Маха обумовлені фізичними процесами в рецепторах ока людини. Рецептори, які реагують на зміну інтенсивностей кольору із більш яскравої сторони сильніше подразнюються порівняно з тими, знаходяться далі від межі (Рисунок 1.2).

Зафарбування Гуро найкраще виглядає в сполученні з простою моделлю освітлення з дифузійною складовою. Якщо відблиск буде розміщено всередині трикутника, то відблиск не буде відтворено.

Метод Фонга забезпечує формування більш реалістичних зображень. Метод подібний методу Гуро. Відмінність полягає у використанні замість інтенсивностей кольору у вершинах векторів до спостерігача, джерела світла та векторів нормалей. Отримані вектори використовуються для розрахунку інтенсивності кольору в кожній точці поверхні. Метод Фонга відтворює відблиски на поверхні (Error: Reference source not found, в, Error: Reference source not found, г). Проте, навіть якщо дзеркальний відбиток не використовується, інтерполяція векторів нормалі дає більш якісні результати, ніж інтерполяція інтенсивності, оскільки апроксимація нормалі в цьому випадку здійснюється в кожній точці. Отже, формується реалістичніше зображення. При цьому практично не помітні смуги Маха.



Рисунок 1.2 – Ефект полос Маха

Недоліком методу Фонга є значені обчислювальні затрати, тому що для кожного пікселя визначаються вектори, які нормалізуються. На основі векторів визначається інтенсивність кольору по вибраній моделі освітлення. Один із можливих підходів до скорочення кількості обчислень припускає використання так званої карти відбитків [29]. Вона являє собою набір попередньо

розрахованих інтенсивностей для визначеного діапазону значень вектора нормалі. У цьому випадку достатньо один раз обчислити карту відбитків, а потім, за відомим вектором нормалі, лише зчитувати з неї значення інтенсивності, не перераховуючи їх.

Рендеринг Фонга забезпечує формування більш реалістичніших зображень, але передбачає складні обчислення, тому для візуалізації тривимірних сцен часто використовують метод Гуро.

1.2 Особливості реалізації рендерингу Гуро

Розглянемо на прикладі процес зафарбування ділянки, обмеженої трикутником, за методом Гуро. В цьому випадку для трикутника задаються координатами вершин (А, В и С), а також значення інтенсивності у вершинах - I_A, I_B, I_C . (Error: Reference source not found).

У подальшому розраховують інтенсивності кольору точок в усіх внутрішніх точках трикутника

Визначимо приріст інтенсивності світла на один піксель уздовж ребер трикутника АВ, ВС, АС:

$$\Delta I_{AB} = (I_A - I_B) / \text{БП}_{AB} \quad (1.1)$$

$$\Delta I_{AC} = (I_A - I_C) / \text{БП}_{AC} \quad (1.2)$$

$$\Delta I_{BC} = (I_B - I_C) / \text{БП}_{BC} \quad (1.3)$$

де БП - більший з приростів координат $\Delta x, \Delta y$ обраного відрізка прямої.

Розглянемо рядок растеризації, що проходить через точку D (рис. 1.3). Кінцевими точками цього рядка будуть точки L і M.

Визначимо прирости інтенсивності вздовж ребер.

Інтенсивність кольору точок M і L розрахуємо таким чином.

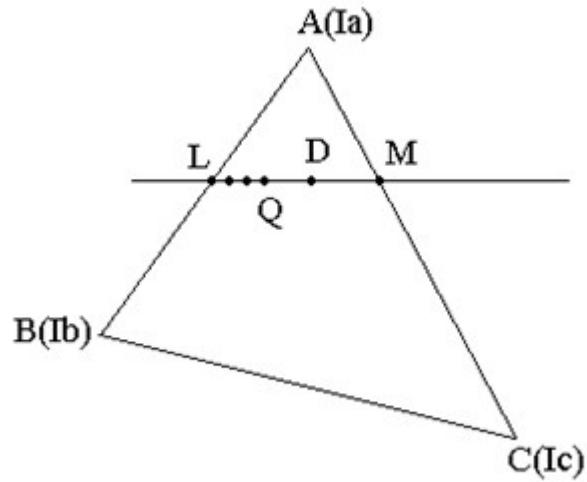


Рисунок 1.3 – Інтерполяція трикутника

$$I_M = I_A + \Delta I_{AB} * \text{БП}_{AM} \quad (1.4)$$

$$I_L = I_A + \Delta I_{AC} * \text{БП}_{AL} \quad (1.5)$$

Розрахуємо приріст інтенсивності вздовж рядка rasterизації LM

$$\Delta I_{LM} = (I_L - I_M) / LM \quad (1.6)$$

Знайдемо інтенсивність кольору точки D:

$$I_D = I_L + \Delta I_{LM} * LD \quad (1.7)$$

Аналогічним способом визначається інтенсивність будь-якої іншої точки, обмеженої заданим полігоном.

Найменша похибка досягається при розбивці поверхні на прямокутні трикутники з катетами паралельними координатним осям, оскільки в цьому випадку більший приріст (БП) катета збігається з його довжиною.

При зафарбуванні всього трикутника рядок, що сканує, послідовно проходить усі точки полігона. Приріст, як уздовж лінії, що сканує, так і між сусідніми лініями, що сканують, у цьому випадку складає один піксель, що дозволяє операції множення замінити операцією нагромаджувального додавання [13,15].

$$I_Q = I_L + \Delta I_{LM} + \Delta I_{LM} + \Delta I_{LM} \quad (1.8)$$

На рисунку 1.4, а наведено приклад рендерингу трикутника при $I_a=5$, $I_b=9$, $I_c=1$. Значення інтенсивностей світла отримані за описаними вище формулами (Рисунок 1.4.б).

Приведені формули інтерполяції використовуються також і в зафарбуванні Фонга з тією лише різницею, що для кожної вершини трикутника задається не інтенсивність світла, а вектор нормалі [14,18,19].

Наведений вище короткий огляд методів формування тривимірних зображень

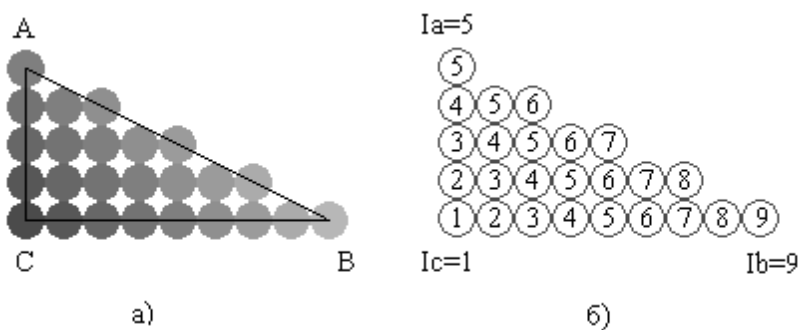


Рисунок 1.4 – Приклад зафарбовування трикутника

дає уяву про складність обчислювальних алгоритмів. Так в роботі [15] наводиться розрахунок необхідної потужності обчислювальних засобів для задач рендеринга. Для машинобудівної деталі середньої складності необхідно приблизно 10000 полігонів на екранний кадр при частоті регенерації 25 кадрів на секунду. Використовуючи метод Гуро, модель світла Торренса-Сперроу (найпростіший випадок), відсікання за алгоритмом Сазерленда-Ходжмана

мінімально необхідна потужність складе приблизно 150Мфлопс. Слід відмітити, що підрахунку необхідної обчислювальної потужності вибрані найпростіші алгоритми, котрі не завжди дають якісне зображення. Такі показники швидкодії досягаються на даний момент лише при використанні паралельних обчислювальних архітектур, що визначає актуальність подальших досліджень в цьому напрямку.

1.3 Висновки

- 1.Проведено аналіз методів зафарбовування.
- 2.Показано, що один із напрямків підвищення продуктивності формування графічних сцен пов'язано з розпаралеленням обчислювального процесу рендерингу.
- 3.Встановлено характерні особливості рендерингу Гуро.
- 4.Сформульовано мету магістерської кваліфікаційної роботи - підвищення продуктивності формування графічних сцен за рахунок розпаралелення рендерингу за методом Гуро.

2 ДОСЛІДЖЕННЯ РЕНДЕРИНГУ ГУРО

Обчислювальна складність методу Гуро значно менша, ніж методу Фонга[15,22]. Вказаним пояснюється переважне застосування в засобах машинної графіки методу Гуро.

В даному розділі досліджуються параметри кодової інтерполяції методу Гуро, його похибки та особливості апаратної реалізації.

2.1 Визначення параметрів кодової інтерполяції при рендерингу Гуро

Згідно методу Гуро, при скануванні рядка, який не проходить через вершини трикутника, інтенсивності точок цього рядка розраховується білінійною кодовою інтерполяцією інтенсивностей точок - вершин трикутника. При цьому для визначення інтенсивності точки Р (рис.2.1) необхідно обчислити інтенсивності точок М і N, а також ΔI - приріст інтенсивності між двома точками рядка сканування MN .

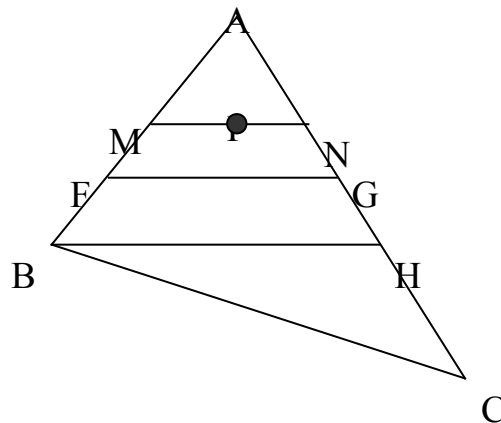


Рис.2.1 – Визначення інтенсивності точки методом Гуро

де ΔI_{MN}

$I_M -$

$$\Delta I_{MN} = \frac{I_M - I_N}{L_{MN}} ,$$

інтенсивність точки M, I_N - інтенсивність точки N, L_{MN} – довжина відрізка MN.

Аналогічні дії проводяться для визначення інтенсивностей всіх інших внутрішніх точок заданого трикутника

Доведемо, що приріст інтенсивності ΔI для будь-якого рядка сканування трикутника ABC - величина постійна.

Нехай трикутник ABC заданий координатами своїх вершин і значеннями їх інтенсивностей:

$$A(X, Y), I_A; \quad B(X_B, Y_B), I_B; \quad C(X_C, Y_C), I_C.$$

Знайдемо ΔI для рядків сканування FG та MN :

$$\Delta I_{FG} = \frac{I_G - I_F}{L_{FG}},$$

(2.1)

$$\Delta I_{MN} = \frac{I_N - I_M}{L_{MN}}. \quad (2.2)$$

Виразимо інтенсивності точок G та F через інтенсивності точок A , B і C :

$$\Delta I_G = \frac{I_C - I_A}{L_{AC}} \cdot L_{AG} + I_A, \quad (2.3)$$

$$\Delta I_F = \frac{I_B - I_A}{L_{AB}} \cdot L_{AF} + I_A. \quad (2.4)$$

Підставимо ці значення у формулу (2.1) для ΔI_{FG} :

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

(2.5)

Аналогічні дії виконуємо для точок M та N :

$$\Delta I_N = \frac{I_C - I_A}{L_{AC}} \cdot L_{AN} + I_A, \quad (2.6)$$

$$\Delta I_M = \frac{I_B - I_A}{L_{AB}} \cdot L_{AM} + I_A. \quad (2.7)$$

Розрахунки для ΔI_{MN} запишемо таким чином:

$$\Delta I_{MN} = \frac{I_C - I_A}{L_{AC}} \cdot \frac{L_{AN}}{L_{MN}} - \frac{I_B - I_A}{L_{AB}} \cdot \frac{L_{AM}}{L_{MN}}. \quad (2.8)$$

Розглянемо ΔAFG та ΔAMN .

У трикутниках $\angle AMN = \angle AFG$, $\angle ANM = \angle AFG$, так як $MN \parallel FG$, а $FG \parallel Ox$, тобто ΔAFG подібний ΔAMN . Так як трикутники AFG і AMN подібні, то:

$$\frac{L_{AG}}{L_{FG}} = \frac{L_{AN}}{L_{MN}}, \quad \frac{L_{AF}}{L_{FG}} = \frac{L_{AM}}{L_{MN}}.$$

(2.9)

З врахуванням виразу (2.5), з рівнянь (2.1) і (2.2) можна зробити висновок:

$$\Delta I_{FG} = \Delta I_{MN} = \Delta I.$$

(2.10)

Таким чином, значення ΔI є постійним для будь-якого рядка сканування трикутника ABC.

Оскільки у виразі для ΔI (2.5) і (2.8) у знаменнику стоїть довжина рядка сканування, то для зменшення похибки, необхідно, щоб знаменник був максимальний.

Довжиною рядка сканування $\triangle ABC$ буде довжина рядка ВН.

Тоді,

$$\Delta I = \frac{I_H - I_B}{L_{BH}},$$

де $L_{BH} = X_H - X_B$.

Знайдемо координату X точки Н.

Формула прямої, що проходить через дві точки має наступний вигляд:

$$\frac{Y - Y_1}{Y_2 - Y_1} = \frac{X - X_1}{X_2 - X_1}. \quad (2.11)$$

Для прямої АС запишемо:

$$\frac{Y - Y_A}{Y_C - Y_A} = \frac{X - X_A}{X_C - X_A}. \quad (2.12)$$

Точка Н належить ребру АС. $Y_H = Y_B$. Підставимо Y_H у вираз (2.12) і знайдемо X_H .

$$X_H = \frac{Y_B - Y_A}{Y_C - Y_A} \cdot (X_C - X_A) + X_A. \quad (2.13)$$

Знайдемо I_H .

$$I_H = \frac{I_C - I_A}{L_{AC}} \cdot L_{AH} + I_A = \Delta I_{AC} \cdot L_{AH} + I_A. \quad (2.14)$$

Кінцева формула для ΔI матиме вигляд:

$$\Delta I = \frac{\Delta I_{AC} \cdot L_{AH} + I_A - I_B}{X_H - X_B}, \quad (2.15)$$

де ΔI_{AC} - приріст інтенсивності в рядку сканування AC, $L_{AH} = \text{Max}\{X_H - X_A; Y_H - Y_A\}$, X_H - визначається з (2.13), $L_{AC} = \text{Max}\{X_C - X_A; Y_C - Y_A\}$.

У роботі [15] вираз для приросту інтенсивності на будь-якому рядку сканування ΔABC має такий вигляд:

$$\Delta I = \frac{\Delta W_{AC} \cdot \Delta Y_{BC} - \Delta W_{BC} \cdot \Delta Y_{AC}}{\Delta X_{AC} \cdot \Delta Y_{BC} - \Delta X_{BC} \cdot \Delta Y_{AC}}. \quad (2.16)$$

Співставимо вирази (2.15) і (2.16).

Безпосередня перевірка показала ідентичність результатів, отриманих по виразах (2.15) і (2.16).

У виразі (2.15) використовується менше "довгих" операцій (множення/ділення), чим у виразі (2.16). Для визначення ΔI відповідно до (2.15) необхідно 4 операції множення/ділення, а відповідно до виразу (2.16) - 5, що дозволяє зменшити обчислювальні витрати при її реалізації.

Вираз (2.15), на відміну від (2.16) дозволяє судити про характер накопичення похибки при обчисленні приросту інтенсивності.

2.2 Визначення максимальної похибки розрахунку інтенсивності внутрішніх точок трикутника

У розділі 2.1. було доведено, що приріст інтенсивності для будь-якого рядка сканування трикутника є величина постійна. При цьому був отриманий вираз для знаходження цього приросту (формула 2.15). В чисельнику виразу (2.15), в силу дискретності представлення відрізків в координатному просторі, використовується не дійсна довжина відрізка L_{AH} , а його більший приріст BP_{AH} . Це вносить похибку при обчисленні приросту інтенсивності точок в середині трикутника.

$$\Delta I = \frac{\Delta I_{AC} \cdot BP_{AH} + I_A - I_B}{L_{BH}}.$$

(2.17)

Розглянемо характер накопичення похибки при визначенні приросту інтенсивності кольору внутрішніх точок трикутника і знайдемо вираз, при якому має місце його максимальне значення.

При рендерингу Гуро інтенсивність точок ребра трикутника визначається методом кодової інтерполяції [15]. Визначимо похибку, яка виникає при визначенні приросту інтенсивності кольору точок в цьому випадку.

Ідеальне значення приросту інтенсивності кольору на ребрі трикутника знаходиться за наступною формулою:

$$\Delta I_i = \frac{I_C - I_A}{L_{AC}}. \quad (2.18)$$

У силу дискретності представлення відрізків в растрових пристроях відображення інформації, у знаменнику виразу (2.18) замість реальної довжини відрізка буде використовуватись його більший приріст $БП$:

$$\Delta I_p = \frac{I_C - I_A}{БП_{AC}}. \quad (2.19)$$

Це вносить похибку при визначенні приросту інтенсивностей кольору точок вздовж ребра AB . Розрахуємо її значення:

$$\delta_{AC} = |\Delta I_i - \Delta I_p|.$$

З врахуванням (4.18) і (4.19)

$$\delta_{AC} = |I_C - I_A| \cdot \left| \frac{1}{L_{AC}} - \frac{1}{БП_{AC}} \right| = |I_C - I_A| \cdot \frac{1}{L_{AC} \cdot БП_{AC}} \cdot |БП_{AC} - L_{AC}|. \quad (2.20)$$

Проаналізуємо отриманий вираз (2.20).

Мінімальне значення похибки δ_{AC} дорівнює 0, коли $L_{AC} = БП_{AC}$. Це має місце, коли відрізок AC паралельний вісям координат (рис.2.2)

Похибка (2.20) максимальна, коли $БП_{AC} = МП_{AC}$. В цьому випадку $БП_{AC} =$

$$\frac{L_{AC}}{\sqrt{2}}.$$

$$\delta_{\max} = \Delta I_i \cdot |1 - \sqrt{2}|.$$

Тобто, при визначенні приросту інтенсивності ребра трикутника, похибка досягає свого максимального значення для відрізків, кут нахилу яких до вісі Ox є кратним 45° . Її чисельне значення розраховується згідно виразу (2.20).

Визначимо похибку, яка має місце при визначенні інтенсивності кольору точки H , що належить ребру AC .

Ідеальне значення інтенсивності точки H розраховується згідно виразу:

$$I_{IH} = I_A + L_{AH} \cdot \Delta I_i, \quad (2.21)$$

де ΔI_i - ідеальне значення приросту інтенсивності на відрізку AC .

При заміні довжини відрізка на його більший приріст, вираз (2.21) буде мати наступний вигляд:

$$I_{HP} = I_A + БП_{AH} \cdot \Delta I_p, \quad (2.22)$$

де ΔI_p - приріст інтенсивності точок на відрізку AC при використанні більшого приросту відрізка.

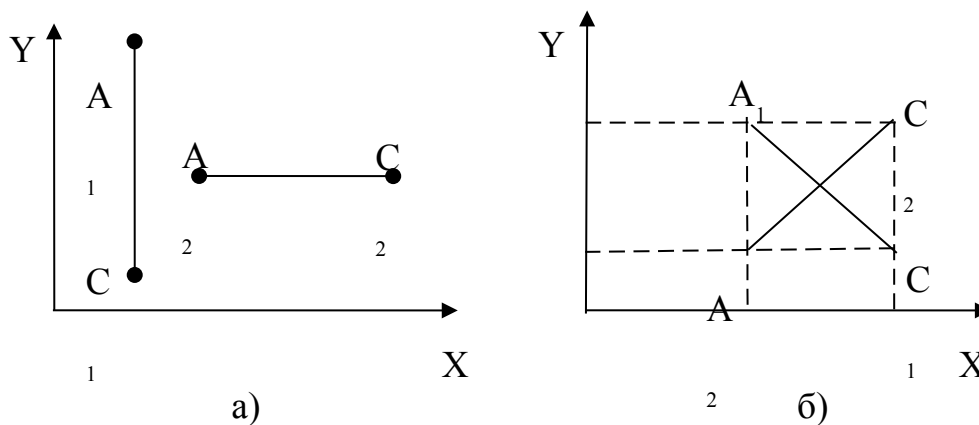


Рис. 2.2 Випадки, коли δ_{AC} досягає свого мінімального (а) та максимального (б) значень

Заміна L_{AH} на $БП_{AH}$ вносить похибку δ_H :

$$\delta_H = |I_{IH} - I_{HP}|.$$

З врахуванням (2.18) і (2.19) вираз для δ_H буде мати наступний вигляд:

$$\delta_H = \left| L_{AH} \cdot \frac{I_C - I_A}{L_{AC}} - БП_{AH} \cdot \frac{I_C - I_A}{БП_{AC}} \right|,$$

$$\delta_H = \left| (I_C - I_A) \cdot \left[\frac{L_{AH}}{L_{AC}} - \frac{БП_{AH}}{БП_{AC}} \right] \right|. \quad (2.23)$$

Проаналізуємо вираз (2.23). Для цього розглянемо $\triangle AC'C$ (рис. 2.3).

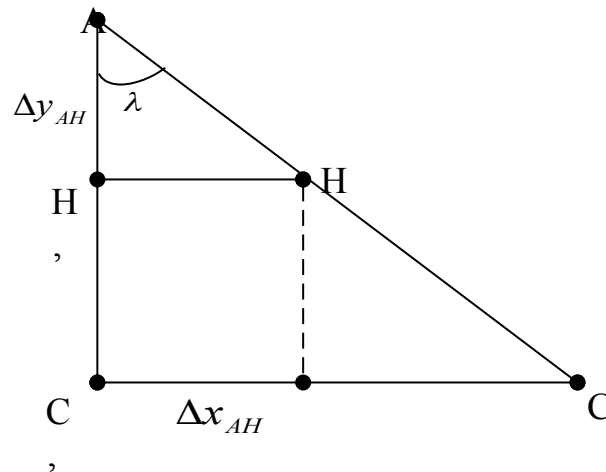


Рис. 2.3 – Аналіз похибки

$\triangle AC'C$ подібний $\triangle AH'H$ по другій ознаці подібності. В цьому випадку

$$\frac{L_{AH'}}{L_{AC'}} = \frac{L_{H'H}}{L_{C'C}} = \frac{L_{AH}}{L_{AC}}, \quad \frac{BP_{AH'}}{BP_{AC'}} = \frac{L_{AH}}{L_{AC}}.$$

З врахуванням останнього виразу, вираз (2.23) буде мати нульове значення для будь-яких значень приростів координат і значень інтенсивностей кольору в кінцевих точках відрізка:

$$\delta_H = 0.$$

Таким чином доведено, що похибка визначення інтенсивності довільної точки відрізка згідно виразу (2.22) дорівнює 0 для будь-яких значень приростів координат між кінцями даного відрізка. Дана властивість дозволяє підвищити швидкодію кодової інтерполяції за допомогою переходу від дійсних значень довжин відрізка до його більшого приросту без накопичення похибки.

При практичній реалізації даного алгоритму вносяться похибки, пов'язані з дискретністю представлення зображення в координатному просторі (рис.2.4).

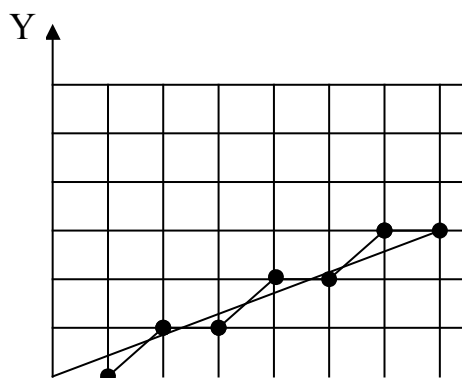




Рисунок 2.4 – Представлення ліній в дискретному координатному просторі

Внаслідок цього, при визначенні похибки необхідно також враховувати похибки інтерполяційних методів (наприклад, алгоритму Брезенхема [22]).

Визначимо похибку визначення інтенсивності довільної точки відрізка з врахуванням похибок методів інтерполяції. В якості інтерполяційного алгоритму розглянемо алгоритм Брезенхема.

В роботі [15] було доведено, що похибка алгоритму Брезенхема дорівнює:

$$E_i = \left| 0,5 - \frac{O\Phi_i}{2 \cdot \Delta x} \right|, \quad (2.24)$$

де $O\Phi_i$ – значення оціночної функції в i -тій точці відрізка, Δx - приріст відрізка прямої по вісі x .

Фізичний зміст E_i в тому, що E_i – це відстань i -тої точки від ідеальної прямої, що визначається по вісі Oy (рис. 2.5).

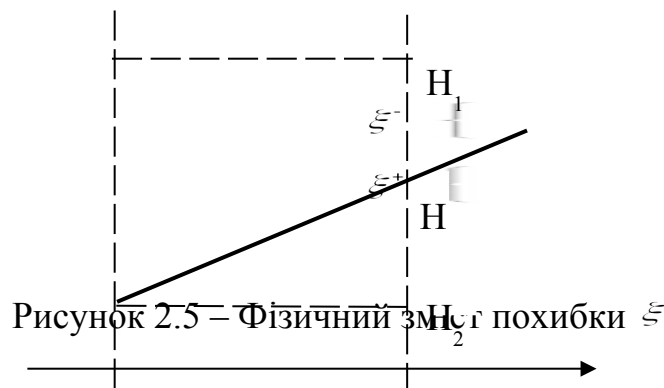


Рисунок 2.5 – Фізичний зміст похибки E_i

Відповідно до виразу (2.22) знаходиться інтенсивність точки, яка розташована на ідеальній прямій (рис. 2.5). Реальне місце розташування точки $N - N_1$ або N_2 потребує корекції інтенсивності. Враховуючи, що значення інтенсивності точок вимагає корекції по відомій вісі координат, приріст між точками N_1 і N_2 буде мати наступний вигляд:

$$\Delta I' = \frac{I_C - I_A}{MPP_{AC}}.$$

Значення, на яке необхідно відкоригувати інтенсивність точок N_1 чи N_2 дорівнює: $\Delta I' \cdot E_i$.

Вираз для похибки інтенсивності в довільній точці відрізка має наступний вигляд:

$$\delta_H = \left| (I_C - I_A) \cdot \left[\frac{L_{AH}}{L_{AC}} - \frac{БП_{AH}}{БП_{AC}} \right] \right| \cdot \frac{|I_C - I_A|}{МП_{AC}} \cdot E_H.$$

З врахуванням (2.23):

$$\delta_H = \frac{|I_C + I_A|}{МП_{AC}} \cdot E_H, \quad (2.25)$$

де E_H - похибка інтерполяційного методу, що визначається згідно виразу (2.24).

Як було показано в мінімальне і максимальне значення похибки дорівнюють:

$$E_{\min} = 0, \quad E_{\max} = \frac{1}{2}.$$

Тоді, $E_{H \min} = 0$ і досягається на відрізках, кут нахилу яких до вісі x є кратним 45° .

Максимальне значення похибки, яка розраховується згідно виразу (2.25), має місце для відрізків, в яких на горизонтальний (вертикальний) крок по ведучій координаті приходиться один діагональний крок (рис.2.6).

При цьому:

$$\delta_{H \max} = \frac{|I_C - I_A|}{2 \cdot МП_{AC}} - \frac{\Delta I_{МП}}{2}.$$

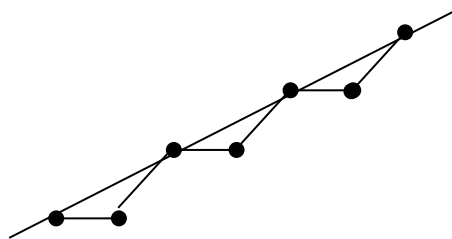


Рисунок 2.6 – Приклад відрізка, де δ_H приймає максимальне значення

Таким чином, похибка при розрахунку інтенсивності довільної точки відрізка з врахуванням похибок інтерполяційних методів не перевищує половини приросту інтенсивності по відомій координаті.

Визначимо похибку при розрахунку інтенсивності згідно виразу (2.15).

Точне значення приросту інтенсивності дорівнює:

$$\Delta I_i = \frac{\Delta I_{iAC} \cdot L_{AH} + I_A - I_B}{L_{BH}}.$$

При заміні значень довжин відрізків на їх більші прирости, формула буде мати наступний вигляд:

$$\Delta I = \frac{\Delta I_{AC} \cdot \text{БП}_{AH} + I_A - I_B}{L_{BH}}.$$

При цьому похибка дорівнює:

$$\delta_{\Delta I} = |\Delta I_i - \Delta I|,$$

$$\delta_{\Delta I} = \frac{|\Delta I_{iAC} \cdot L_{AH} - \Delta I_{AC} \cdot \text{БП}_{AH}|}{L_{BH}}.$$

З врахуванням (2.2) і (2.3), попередній вираз буде мати вигляд:

$$\delta_{\Delta I} = \frac{|I_C - I_A|}{L_{BH}} \cdot \left| \frac{L_{AH}}{L_{AC}} - \frac{\text{БП}_{AH}}{\text{БП}_{AC}} \right|. \quad (2.26)$$

Проаналізуємо отриманий вираз. Для цього розглянемо $\Delta B'CA$ і ΔBHA (рис.2.7)

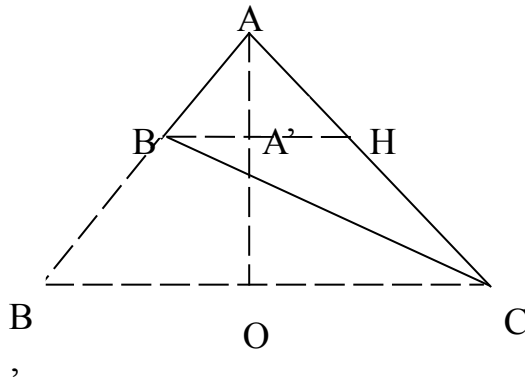


Рисунок 2.7 – Визначення похибки при розрахунку інтенсивності

З признаку подібності $\Delta B'CA$ і ΔBHA витікає:

$$\frac{L_{AA'}}{L_{AO}} = \frac{L_{A'H}}{L_{OC}} = \frac{L_{AH}}{L_{AC}}, \quad \frac{\text{БП}_{AH}}{\text{БП}_{AC}} = \frac{L_{AH}}{L_{AC}}.$$

З врахуванням останньої тотожності, вираз (2.26) приймає нульове значення.

Таким чином, приріст інтенсивності для будь-якого рядка сканування трикутника, що визначається відповідно до виразу (2.15), з врахуванням переходу від дійсних довжин відрізків до їхніх більших приростів, обчислюється без похибки. Це дозволяє обчислювати ΔI без знаходження дійсних довжин сторін трикутника, користуючись тільки приростами координат його сторін, що істотно спрощує обчислення.

Визначимо похибку визначення інтенсивності кольору довільної внутрішньої точки E, при зафарбовуванні трикутника по методу Гуро (рис.2.8).

$$\delta_E = \delta_{AD} + \delta_{DE},$$

де δ_{AD} - похибка, що отримується при визначенні інтенсивності точки D на ребрі трикутника, δ_{DE} - похибка, що отримується на рядку сканування DE' при визначенні інтенсивності точки E.

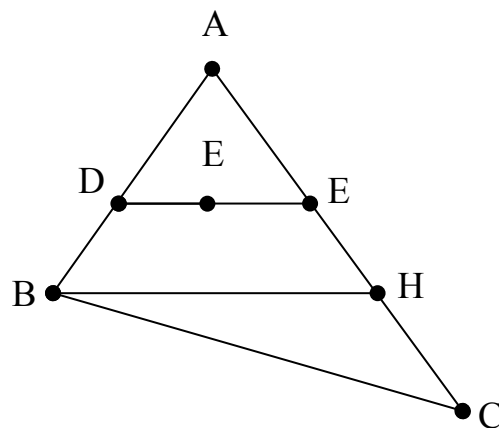


Рисунок 2.8 – Похибка визначення інтенсивності точки E по методу Гуро

Відповідно до виразів (2.21) та (2.25):

$$\delta_{AD} = \frac{|I_C - I_A|}{MII_{AC}} \cdot ED, \quad \delta_{DE} = |I_{DE} - I_E|.$$

Враховуючи вираз (2.23):

$$\delta_{DE} = 0.$$

Таким чином $\delta_E = \delta_{AD}$:

$$\delta_E = \frac{|I_C - I_A|}{MII_{AC}} \cdot \left| 0.5 - \frac{O\Phi}{2 \cdot \Delta x} \right|. \quad (2.27)$$

Вираз (2.27) визначає похибку при обчисленні інтенсивності довільної внутрішньої точки трикутника, що зафарбовується по методу Гуро.

Проаналізуємо отриманий вираз.

1. Вираз (2.27) приймає мінімальне значення в випадку, коли точка D лежить на ідеальній прямій АВ (кут нахилу прямої до вісі x є кратним 45°).

2. Визначимо $\delta_{E_{\max}}$. Як було показано в [30] $\delta_{D_{\max}} = \frac{1}{2}$ тоді,

$$\delta_{E_{\max}} = \frac{|I_C - I_A|}{2 \cdot MП_{AC}}.$$

Можна запропонувати декілька шляхів для зменшення похибки при визначенні інтенсивності кольору внутрішніх точок трикутника по методу Гуро:

1. В процесі тріангуляції полігональну область потрібно розбивати на прямокутні трикутники.

2. При декомпозиції полігонів на співставні трикутники необхідно намагатися, щоб трикутники були приблизно однакових розмірів (площ).

2.3 Висновки

1. Доведено, що приріст інтенсивності кольору для ділянки поверхні, обмеженої трикутником, є сталою величиною, що дозволяє суттєво підвищити продуктивність зафарбовування та спростити апаратну реалізацію пристроїв кінцевої візуалізації.

2. Визначено максимальну похибку розрахунку інтенсивності внутрішніх точок трикутника.

3 РОЗРОБКА МЕТОДІВ РОЗПАРАЛЕЛЕННЯ РЕНДЕРИНГУ ГУРО

3.1 Розробка методів розпаралелення

При класичному підході до реалізації рендеринга використовуються два лінійних інтерполятора, що формують координати точок, що складають ребра трикутника. По розрахунковому значенню приросту інтенсивності по всіх ребрах трикутника здійснюється кодова інтерполяція. Для формування координат внутрішніх точок трикутника використовується рядковий лінійний інтерполятор. Для визначення інтенсивності внутрішніх точок трикутника використовується відповідний кодівий інтерполятор, який можна поєднати з кодівим інтерполятором, що формує ребра трикутника. Таким чином, класична реалізація рендеринга потребує два лінійних інтерполятора, один кодівий і один рядковий.

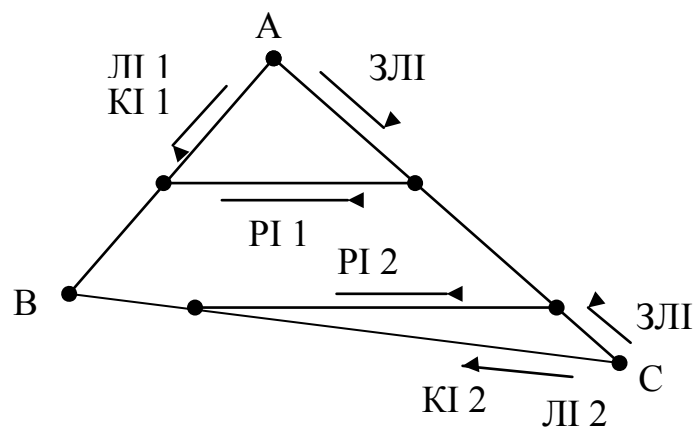


Рисунок 3.1 - Розподіл обчислювального процесу між інтерполяторами

Один із запропонованих підходів (рис. 3.2) до реалізації паралельного зафарбування полігона трикутника заснований на зустрічній кодівій інтерполяції.

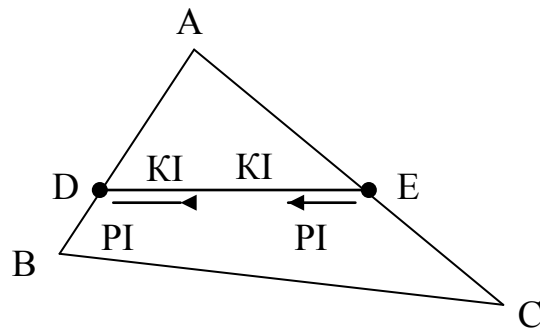


Рисунок 3.2 – Зафарбовування методом зустрічної кодової інтерполяції

Він полягає в тому, що значення інтенсивності в рядку трикутника, що сканується, формують два кодових інтерполятора, що працюють назустріч один одному з початкової і кінцевої точок рядка, що сканується. Лінійна інтерполяція ребер трикутника відбувається аналогічно класичному методу рендеринга. Запропонований принцип дозволяє прискорити кодову інтерполяцію рядка, що сканується, у два рази, а також зменшити похибку рендеринга за рахунок розподілу її між двома кодовими інтерполяторами. У порівнянні з класичним методом додатково вимагається один кодовий і один рядковий інтерполятори.

Інший можливий шлях прискорення процесу рендеринга полягає в незалежному обчисленні інтенсивностей пікселів для парних і непарних точок рядка, що сканується (рис. 3).

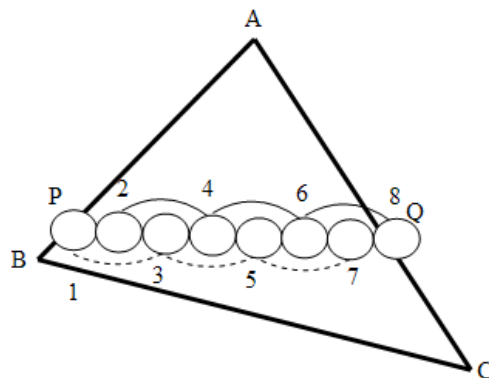


Рисунок 3.3 - Визначення інтенсивностей кольору в парних і непарних точках

Перший кодовий інтерполятор визначає інтенсивність тільки непарних точок рядка трикутника, використовуючи подвоєне значення приросту

інтенсивності в рядку, що сканується, ($2 \cdot \Delta I$). Інтенсивність парних точок рядка, що сканується, визначає другий кодовий інтерполятор. Даний метод дає можливість прискорити кодову інтерполяцію скануючого рядка, у два рази. Додаткові апаратні витрати для даного підходу, як і в попереднього методу, складають один кодовий і один рядковий інтерполятори.

Запропонований принцип можна поширити і на більшу кількість кодових інтерполяторів. Їхнє оптимальне число не перевищує чотирьох, це пов'язано з тим, що розміри $2 \cdot \Delta I$, $3 \cdot \Delta I$, $4 \cdot \Delta I$ можна визначити за допомогою однієї операції зсуву й одного додавання. Визначення приросту інтенсивності більшої кратності потребує додаткових обчислювальних витрат, що в більшості випадків не виправдано.

3.2 Розробка апаратних засобів розпаралелення

Структурна схема адресного блоку рендерера, в якому здійснено розпаралелення обчислювального процесу за рахунок незалежного зафарбовування парних і непарних точок в рядках растеризації, приведена на рис.3.4. На відміну від класичного підходу адресний блок додатково включає адресний лічильник для парних точок а також блок для блокування роботи адресного лічильника парних точок.

В цілому робота адресного блока мало відрізняється від описаної вище роботи адресного блока при класичному підході. Відмінність полягає в наступному. Адресний лічильник, який формує адреси непарних точок в рядку растеризації, виконує мікрооперації інкрементування, декрементування і також збільшення вмісту на два. Адресний лічильник $L_{чП}$ працює тільки на збільшення свого вмісту на одиницю та два.

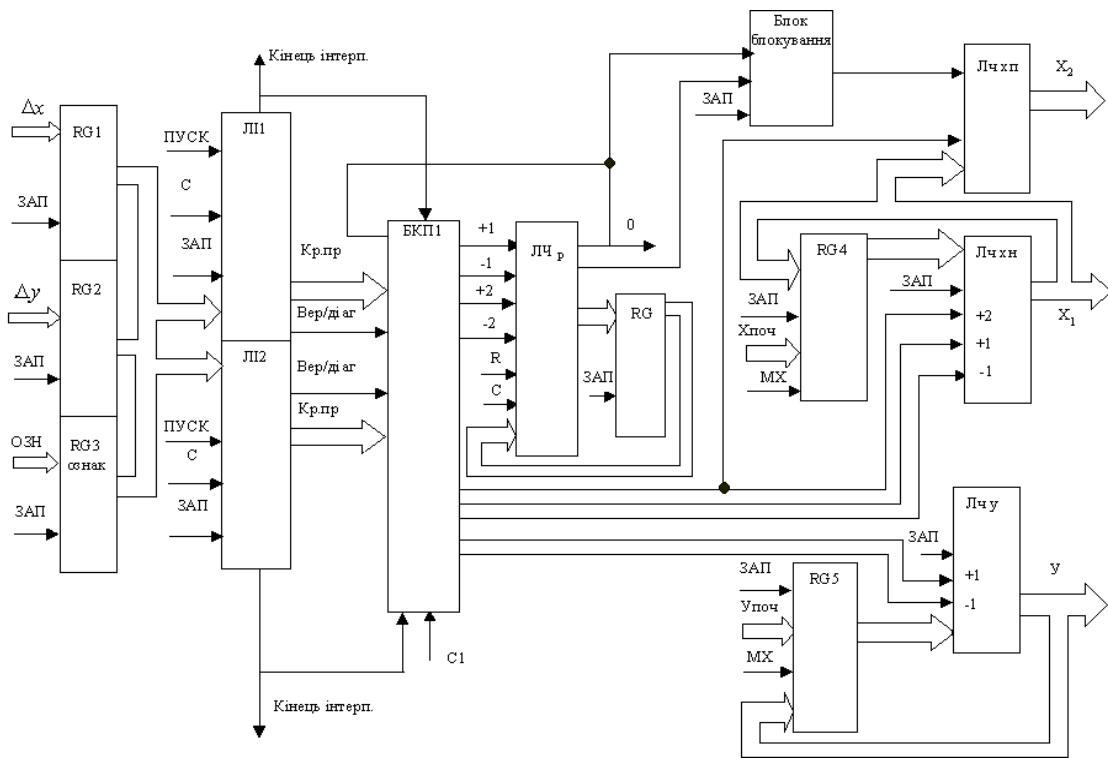


Рис. 3.4 – Структурна схема адресного блоку

При непарній кількості точок в рядку rasterизації в останньому такті формування адрес точок в рядку rasterизації здійснюється блокування роботи адресного лічильника ЛЧХП. Для цього введений спеціальний блок для його блокування. При формуванні лінійними інтерполяторами ЛП1, ЛП2 вертикальних(діагональних) крокових приростів здійснюється перехід до формування рядка rasterизації. При цьому адресний лічильник ЛЧХП містить координату першої точки рядка rasterизації. Будем вважати її парною. В лічильник імпульсів ЛЧХП з адресного лічильника ЛЧХН переноситься адреса першої непарної точки рядка rasterизації. Оскільки адресний лічильник ЛЧХП формує парні точки рядка rasterизації, то його вміст інкрементується. Таким чином в циклі підготування до формування рядка rasterизації в адресних лічильниках ЛЧХН, ЛЧХП зберігаються відповідно адреси першої та другої точки рядка rasterизації.

В циклі rasterизації адресні лічильники ЛЧХН, ЛЧХП синхронно збільшують свій вміст на два, формуючи таким чином адреси парних та

непарних точок рядка растерізації. Оскільки ординати всіх точок в рядку растерізації однакові, то адресний блок включає тільки один координатний лічильник імпульсів ЛПУ. В кожному такті цикла растерізації вміст лічильника ЛЧр зменшують на два. При формуванні на виході лічильника ЛЧр сигналу переповнення растерізацію рядка завершують.

При парному початковому значенню рядкового лічильника Лчр на його виході парності формується сигнал логічної одиниці, який запам'ятовується в блоку блокування координатного лічильника ЛЧХП. При парності довжини рядка чканування в останньому такті растерізації рядка на виході блока блокування формується сигнал, який забороняє зміну стану лічильника ЛЧХП.

Структурна схема блоку визначення інтенсивностей кольору точок рендерера, в якому здійснено розпаралелення обчислювального процесу за рахунок незалежного зафарбовування парних і непарних точок в рядках растерізації приведена на рис. 3.5. На відміну від ідентичного блоку, реалізованого по класичному підходу, в блок визначення інтенсивностей кольору додатково введено суматор Sm2, регістри RG6, RG7.

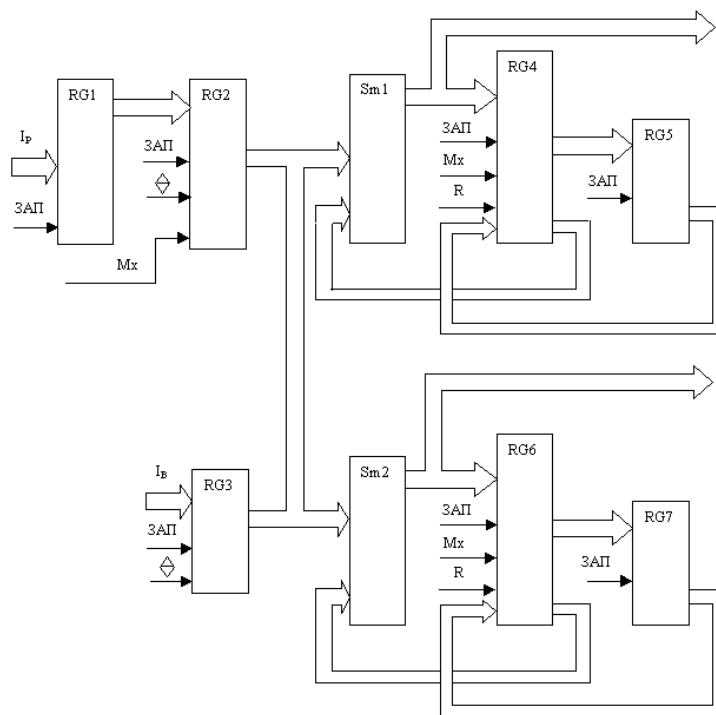


Рис.3.5 – Структурна схема блоку визначення інтенсивностей кольору внутрішніх точок трикутника

Регістри RG1...RG5 використовуються для визначення інтенсивностей непарних точок рядка растерізації. Вказана частина схеми по своїй суті і принципу роботи не відрізняється від базового блоку (класичний метод). Регістри RG3, RG6, RG7 та суматор Sm2 визначають інтенсивності непарних точок рядка растерізації. Регістр RG3 є зсувним регістром.

У циклі підготовки в регістри RG2, RG1 заносять відповідно прирости інтенсивності кольору ребр, від яких здійснюється растерізація.

Регістр RG3 є багатофінаціональним. У нього заноситься значення інтенсивності кольору точки трикутника, яка має найменшу ординату. верхньої. При цьому регістри RG4 і RG6 обнуляються, що забезпечує передачу на вихід суматорів Sm1 і Sm2 вмісту регістра RG3, оскільки регістр RG2 блокується. У регістрах RG4 і RG6 зберігається інтенсивності кольору точки верхньої вершини трикутника.

У RG3 заноситься значення приросту інтенсивності кольору вздовж рядка растерізації.

На вхід суматора Sm2 з регістра RG3 надходить значення приросту інтенсивності кольору вздовж ребр растерізації, яке сумується зі значенням інтенсивності кольору верхньої точки і зберігається в регістрі RG6. Таким чином в регістрі RG6 знаходиться інтенсивність кольору наступної точки рядка растерізації по відношенню до граничної. Значення інтенсивностей кольору, які зберігаються в регістрах RG4, RG6 переносять в регістри RG5, RG7. Вміст регістру RG3 зсувається в сторону старших розрядів, тобто визначає значення $2\Delta I$ рядкове.

У циклі визначення інтенсивностей точок трикутника спочатку визначаються інтенсивності по ребрам. Для цього регістр RG3 переводять в високоімпедантний стан, а вміст регістра RG2 подається на входи суматорів Sm1, Sm2 і здійснюють потактне накопичуюче додавання. Результати додавання зберігаються в регістрах RG4, RG6. При формуванні адресним блоком діагонального(вертикального) кроку виконується перехід до визначення інтенсивності кольору рядка растерізації. Для цього регістр RG2 блокується, а

регістр RG3 виводять з третього стану. Вміст регістра RG3 поступає на накопичуючі суматори (Σ_1 , RG4), (Σ_2 , RG6). У вказаних нагромаджувальних суматорах в кожному такті визначається інтенсивність відповідно парних і непарних точок рядка растеризації. Після закінчення формування інтенсивностей кольору рядка растеризації, регістр RG3 блокується, а в регістри RG4, RG6 заноситься вміст регістрів RG5, RG7.

Вказані дії виконуються до закінчення формування адресним блоком останньої точки провідного ребра.

Адресний блок завершує роботу після визначення адреси останньої точки привідного ребра.

У подальшому у регістр RG2 переноситься вміст регістра RG1, у якому зберігається приріст інтенсивності другого провідного ребра або непровідного ребра, яке має з провідним ребром спільну точку з його нижньою вершиною.

На рис. 3.7, 3.8 зображено структурні схема пристроїв для визначення приростів інтенсивностей кольору та інтенсивностей кольору точок рядка растеризації.

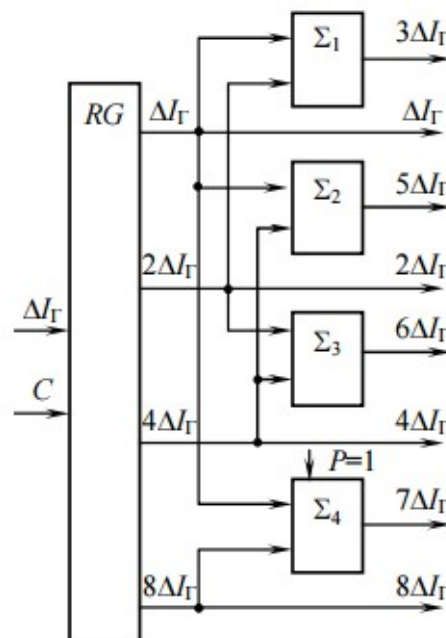


Рисунок 3.6 – Структурна схема пристрою для визначення приростів інтенсивностей кольору

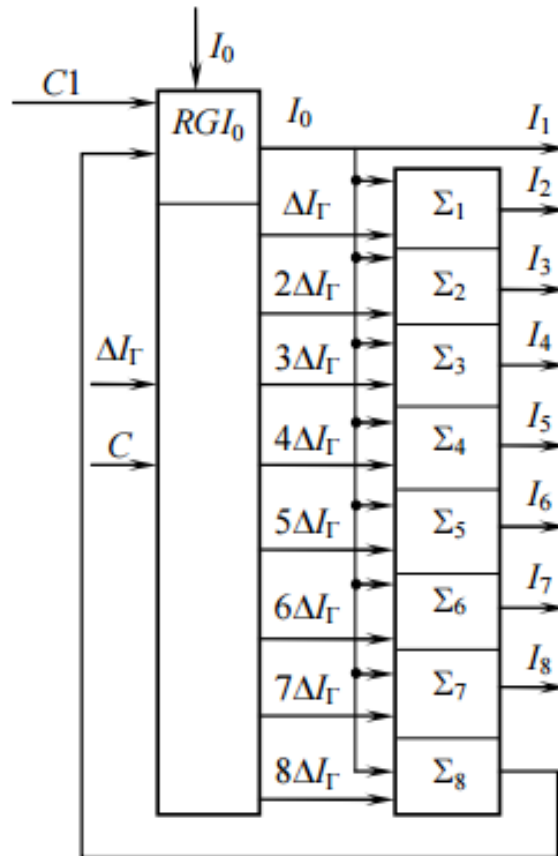


Рисунок 3.7 – Структурна схема пристрою для визначення інтенсивностей кольору

3.3 Паралельний рендеринг з використанням додаткової триангуляції

Пропонується метод, який дозволяє прискорити процес зафарбування трикутника. Для цього проаналізуємо існуючу теорему про середню лінію трикутника.

Проведемо в трикутнику ABC (рис 3.8) три середні лінії KL, LM, KM. За визначенням середня лінія - це лінія, що проходить через середини сторін трикутника. Властивості середньої лінії такі:

- а) середня лінія паралельна одній із сторін трикутника;
- б) довжина середньої лінії дорівнює половині сторони трикутника, якій вона паралельна.

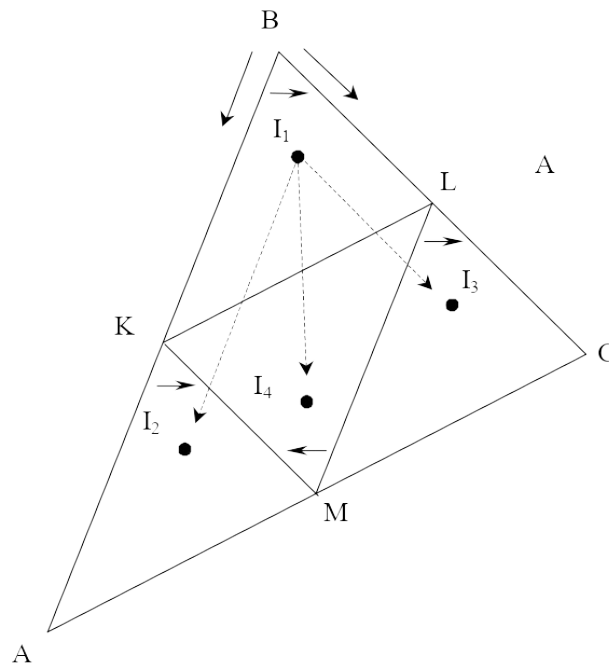


Рисунок 3.8 – Паралельний рендеринг трикутника

Доведемо рівність трикутників АКМ і МLC. Ці трикутники рівні, тому що в них відповідні сторони рівні:

$$AM=MC=0.5*AC \quad (3.1)$$

$$AK=ML=0.5*AB \quad (3.2)$$

$$KM=LC=0.5*BC \quad (3.3)$$

Аналогічно доводиться взаємна рівність трикутників АКМ, МLC, KBL, LMK.

Розглянемо тепер процес інтерполяції. При інтерполяції трикутника приріст уздовж його сторони величина постійна, тому можна вважати, що після інтерполяції кожна точка трикутника буде відрізнятися від відповідної їй точки іншого трикутника лише певну на константу. Наприклад точка С переходить у точку М, а точка М переходить у точку А. Приріст СМ дорівнює приросту МА і дорівнює половині приросту СА.

Таким чином немає необхідності зафарбовувати весь трикутник ABC достатньо зафарбувати один малий трикутник, наприклад, KBL, а потім розмножити результат зафарбування на інші трикутники.

Ефективно проводити паралельно процес зафарбування малого трикутника і формування значень інтенсивності світла для інших малих трикутників на трьох додаткових суматорах. Починати зафарбування будемо завжди із самого верхнього трикутника. Перед початком рендеринга за відомим значенням інтенсивності у вершинах трикутника обчислюються значення констант зсуву.

Для точки K:

$$\begin{cases} \Delta X = (X_A - X_B)/2 \\ \Delta Y = (Y_A - Y_B)/2 \\ \Delta I = (I_A - I_B)/2 \end{cases} \quad (3.4)$$

Для точки L:

$$\begin{cases} \Delta X = (X_C - X_B)/2 \\ \Delta Y = (Y_C - Y_B)/2 \\ \Delta I = (I_C - I_B)/2 \end{cases} \quad (3.5)$$

Для точки M:

$$\begin{cases} \Delta X = (X_A - X_C)/2 \\ \Delta Y = (Y_A - Y_C)/2 \\ \Delta I = (I_A - I_C)/2 \end{cases} \quad (3.6)$$

У такий спосіб для роботи методу необхідно 4 рядкових інтерполятора по одному на кожен малий трикутник для визначення координат пікселя, 2 лінійних інтерполятора для формування сторін трикутника, 1 кодівий інтерполятор для формування значень інтенсивності пікселів на сторонах і в

середині малого трикутника і три суматора, що формують значення інтенсивності пікселів для трьох малих трикутників.

Використання даного методу дозволяє прискорити процес зафарбування приблизно в чотири рази, так як площа малого трикутника в чотири рази менша площі початкового трикутника, а зафарбування частин трикутника, що залишилися, відбувається паралельно і не потребує витрат часу.

Труднощі запропонованого підходу полягають у тому, що у дискретних системах відображення інформації відрізок ділиться чітко пополам, тобто його реальна середина попадає в точку сітки зображення лише у випадку, коли обидва, і більший і менший, прирости його координат парні. В інших випадках середина відрізка не буде попадати в конкретну точку сітки зображення. Крім того, слід довести відсутність точок просікання між чотирма складовими трикутниками.

Пропонуються два напрямки, щодо розв'язання проблеми побудови дискретного відрізка з двох однакових частин:

а) зміна координат вузлових точок таким чином, щоб вони задовольняли умовам парності приростів координат сторін:

- коригування точок полігональної сітки;
- побудова трикутника $A_1B_1C_1$ менших розмірів з парними значеннями БП і МП його сторін з додатковою доводкою ребер трикутника ABC;

- побудова трикутника $A_1B_1C_1$ більших розмірів з парними значеннями БП і МП його сторін;

б) побудова дискретного відрізка з двох частин з певною, допустимою похибкою:

- метод побудови дотиком відрізків;
- метод побудови частковим накладанням відрізків;

Розглянемо послідовно кожен із запропонованих методів з метою визначення найкращого серед них.

Метод зафарбовування трикутника за допомогою поділу його на чотири частини для найкращого, оптимального випадку вимагає щоб всі сторони трикутника мали парні значення БП та МП, тобто іншими словами приріст по X та Y для будь-якої пари точок має бути парним.

Як відомо, цілі числа мають такі властивості:

- а) парне \pm парне = парне
- б) непарне \pm непарне = парне
- в) парне \pm непарне = непарне

Сформуємо достатні вимоги (табл.1.1):

Таблиця 3.1 Достатні вимоги до вузлових точок полігональної сітки.

	для всіх точок X	для всіх точок Y
1	x - парне	y - парне
2	x - непарне	y - парне
3	x - парне	y - непарне
4	x - непарне	y - непарне

Отже при виконанні будь-якої з цих вимог для будь-якої пари точок полігональної сітки БП та МП прирости будуть парними.

Досягти достатніх умов можна записавши значення «0» – вирівнювання по парності, або «1» – вирівнювання по непарності, в молодший розряд X та Y координат для всіх точок координатної сітки. Ця операція нормалізації викличе спотворення у структурі зображення, так як змінюючи координати вузлових точок на одиницю, ми не змінюємо значень інтенсивності. Дискретність формуємих зображень, точність алгоритмів їх побудови та властивості сприйняття зображень оком людини дозволяють стверджувати, що операція нормалізації полігональної сітки не викличе суттєвих або помітних спотворень об'єктів.

У цьому випадку похибка розрахунків рівна похибці класичного методу рендеринга. Однак похибка нормалізації обмежує застосування даного методу для зображень з великою кількістю дрібних деталей, що між іншим є загальною проблемою багатьох алгоритмів рендеринга.

3.4 Висновки

1. Розроблено методи розпаралелення рендерингу Гуро:
 - метод зустрічної інтерполяції;
 - метод сегментного визначення інтенсивностей кольору;
 - метод визначення інтенсивностей кольору в двох сусідніх точках рядка пастеризації;
 - метод з використанням додаткової триангуляції.
2. Розроблено структурні схеми пристроїв.

4 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ЗАФАРБОВУВАННЯ

4.1 Інтерфейс програмного модуля для високопродуктивного зафарбовування Гуро

На рис. 4.1 зображено зовнішній вигляд інтерфейсу програми, який включає ряд кнопок.

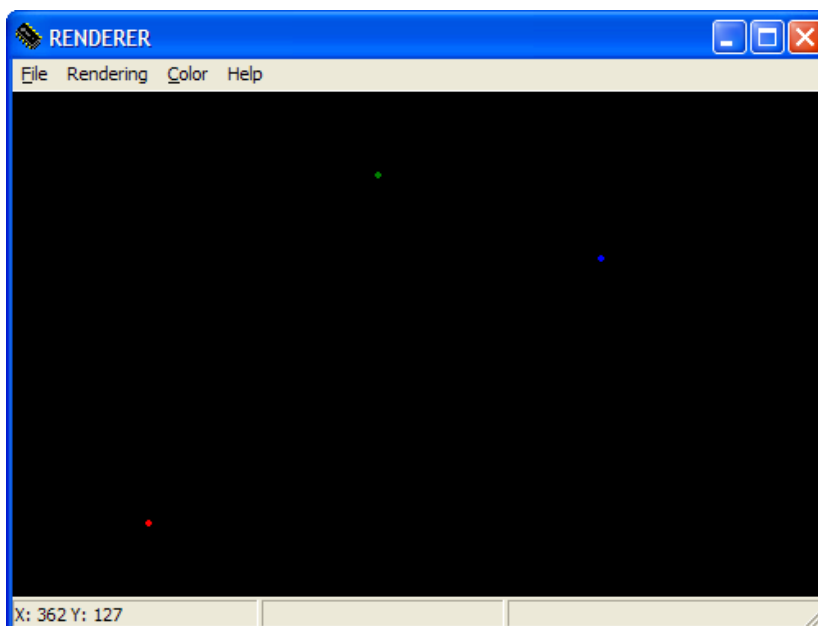


Рисунок 4.1 – Функціональні кнопки

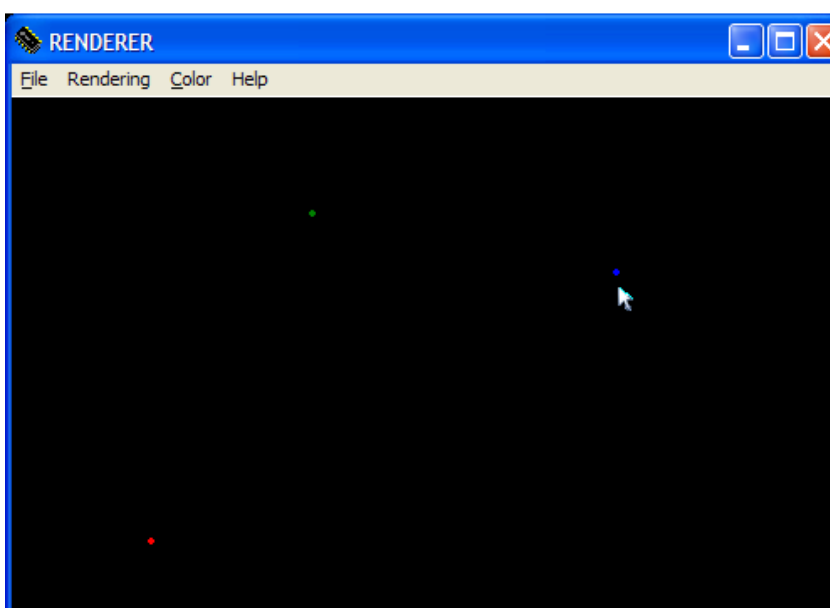


Рисунок 4.2 – Задання вершин трикутника

Кнопка File дозволяє ввести вихідний файл для реалізації зафарбовування. Передбачено можливість з використанням стрілки змінювати положення вершин трикутника шляхом їх перетягування.

З використанням кнопки Rendering можна вибрати метод зафарбовування класичний чи модифікований, що був розроблений в роботі.

Кнопка Color дозволяє вибрати колір вершин трикутника.

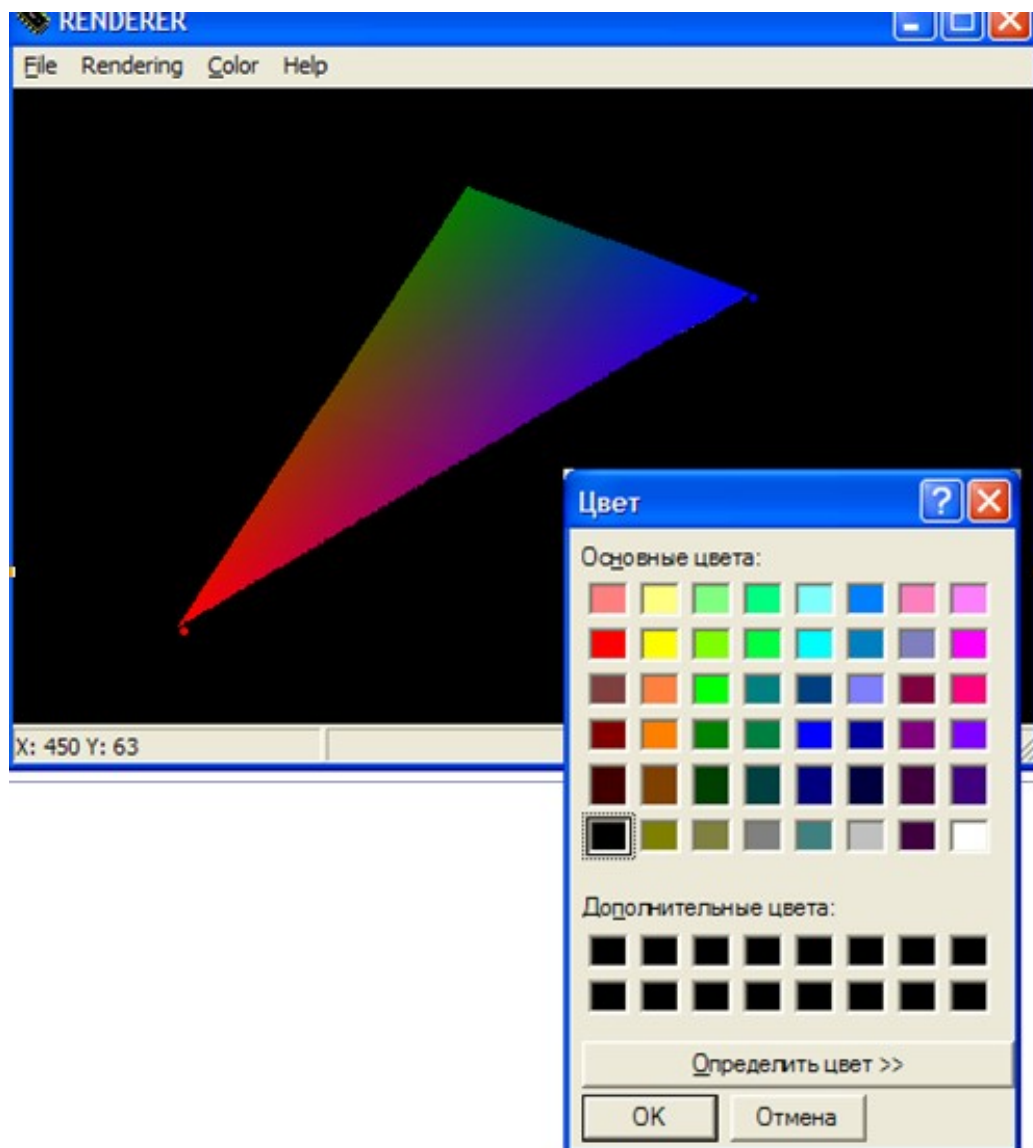


Рисунок 4.3 – Вибір кольорів для зафарбовування

Кнопка Help вносить пояснення у випадку, коли користувач програми має поточні зауваження

4.2 Програми для високопродуктивного зафарбовування

У розробленій програмі реалізовано метод зафарбовування Гуро з використанням додаткової тріангуляції трикутника (трикутник розбивається на чотири складових однакової площі).

Дана програма працює наступним чином: спочатку задаються координати вершин трикутника, координати джерела світла та спостерігача; задаються вектори нормалей у вершинах трикутника; задаються кольори матеріалу. Далі сортується вершини трикутника за координатою Y . Після сортування отримуємо, що найбільший приріст (дельта Y) буде у сторони AB (Рисунок 4.1), а точка C буде знаходитись, по координаті Y , між точками A і B .

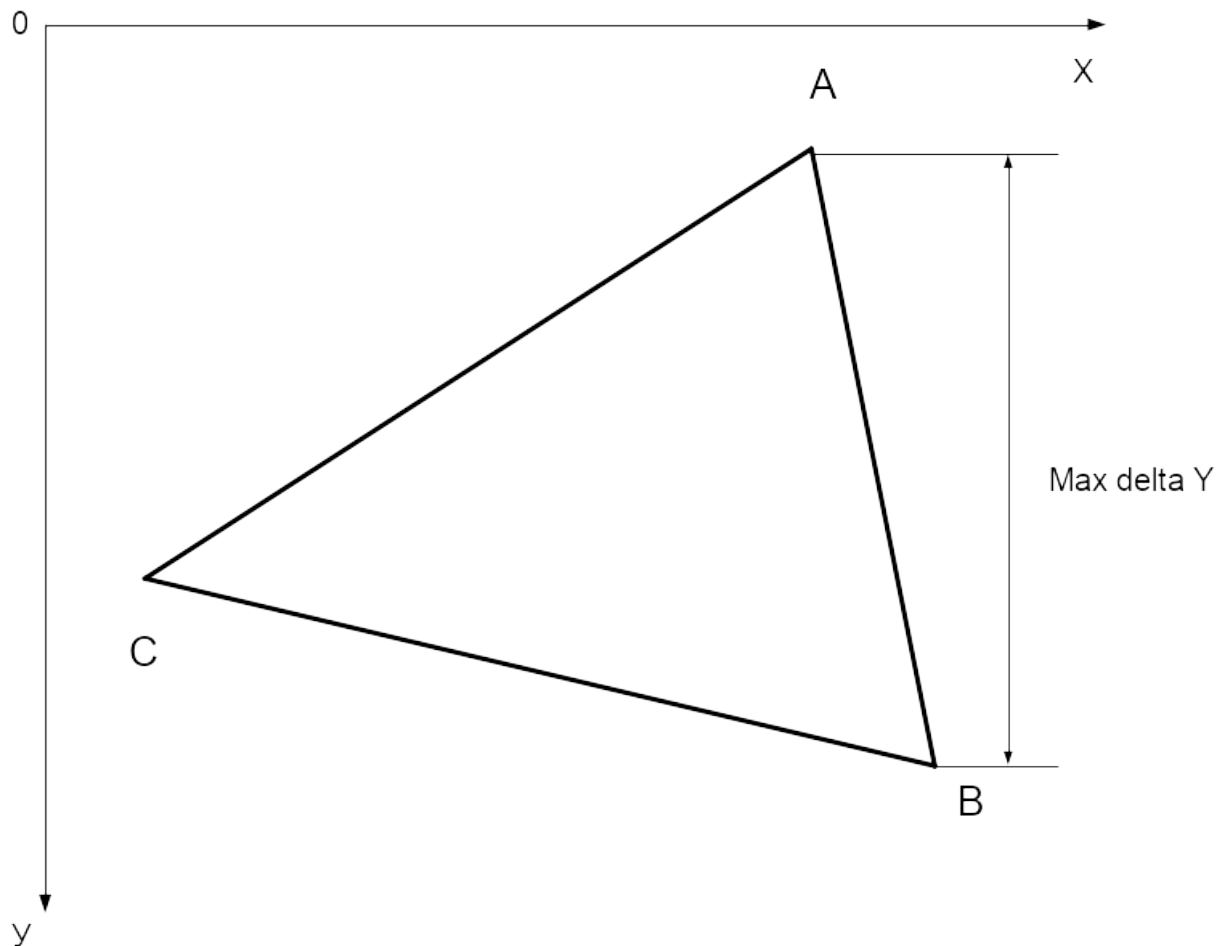


Рисунок 4.4 – вигляд трикутника

Далі обчислюємо ΔX та ΔY та визначаємо більший та менший приріст, визначаємо початкове значення оціночної функції (так для кожної сторони

трикутника). Визначаємо ΔZ та, поділивши його на більший приріст отримуємо крок зміни Z ; визначаємо крок зміни інтенсивностей кольору, поділивши різницю в значеннях нормалей на більший приріст (так для кожної сторони трикутника). Після цього запускаємо цикл на кількість ітерацій яка дорівнює більшому приросту сторони АВ. В цьому циклі визначаємо значення оціночної функції і відповідно до її значення змінюємо координати поточної точки (робимо горизонтальний, вертикальний чи діагональний кроки; та змінюємо координату на крок Z), далі, знаючи координати точки на стороні трикутника і координати джерела світла та спостерігача, визначаємо вектори світла та спостерігача і додаємо їх. В результаті попередніх дій отримаємо вектори W – суми векторів світла та спостерігача. Одночасно з цим змінюємо значення векторів нормалей до поверхні. Знаходимо кути: один кут між вектором нормалі та вектором W - ϕ , другий кут між вектором нормалі та вектором світла – β . Знаючи ці кути можна визначити колір даної точки за формулою

$$I = I_a + I_d + I_s = I_a + I_1 \cdot \cos \beta + I_1 \cdot \cos^n \phi$$

де I_a – інтенсивність випромінюваного поверхнею світла;

I_1 – інтенсивність світла, яке йде з джерела світла;

n – коефіцієнт спекулярності;

кути були описані вище.

Коли по найдовшій(по координаті Y) стороні алгоритм доходить до діагонального чи вертикального кроку переходимо на сторону АС або СВ в залежності від того на якій ітерації ми зараз знаходимось та яка координата Y у точки С. Колір кожної точки обчислюється так само як було зазначено вище(за тією ж самою формулою).

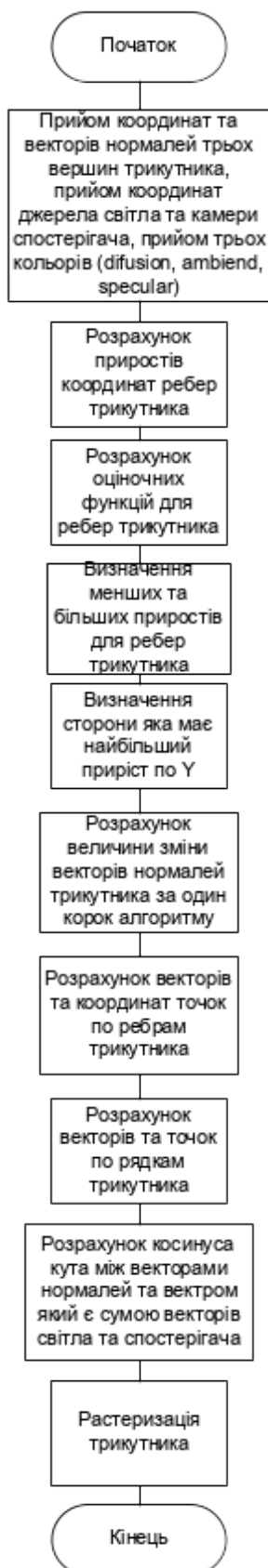


Рисунок 4.5 – Граф-схема алгоритму роботи програми

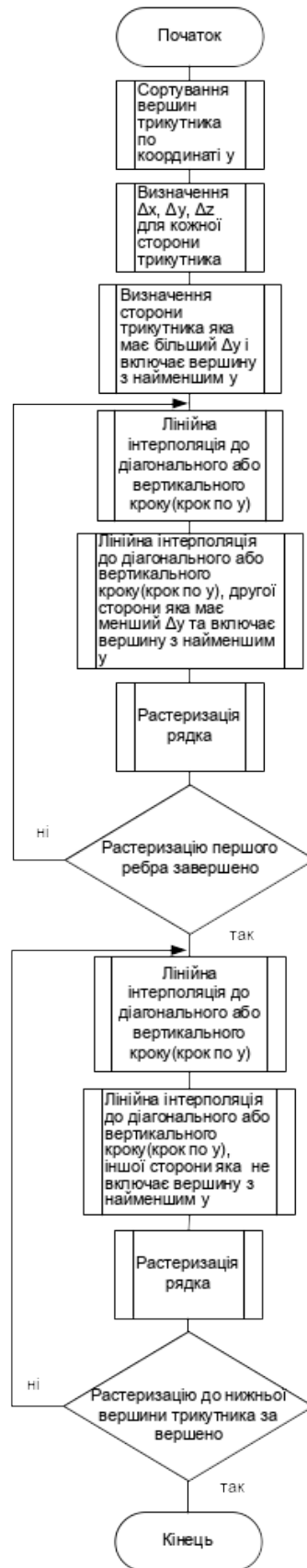


Рисунок 4.6 – Граф-схема алгоритму растризації трикутника

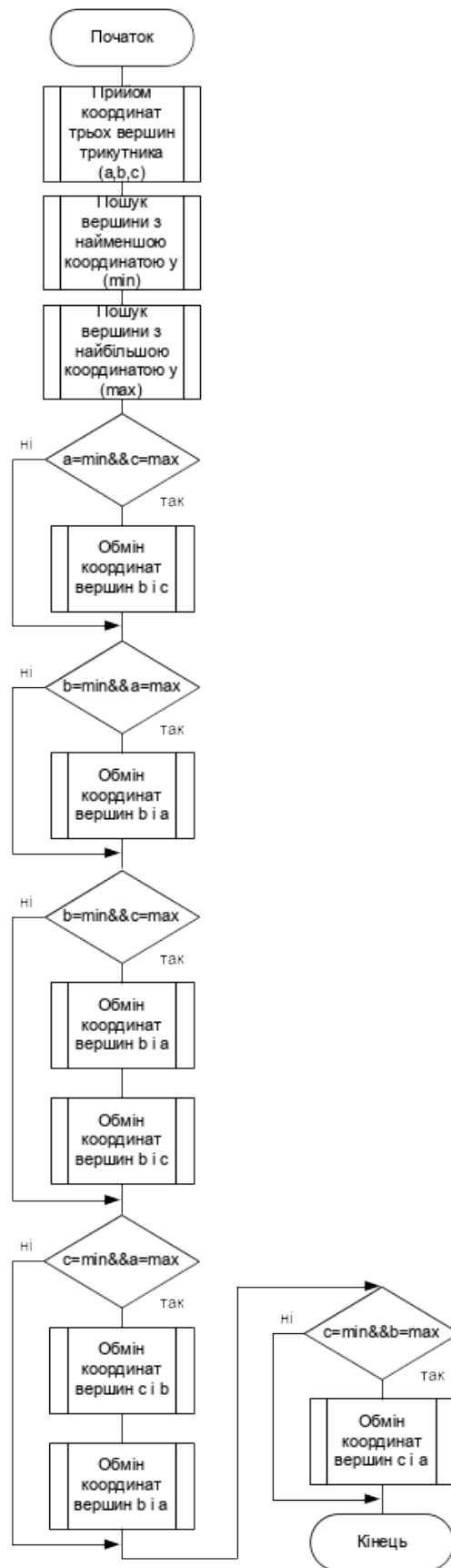


Рисунок 4.7 – Граф-схема сортування верши трикутника

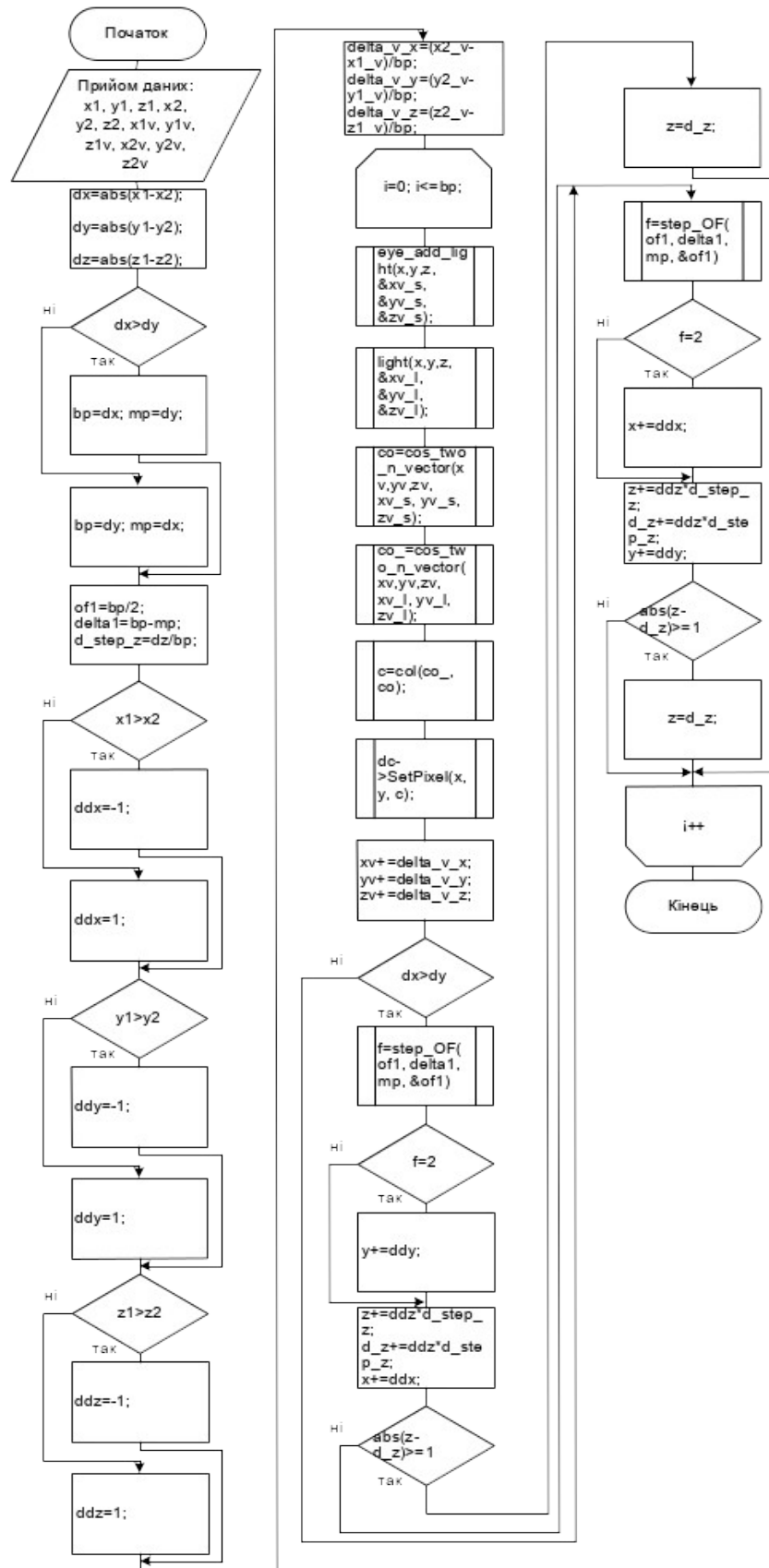


Рисунок 4.8 – граф-схема підпрограми інтерполяції відрізка прямої

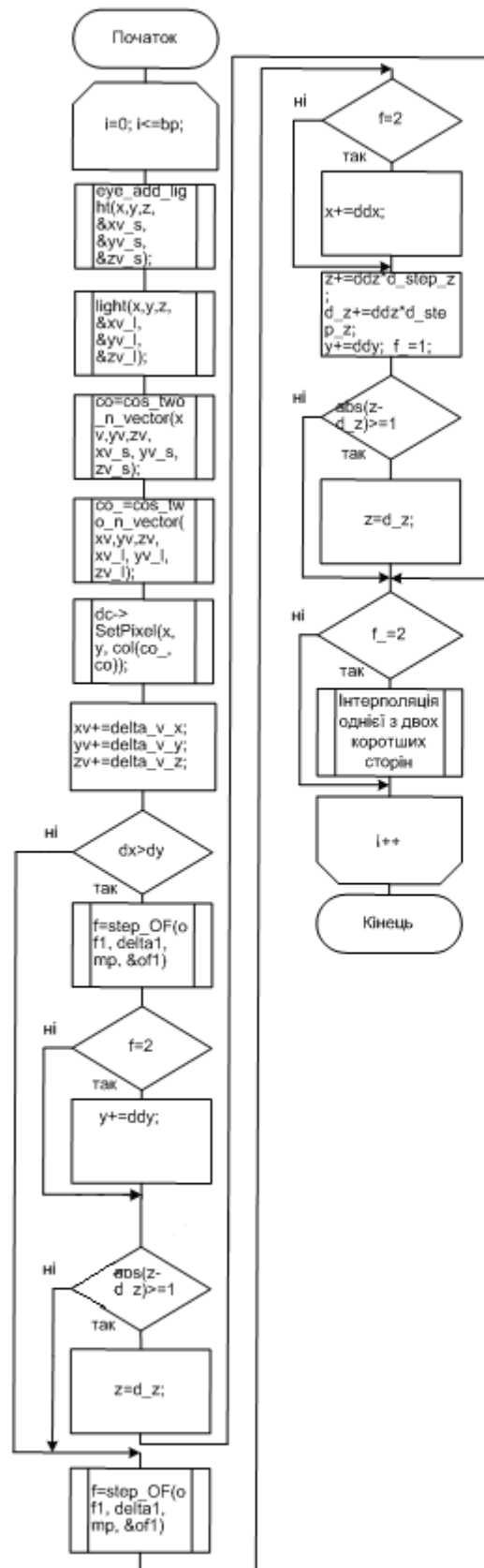


Рисунок 4.9 – Граф-схема алгоритму інтерполяції до діагонального кроку

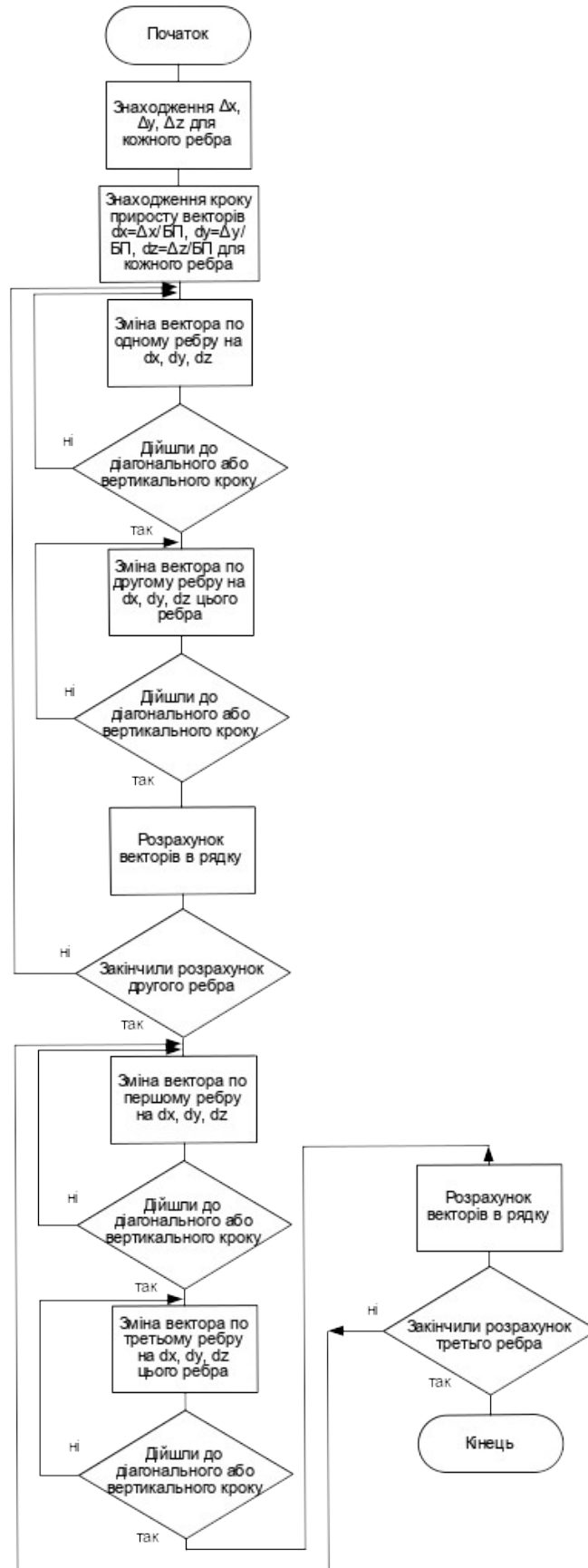


Рисунок 4.10 – Граф-схема розрахунку векторів нормалей

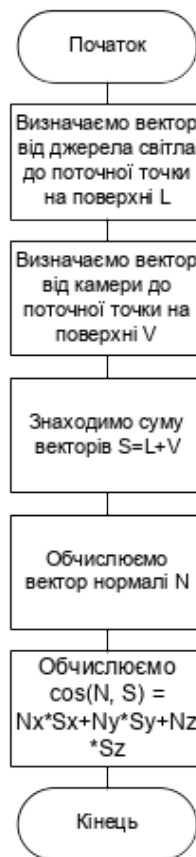


Рисунок 4.11 – Граф-схема алгоритму визначення $\cos\phi$

На рисунках 4.5 – 4.11 показані граф-схеми роботи окремих частин програми (Рисунок 4.5 – загальна схема алгоритму). На рисунку 4.6 зображена граф-схема растеризації трикутника, як можна побачити з цього рисунка растеризацію трикутника можна поділити на растеризацію однієї частини трикутника, а потім растеризації другої частини. На рисунку 4.7 відображена граф-схема сортування вершин трикутника(сортуються вершини по координаті Y тривіальним способом). На рисунку 4.8 – граф-схема підпрограми інтерполяції відрізка прямої – ця підпрограма використовується для інтерполяції рядків трикутника, але вона може і використовуватись для інтерполяції будь-якого відрізка прямої. Рисунок 4.9 відображає граф-схему інтерполяції до діагонального кроку тобто до того моменту коли алгоритм доходить до діагонального чи вертикального кроку і робить перехід до інтерполяції іншої сторони. На рисунках 4.10 та 4.11 показані граф-схеми визначення векторів нормалей та обрахунку $\cos\phi$ відповідно.

4.3 Розробка загального алгоритму

Алгоритм передбачає виконання таких дій.

Задаються тривимірні координати кожної вершини трикутника, вектор нормалі в кожній із цих вершин та інтенсивності трьох кольорів. Далі визначається вершина, яка має найменшу координату Y (буде виконуватись побудова з цієї вершини). Визначаємо всі прирости по ребрах трикутника (знаходимо ΔX , ΔY , ΔZ для кожної сторони трикутника). Наступним кроком є визначення більшого та меншого приростів для кожного ребра трикутника. Та із граней, що виходять з вершини з найменшим значенням координати Y і має більший приріст по значенню ΔY , буде основним ребром (це ребро будемо називати довшою стороною). Інша з цих двох граней буде коротшою стороною. Ребро, що залишилася будемо називати нижньою стороною. Інтерполяцію всіх відрізків прямих будемо виконувати за допомогою алгоритму Обідника-Петуха (цей алгоритм використовує метод оціночної функції). Для алгоритму Обідника-Петуха потрібно визначити початкові значення функції оцінювання $OF_0 = [BP/2]$ (ціла частина від ділення більшого приросту на два). Визначаємо початкові значення функції оцінювання для кожного з ребер трикутника. Для подальших розрахунків необхідно знати, на яку величину буде змінюватися приріст інтенсивності вздовж ребра. Для цього ділимо різницю інтенсивностей кольору в початковій та кінцевій точках вектора на більший приріст відповідного ребра.

Для визначення інтенсивностей кольору точок ребра трикутника до значення інтенсивностей кольору в початковій точці додається отриманий раніше приріст інтенсивностей.

Аналогічні дії виконуються і для рядків растеризації.

Для розрахунку координат точок по ребрам виконуємо інтерполяцію відрізка прямої довшої сторони. Коли формується крок, який містить переміщення по координаті Y (тобто вертикальний чи діагональний кроки), то переходимо до інтерполяції коротшої прямої і, аналогічно, коли здійснюємо крок, який містить переміщення по координаті Y , то переходимо до

інтерполяції рядка між довшою та коротшою прямою. Після закінчення інтерполяції рядка знову повертаємось до інтерполяції довшого вектора. Ці дії виконуємо, далі поки не закінчимо інтерполяцію частини трикутника, яка обмежена коротшим вектором, частиною довшого вектора та рядком, точки якого мають координату Y , що дорівнює координаті Y закінчення коротшого вектора. Інша частина трикутника інтерполюється аналогічним чином, але замість коротшого вектора використовується нижній вектор сторона, а довший вектор залишається. Інтерполяція кожного відрізка прямої виконується за кількість тактів, що дорівнює більшому приросту цього відрізка. При кожному кроці алгоритму змінюємо ту координату, якій відповідає більший приріст відрізка прямої, а іншу координату (маються на увазі двохвимірні координати) змінюємо залежно від того значення оціночної функції на цьому кроці алгоритму.

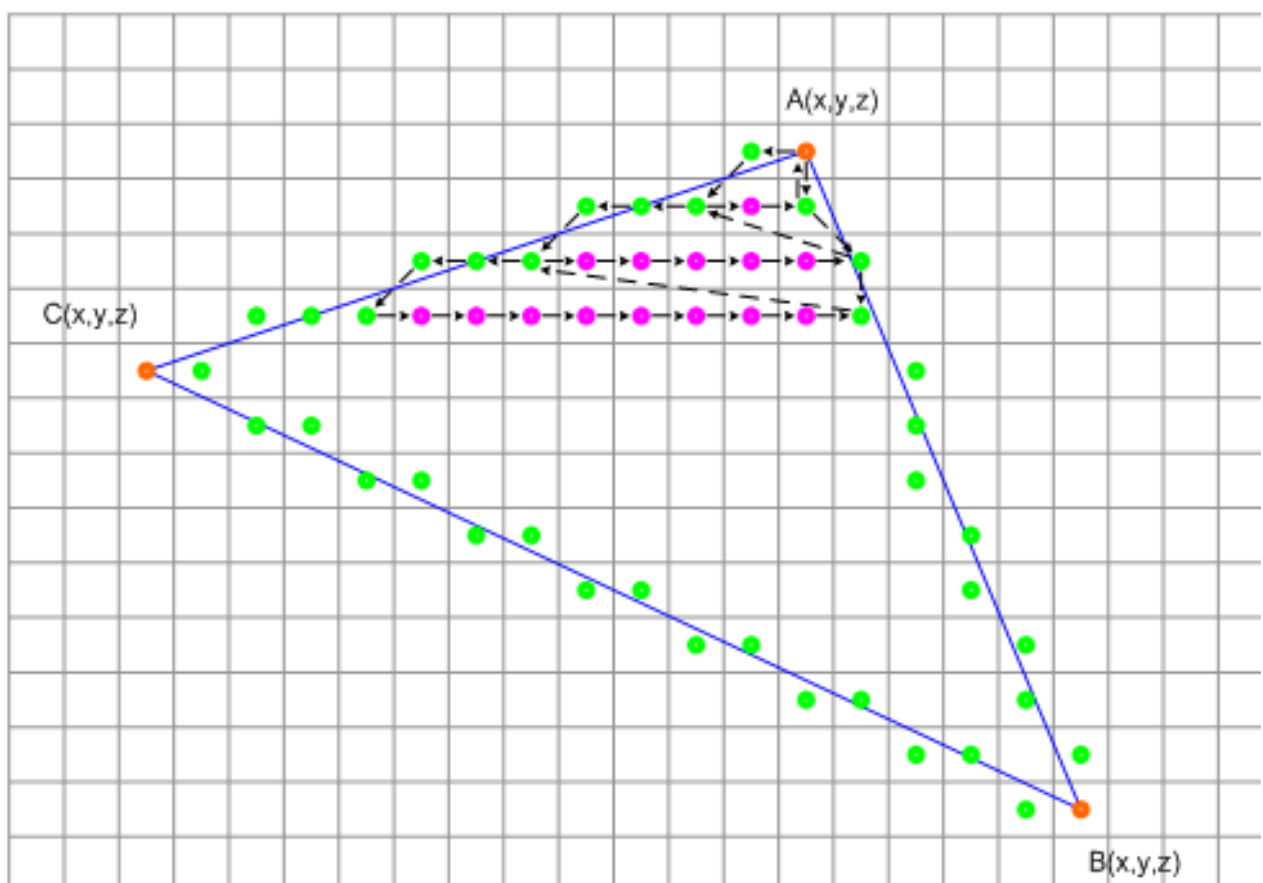


Рисунок 4.12 – Растеризація трикутника

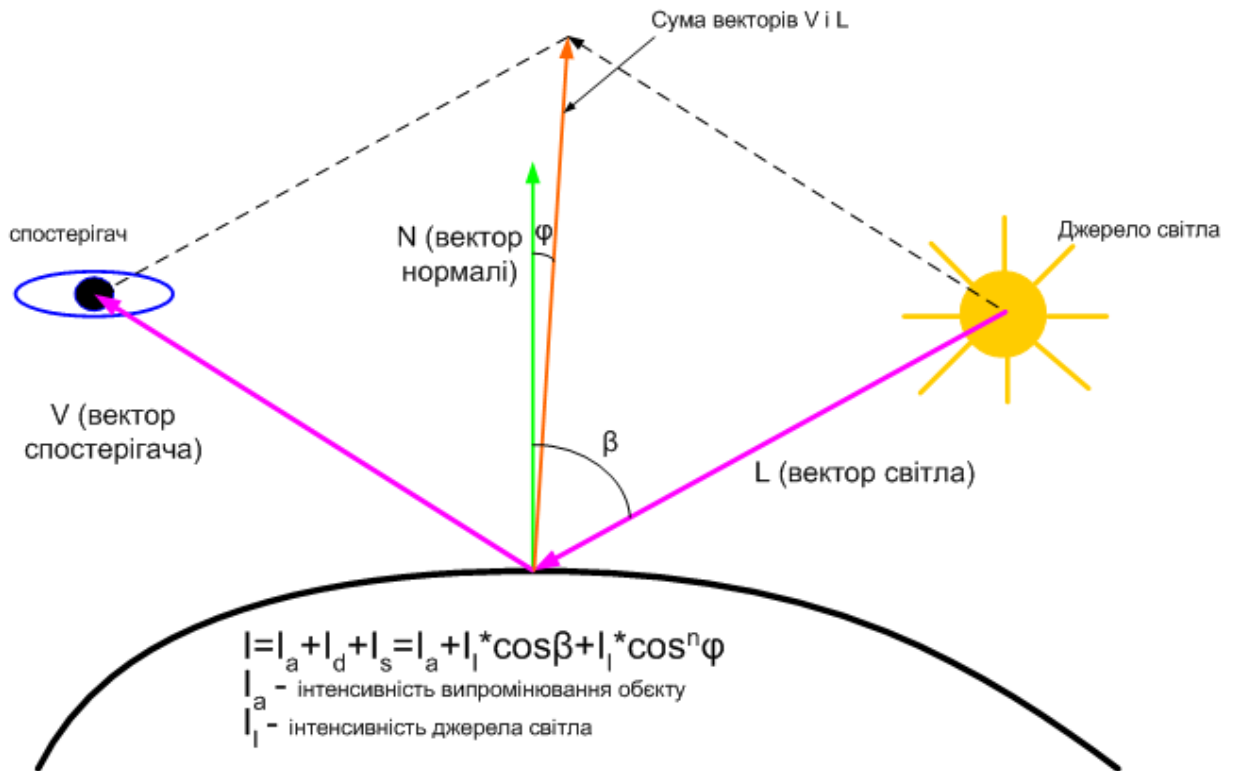


Рисунок 4.13 – Розрахунок інтенсивності кольору у вершинах трикутника

Наступною частиною алгоритму є растеризація трикутника, тобто визначення кольору кожної точки поверхні та відображення її на моніторі.

Розрахунки інтенсивності кольору виконуються для кожної компоненти (червоний - R, зелений - G, синій - B) окремо.

4.4 Висновки

1. Розроблено програмні засоби для рендерингу Гуро, які дозволили суттєво підвищити продуктивність формування графічних сцен.
2. Отримано оцінки продуктивності зафарбовування трикутника.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Технологічний аудит розроблених методу та програмних засобів для розпаралелення рендерингу Гуро

Як було підкреслено у попередніх розділах роботи, на сучасному етапі розвитку комп'ютерної графіки значна увага приділяється формуванню просторових зображень, які є найбільш реалістичними та інформативними, оскільки точно відтворюють конструктивні, візуальні та образотворчі особливості об'єкта.

Однією з основних та найбільш трудомістких процедур формування елементів зображення (тобто рендерингу) є процедура зафарбовування, згідно якої для кожної точки поверхні визначається інтенсивність кольору та екранні координати. При визначенні інтенсивності кольору точок зображення враховують розташування джерела світла та спостерігача, оптичні властивості матеріалу, спектральні характеристики джерела світла, кривизну поверхні тощо, у зв'язку з чим процедура зафарбовування характеризується значними обчислювальними витратами, що обумовлює необхідність розробки високопродуктивних методів, алгоритмів і структур.

Існуючі підходи до зафарбовування базуються на тому, що більшість поверхонь, включаючи криволінійні, можуть бути представлені полігональними мережами. Ці підходи вимагають найменших обчислювальних затрат, але практично невикористовуються, оскільки отримані зображення мають низьку якість.

Метод Гуро забезпечує формування більш реалістичних згладжених зображень, оскільки в ньому розраховуються значення інтенсивностей для полігональних вершин, які потім в процесі растеризації лінійно інтерполюються вздовж ребер і рядків сканування. Перспективним вважається і метод Фонга, в якому замість значень інтенсивності кольору інтерполюються вектори нормалей, які потім використовуються у функції тонування для обчислення інтенсивності кольору кожного елемента зображення. Метод Фонга

характеризується по відношенню до методу Гуро значно більшими обчислювальними витратами, однак при цьому досягається краща локальна апроксимація кривизни поверхні і, як наслідок, отримують більш реалістичні зображення.

Таким чином, на сьогодні існує проблема пошуку таких методів і засобів зафарбовування тривимірних графічних об'єктів, які б були більш прості за формою реалізації та мали більш просту апаратну реалізацію. Актуальність дослідження методів і засобів підвищення продуктивності зафарбовування тривимірних об'єктів обумовлена також стрімким розвитком виробництва графічних акселераторів, в яких широко використовуються методи формування реалістичних зображень.

Тому перед виконаною нами магістерській кваліфікаційній роботі було поставлене завдання розробити нові, оригінальні методи та програмні засоби, які б суттєво підвищили продуктивність зафарбовування тривимірних об'єктів у системах комп'ютерної графіки за рахунок удосконалення та розпаралелювання обчислювального процесу.

Для виконання поставленого завдання нами було: проведено аналіз існуючих методів і засобів зафарбовування поверхонь з метою визначення основних підходів до підвищення їх продуктивності; отримано більш прості аналітичні залежності по визначенню вихідних параметрів для процедур зафарбовування з метою спрощення обчислювального процесу; розроблено методи підвищення продуктивності зафарбовування Гуро і Фонга за рахунок удосконалення та розпаралелювання обчислювального процесу; розроблено та досліджено нову двопротеневу дистрибутивну функцію, яка порівняно з відомими має меншу обчислювальну складність; досліджено можливості використання при зафарбовуванні різних моделей освітлення залежно від наявності відблисків на поверхні об'єкта; розроблено рекомендації щодо проектування засобів для підсистем зафарбовування; розроблено алгоритми та структури для зафарбовування графічних об'єктів; розроблено спеціалізовані

програмні засоби для тестування методів зафарбовування тривимірних графічних об'єктів.

В результаті виконаної роботи було створено діючі програмні засоби для зафарбовування тривимірних графічних об'єктів у системах комп'ютерної графіки та розроблено структурні схеми пристроїв для апаратної реалізації зафарбовування в графічних акселераторах.

Для оцінювання технічного рівня та комерційного потенціалу нашої розробки проведемо її технологічний аудит. Для проведення технологічного аудиту було залучено трьох експертів, які мають необхідні знання і досвід роботи в досліджуваній області.

Оцінювання технічного рівня та комерційного потенціалу розробки здійснювалося за критеріями, які наведені у таблиці 5.1.

Таблиця 5.1 – Критерії оцінювання технічного рівня та комерційного потенціалу розробки і їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Критерій	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 роботоздатність 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

Продовження таблиці 5.1

Бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
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 витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та звільнення 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 використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Завершення таблиці 5.1

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
Бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
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 технічного рівня та комерційного потенціалу розробки зведено в таблицю 5.2.

Таблиця 5.2 – Результати оцінювання технічного рівня та комерційного потенціалу розробки

Критерії	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами:		
1	3	3	3
2	3	3	4
3	3	3	3
4	3	3	4
5	4	3	3
6	3	4	4
7	4	3	4

Завершення таблиці 5.2

8	4	3	4
9	4	3	3
10	3	4	4
11	3	3	3

12	3	3	3
Сума балів	СБ ₁ = 40	СБ ₂ = 38	СБ ₃ = 42

Для визначення технічного рівня та комерційного потенціалу розробки скористаємося таблицею 5.3, в якій представлено узагальнені результати можливого експертного оцінювання.

Таблиця 5.3 – Технічні рівні та комерційний потенціал розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Технічні рівні та комерційний потенціал розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Далі розрахуємо середньоарифметичне значення експертних оцінок:

Середньоарифметичне значення $\overline{СБ}$	$СБ_{сер} = \frac{СБ_1 + СБ_2 + СБ_3}{3} = \frac{40 + 38 + 42}{3} = \frac{120}{3} = 40.$
--	--

Порівнявши результати експертного оцінювання технічного рівня та комерційного потенціалу нашої розробки з даними таблиці 4.3, можна зробити висновок, що технічний рівень та комерційний потенціал нашої розробки знаходиться у межах «вище середнього - високий».

Такий досить високий технічний рівень та комерційний потенціал нашої розробки пояснюється тим, що нами розроблено нові модифікації методу Гуро, в яких уведено розпаралелювання обчислювального процесу зафарбовування поверхонь, що дозволяє підвищити швидкодію зафарбовування при збалансованому завантаженні зафарбовувачів

5.2 Розрахунок витрат на виконання роботи

Витрати на розробку методу та програмних засобів для розпаралелення рендерингу Гуро складаються із таких статей:

1). Основна заробітна плата Z_o виконавців, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t \text{ грн.}, \quad (5.1)$$

де M – місячний посадовий оклад конкретного виконавця, грн. В 2019 році величини окладів знаходиться в межах (4173...16550) грн/місяць;

T_p – число робочих днів в місяці; прийmemo $T_p = 21$ день;

t – число робочих днів роботи виконавців.

Зроблені розрахунки зведемо до таблиці 5.4:

Таблиця 5.4 – Основна заробітна плата виконавців

Найменування посади виконавця	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на оплату праці, грн
1. Науковий керівник роботи	15750	750	25 годин	3125
2. Студент-магістрант	1890	90	59	5310
3. Консультант з економічної частини	11550	550	2,5 годин	229
4. Інші консультанти	8400	400	2 дні	800
Загалом				$Z_o = 9464$ грн

2). Додаткова заробітна плата Z_d виконавців розраховується як (10...12)% від величини їх основної заробітної плати, тобто:

$$Z_d = (0,1...0,12) \cdot Z_o, \quad (5.2)$$

Для нашого випадку:

$$Z_d = 0,12 \times 9464 = 1136 \text{ грн.}$$

3). Нарахування на заробітну плату $H_{зп}$ виконавців розраховуються за формулою:

$$H_{зп} = (Z_o + Z_d) \cdot \frac{\beta}{100}, \quad (5.3)$$

де β – ставка обов'язкового єдиного внеску на державне соціальне страхування, %. $\beta = 22\%$. Тоді:

$$H_{\text{зп}} = (9464 + 1136) \times 0,22 = 2332 \text{ грн.}$$

4). Амортизація основних засобів A , які використовувались під час виконання магістерської роботи, може бути розрахована за формулою:

$$A = \frac{\text{Ц} \cdot H_a}{100} \cdot \frac{T}{12} \text{ грн,} \quad (5.4)$$

де Ц – загальна балансова вартість основних засобів, грн;

H_a – річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що $H_a = (5...25)\%$;

T – термін використання основних засобів, місяці.

Зроблені розрахунки зведено в таблицю 5.5.

Таблиця 4.5 – Розрахунок амортизаційних відрахувань

Найменування обладнання, приміщень тощо	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
1. Комп'ютерна техніка, принтери тощо	28000	20	2,4	1120
2. Приміщення університету та кафедри	11000	5	2,4	110
Всього				1230

5). Витрати на матеріали M розраховуються за формулою:

$$M = \sum_1^n H_i \cdot \text{Ц}_i \cdot K_i - \sum_1^n V_i \cdot \text{Ц}_v \text{ грн,} \quad (5.5)$$

де H_i – витрати матеріалу i -го найменування, кг; Ц_i – вартість матеріалу i -го найменування; K_i – коефіцієнт транспортних витрат, $K_i = (1,1...1,15)$; V_i – маса відходів матеріалу i -го найменування; Ц_v – ціна відходів матеріалу i -го найменування; n – кількість видів матеріалів.

6. Витрати на комплектуючі K розраховуються за формулою:

$$K = \sum_1^n H_i \cdot \text{Ц}_i \cdot K_i \text{ грн,} \quad (5.6)$$

де N_i – кількість комплектуючих i -го виду, шт.; C_i – ціна комплектуючих i -го виду; K_i – коефіцієнт транспортних витрат, $K_i = (1, 1 \dots 1, 15)$; n – кількість видів комплектуючих.

При проведенні досліджень та розробці методу та програмних засобів для розпаралелення рендерингу Гуро витрати на матеріали та комплектуючі склали приблизно 1050 грн.

7). Витрати на силову електроенергію V_e розраховуються за формулою:

$$V_e = \frac{V \cdot \Pi \cdot \Phi \cdot K_{\Pi}}{K_d}, \quad (5.7)$$

де V – вартість 1 кВт-год. електроенергії, в 2019 р. $V \approx 2,2$ грн/кВт;

Π – установлена потужність обладнання, кВт; $\Pi = 1,95$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

Приймемо, що $\Phi = 135$ годин;

K_{Π} – коефіцієнт використання потужності; $K_{\Pi} < 1 = 0,9$.

K_d – коефіцієнт корисної дії, $K_d = 0,75$.

Тоді витрати на електроенергію будуть дорівнювати:

$$V_e = \frac{V \cdot \Pi \cdot \Phi \cdot K_{\Pi}}{K_d} = \frac{2,2 \cdot 1,95 \cdot 135 \cdot 0,9}{0,75} = 694,98 \approx 695 \text{ грн.}$$

8). Інші витрати $V_{\text{інш}}$ (опалення, освітлення, прибирання тощо) можна прийняти як (100...300)% від суми основної заробітної плати виконавців:

$$V_{\text{інш}} = (0,1 \dots 3) \times Z_o. \quad (5.8)$$

Для нашого випадку отримаємо:

$$V_{\text{інш}} = 1,45 \times 9464 = 13723 \text{ грн.}$$

9). Сума всіх попередніх статей витрат складає витрати на виконання цієї роботи безпосередньо магістрантом – V .

$$V = 9464 + 1136 + 2332 + 1230 + 1050 + 695 + 13723 = 29630 \text{ грн.}$$

10). Загальні витрати Z_{AG} на завершення виконаної нами роботи з розробки методу та програмних засобів для розпаралелення рендерингу Гуро розраховуються за формулою:

$$Z_{\text{AG}} = \frac{B}{\beta}, \quad (5.9)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання нашої роботи.

Прийmemo, що $\beta \approx 0,85$ [7], оскільки робота практично готова до впровадження.

$$\text{Тоді: } Z_{\text{AG}} = \frac{29630}{0,85} = 34858,82 \text{ грн або приблизно 35 тис. грн.}$$

Тобто прогнозовані витрати на розробку методу та програмних засобів для розпаралелення рендерингу Гуро становлять приблизно 35 тис. грн.

5.3 Розрахунок економічного ефекту від можливої комерціалізації

Проведені дослідження показали, що щорічний попит на розробки, подібні до нашої, становить приблизно 40 шт., а ціна їх реалізації – 45 тис. грн.

Оскільки розроблені нами метод та програмні засоби для розпаралелення рендерингу Гуро мають набагато кращі функціональні можливості, то їх можна буде реалізовувати на ринку дещо дорожче (наприклад, за 50 тисяч грн, тобто на 5 тисяч грн дорожче).

Окрім того не викликає сумнівів, що наша розробка буде користуватися підвищеним попитом на ринку принаймні протягом 3-х років після її впровадження, тобто, якщо наша розробка буде впроваджена з 1 січня 2020 року (оскільки вона практично готова до впровадження), то її результати будуть виявлятися протягом 2020-го, 2021-го та 2022-го років.

Прогноз зростання попиту на нашу розробку складає по роках:

1-й рік після впровадження (2020 р.) – приблизно +25 шт.;

2-й рік після впровадження (2021 р.) – приблизно +35 шт.;

3-й рік після впровадження (2022 р.) – приблизно +50 шт.

Можливе збільшення чистого прибутку $\Delta\Pi_i$, що його може отримати потенційний інвестор від фінансування та впровадження нашої розробки, становитиме [7]:

$$\sum_1^n (\Delta C_0 \cdot N + C_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (5.10)$$

де ΔC_0 – зміна основного якісного показника від впровадження результатів розробки у цьому році. Таким показником є зміна ціни нової розробки відносно аналогів, грн.

Для нашого випадку це буде $\Delta C_0 = (50-45) = 5$ тис. грн, тобто наша розробка буде продаватися дещо дорожче, ніж подібні;

N – основний кількісний показник, який визначає обсяг діяльності у цьому році до впровадження результатів розробки; $N = 40$ шт.;

ΔN – покращення основного кількісного показника від впровадження результатів розробки. Таке покращення по роках становитиме, відповідно: +25 (2020 р.) шт.; +35 (2021 р.) шт. та +50 (2022 р.) шт.;

C_0 – основний якісний показник, який визначає обсяг діяльності (тобто ціну) у році після впровадження результатів розробки, грн; $C_0 = 50$ тис. грн;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки; $n = 3$ роки;

λ – коефіцієнт, який враховує сплату податку на додану вартість; $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати $\rho = (0,2...0,5)$; візьмемо $\rho = 0,45$;

u – ставка податку на прибуток. У 2019 році $u = 18\%$.

Тоді збільшення чистого прибутку для потенційного інвестора $\Delta\Pi_1$ протягом першого року від можливої реалізації нашої розробки (2020 р.) складатиме:

$$\Delta\Pi_1 = [5 \cdot 40 + 50 \cdot 25] \cdot 0,8333 \cdot 0,45 \cdot \left(1 - \frac{18}{100}\right) \approx 446 \text{ тис. грн.}$$

Величина чистого прибутку $\Delta\Pi_2$ для потенційного інвестора від можливого впровадження нашої розробки протягом другого (2021 р.) року складе:

$$\Delta\Pi_2 = [5 \cdot 40 + 50 \cdot 35] \cdot 0,8333 \cdot 0,45 \cdot \left(1 - \frac{18}{100}\right) \approx 600 \text{ тис. грн.}$$

Величина чистого прибутку $\Delta\Pi_3$ для потенційного інвестора від можливого впровадження нашої розробки протягом третього (2022 р.) року складе:

$$\Delta\Pi_3 = [5 \cdot 40 + 50 \cdot 50] \cdot 0,8333 \cdot 0,45 \cdot \left(1 - \frac{18}{100}\right) = 830 \text{ тис. грн.}$$

Далі розрахуємо теперішню вартість інвестицій PV , що повинні бути вкладені в нашу розробку: $PV = (2 \dots 5) \times 3_{AG}$.

Для нашого випадку $PV = (2 \dots 5) \times 35 = 5 \times 35 = 175$ тис. грн.

Розраховуємо абсолютний ефект вкладених інвестицій $E_{абс}$.

$$E_{абс} = ПП - PV, \quad (5.11)$$

де $ПП$ – приведена вартість всіх можливих чистих прибутків від можливого впровадження нашої розробки, грн;

PV – теперішня вартість інвестицій $PV = 175$ тис. грн.

Приведена вартість всіх чистих прибутків $ПП$ розраховується так:

$$ПП = \sum_1^t \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.12)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої роботи, грн;

t – період часу, протягом якого виявляються результати впровадженої роботи, роки. Для нашого випадку $t = 3$ роки;

τ – ставка дисконтування. Керуючись даними статистики за 2019 рік та прогнозами експертів, прийmemo ставку дисконтування $\tau = 0,12$ (12%);

t – період часу від моменту отримання прибутків до початку впровадження розробки.

Тоді приведена вартість всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження нашої розробки, складе:

$$ПП = \frac{446}{(1 + 0,12)^1} + \frac{600}{(1 + 0,12)^2} + \frac{830}{(1 + 0,12)^3} \approx 398 + 478 + 591 = 1467 \text{ тис. грн.}$$

Абсолютний ефект від можливого впровадження нашої розробки (при прогнозованому ринку збуту) за три роки (2020-2022 рр.) складе:

$$E_{абс} = 1467 - 175 = 1292 \text{ тис. грн. або } 430,67 \text{ тис. грн щорічно.}$$

Оскільки $E_{абс} > 0$, то вкладання коштів в нашу розробку може бути доцільним.

Далі розрахуємо внутрішню норму дохідності E_B вкладених коштів:

$$E_B = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1, \quad (5.13)$$

де $E_{\text{абс}}$ – абсолютний ефект вкладених коштів; $E_{\text{абс}} = 1292$ тис. грн;

PV – теперішня вартість початкових інвестицій $PV = 175$ тис. грн;

$T_{\text{ж}}$ – життєвий цикл розробки, роки. $T_{\text{ж}} = 4$ роки (з 2020 р. по 2022 р.).

Для нашого випадку:

$$E_B = \sqrt[4]{1 + \frac{1292}{175}} - 1 = \sqrt[4]{1 + 7,3828} - 1 = \sqrt[4]{8,3828} - 1 = 1,70 - 1 = 0,70 = 70\%.$$

Далі визначимо ту мінімальну дохідність, нижче за яку інвестор не буде вкладати кошти в нашу розробку. Мінімальна дохідність або мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau = d + f, \quad (5.14)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні $d = (0,15 \dots 0,50)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, $f = (0,05 \dots 0,5)$, але може бути і значно більше. Прийнемо, що $f = 50\%$.

Для нашого випадку отримаємо:

$$\tau_{\text{мін}} = 0,15 + 0,50 = 0,65 \text{ або } \tau_{\text{мін}} = 65\%.$$

Оскільки величина $E_B = 70\% > \tau_{\text{мін}} = 65\%$, то потенційний інвестор може бути зацікавлений у фінансуванні та подальшої комерціалізації нашої розробки.

Далі розраховуємо термін окупності коштів, вкладених у нашу розробку. Термін окупності $T_{\text{ок}}$ можна розрахувати за формулою:

$$T_{\text{ок}} = \frac{1}{E_B}. \quad (5.15)$$

Для нашого випадку термін окупності $T_{\text{ок}}$ коштів складе:

$$T_{\text{ок}} = \frac{1}{0,70} \approx 1,428 \text{ років,}$$

що свідчить про потенційну доцільність комерціалізації нашої розробки.

Результати виконаної економічної частини магістерської кваліфікаційної роботи зведено у таблицю:

Показники	Задані у ТЗ	Досягнуті у магістерській кваліфікаційній роботі	Висновок
1. Витрати на розробку методу та програмних засобів	Не більше 40 тис. грн	35 тис. грн.	Досягнуто
2. Абсолютний ефект від впровадження розроблених методу та програмних засобів тис. грн	не менше 400 тис. грн за рік	430,67 тис. грн щорічно	Виконано
3. Внутрішня норма дохідності інвестицій, %	не менше 65%	70%	Досягнуто
4. Термін окупності вкладених коштів, роки	до 3-х років	1,428 років	Виконано

5.4 Висновки

Таким чином, основні техніко-економічні показники розроблених нами методу та програмних засобів для розпаралелення рендерингу Гуро, визначені у технічному завданні, виконані.

ВИСНОВКИ

1. Уперше встановлено залежність приросту інтенсивностей кольору для точок поверхні, обмеженої трикутником, що дає можливість підвищити продуктивність зафарбовування та спростити апаратну реалізацію пристроїв зафарбовування.

2. Подальшого розвитку отримав метод зафарбовування Гуро, який відрізняється від відомого визначення в кожному такті не однієї, а відразу кількох інтенсивностей точок трикутника, що дозволяє підвищити продуктивність формування графічних сцен.

3. На основі отриманих в магістерській кваліфікаційній роботі теоретичних положень запропоновано алгоритми та розроблено програмні засоби зафарбовування у комп'ютерних системах візуалізації тривимірних зображень. Розроблено структурні схеми рендерів.

4. Отримані в магістерській кваліфікаційній роботі наукові та практичні результати можна використати для побудови високопродуктивних систем рендерингу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Авербух В. Л. К теории компьютерной визуализации / В. Л. Авербух // Вычислительные технологии. — 2005. — Том 10. — № 4. — С. 21—51.
2. Барладян Б. Х. Моделирование освещенности и синтез фотореалистичных изображений с использованием Интернет - технологий / Б. Х. Барладян, А. Г. Волобой, Н. И. Вьюкова, В. А. Галактионов // Программирование. — 2005. — № 5. — С. 66—80.
3. Баяковский Ю. М. Архитектура высокопроизводительных графических систем / Ю. М. Баяковский, В. А. Галактионов, А. Б. Ходулев. — ИПМ, 1998. — 26 с. — (Препринт № 3 / НАН России, Инст. прикл. мат. им. М. В. Келдыша; ИПМ 1998).
4. Вяткин С. И. Растеризационные методики и архитектуры систем визуализации реального времени / С. И. Вяткин, Б. С. Долговесов, В. М. Фомичев // Труды 17—й Междунар. конф. "Графикон—2007", 2007. — С. 164—169.
5. Игнатенко А. Современные технологии улучшения качества 3D—изображений / А. Игнатенко // Графика и Мультимедиа. — 2004. Режим доступа до журн. : <http://cgm.graphicon.ru/content/view/56/61/>.
6. Калютов А. В. Введение в фотореалистическую графику / А. В. Калютов. — СПб. : Политехника, 2006. — 118 с.
7. Козловський В. О. Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт. — Вінниця: ВНТУ, 2012.
8. Мальований О.Г. Комп'ютерна програма «Комп'ютерна програма для зафарбовування поверхонь тривимірних об'єктів за модифікованим методом Гуро» / О. Н. Романюк, О. Г. Мальований // Свідоцтво про реєстрацію авторського права на твір № 91846. — Київ : Міністерство економічного розвитку і торгівлі. — Дата реєстрації: 28.08.2019.
9. Мальований О.Г. Комп'ютерна програма «Комп'ютерна програма для адаптивної нормалізації векторів у рядку растеризації» / О.

- Н. Романюк, О. Г. Мальований // Свідоцтво про реєстрацію авторського права на твір № 91847. — Київ : Міністерство економічного розвитку і торгівлі. — Дата реєстрації: 28.08.2019.
10. Мальований О.Г. Комп'ютерна програма «Комп'ютерна програма для зафарбовування поверхонь тривимірних об'єктів за модифікованим методом Фонга» / О. Н. Романюк, О. Г. Мальований // Свідоцтво про реєстрацію авторського права на твір № 91848. — Київ : Міністерство економічного розвитку і торгівлі. — Дата реєстрації: 28.08.2019.
11. Методичні рекомендації з комерціалізації розробок, створених в результаті науково-технічної діяльності – К.: Наказ Державного комітету України з питань науки, інновацій та інформатики (Лист № 1/06-4-97 від 13.09.2010 р.)
12. Роджерс Д., Адамс Дж. Математические основы машинной графики / Д. Роджерс, Дж. Адамс : Пер. с англ. — М. : Мир, 2001. — 604 с.
13. Романюк А. Н. Алгоритмы рендеринга / А. Н. Романюк, Т. А. Замковой Компьютеры+программы. — 1998. — № 6. — С. 54—57.
14. Романюк А. Н. Основные процедуры 3D-конвейеризации / А. Н. Романюк // Труды филиала МВТУ им. Н. Э. Баумана в г. Калуге. Специальный выпуск: материалы международного симпозиума «Наука и предпринимательство, 2000. — С. 471—476.
15. Романюк О. Н. Високопродуктивні методи та засоби зафарбовування тривимірних графічних об'єктів. Монографія. / О. Н. Романюк, А. В. Чорний. - Вінниця : УНІВЕСУМ-Вінниця, 2006. — 190 с.
16. Романюк О. Н. Эффективная модель для відтворення спекулярної складової кольору / О. Н. Романюк // Проблеми інформатизації та управління: Збірник наукових праць. — 2007. — Випуск 2 (20). — С. 115—120.
17. Романюк О. Н. Комп'ютерна графіка. Навчальний посібник / О. Н. Романюк — Вінниця: ВДТУ, 2001. — 129 с.
18. Романюк О. Н. Метод зафарбовування тривимірних графічних об'єктів без нормалізації векторів нормалей / О. Н. Романюк, А. А. Шаманський //

- Інформаційні технології та комп'ютерна інженерія. — 2006. — № 2 (6). — С. 111—115.
19. Романюк О. Н. Метод прискороного зафарбовування тривимірних поверхонь з урахуванням їх локальної кривизни / О. Н. Романюк // Вісник Східноукраїнського національного університету. — 2008. — № 12 (130). — С. 166—172.
 20. Романюк О.Н. Методи підвищення продуктивності зафарбовування / О. Н. Романюк, О.Г. Мальований // 625. — 2007. — № 2. — С. 71—87.
 21. Технологии создания трехмерного изображения [Электронный ресурс] / Д. Чеканов — Режим доступа: <http://www.3dnews.ru/video/3dgraphics/>.
 22. Херн Д., Бейкер М. Компьютерная графика и стандарт OpenGL / Д. Херн, М. Бейкер. — М. : Издательский дом "Вильямс", 2005. — 1168 с.
 23. Хилл Ф. OpenGL. Программирование компьютерной графики / Ф. Хилл. — Спб. : Питер, 2002. — 1088 с.
 24. Цисарж В. В. Математические методы компьютерной графики / В. В. Цисарж., Р. И. Марусин. — К. : Факт, 2004. — 464 с.
 25. Шикин Е. В. Компьютерная графика. Полигональные модели / Е. В. Шикин, А. В. Боресков.— М. : ДИАЛОГ—МИФИ, 2001.— 464 с.
 26. Эгрон Ж. Синтез изображений. Базовые алгоритмы / Ж. Эгрон: Пер. с фр. — М. : Радио и связь, 2013. — 216 с.
 27. Эйнджел Э. Интерактивная компьютерная графика. Вводный курс на базе OpenGL. / Э. Эйнджел. — Вильямс, 2001. — 592 с.
 28. Akenine-Möller T., Hoffman N., Haines E. Real-Time Rendering / T. Akenine-Möller, N. Hoffman., E. Haines. — Wellesley: A. Peters, 2007. — 1045 p.
 29. Dempksi K. Advanced Lighting and Materials with Shaders / Kelly Dempksi, Emmanuel Viale. — Wordware Publishing, Ins. Plano, Texas, 2004. — 361 p.
 30. Farin G., Hansford D. Mathematical Principles for Scientific Computing and Visualization / G. Farin, D. Hansford // Peters, Ltd. 2008. — 275 p.

ДОДАТКИ

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
" ____ " _____ 2019 р.

Технічне завдання
на магістерську кваліфікаційну роботу «Розробка методу та програмних
засобів для розпаралелення рендерингу Гуро» за спеціальністю
121 – Інженерія програмного забезпечення

Керівник магістерської кваліфікаційної роботи:
_____ д. т. н., проф. О.Н.Романюк
" ____ " _____ 2019 р.

Виконав:
_____ студент гр. 2ПІ-18м О. Г. Мальований
" ____ " _____ 2019 р.

Вінниця – 2019 року

1. Найменування та галузь застосування

Магістерська кваліфікаційна робота: «Розробка методу та програмних засобів для розпаралелення рендерингу Гуро».

Галузь застосування - системи комп'ютерної графіки.

2. Підстава для розробки.

Підставою для виконання магістерської кваліфікаційної роботи (МКР) є індивідуальне завдання на МКР та наказ № ректора по ВНТУ про закріплення тем МКР.

3. Мета та призначення розробки.

Метою роботи є підвищення продуктивності засобів формування тривимірних зображень за рахунок розпаралелення обчислювального процесу.

Призначення роботи – розробка методу та програмних засобів для розпаралелення рендерингу Гуро

3 Вихідні дані для проведення НДР

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

1. Цисарж В. В. Математические методы компьютерной графики / В. В. Цисарж., Р. И. Марусин. — К. : Факт, 2004. — 464 с.
2. Романюк О. Н. Прискорене визначення дифузної складової кольору при зафарбовуванні тривимірних графічних фігур / О. Н. Романюк // Оптико-електронні інформаційно-енергетичні технології. — 2006. — № 2 (12). — С. 71—76.
3. Романюк О. Н. Метод підвищення реалістичності відтворення тривимірних графічних об'єктів / О. Н. Романюк // Інформаційні технології та комп'ютерна інженерія. — 2007. — № 1 (8). — С. 192—196.

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

4. Технічні вимоги

Базовий метод зафарбовування – метод Гуро; модель освітлення – модель Бліна; графічний режим – TrueColor; вихідні дані для зафарбовування – координати вершин трикутників, інтенсивності кольору у вершинах трикутника; максимальна розрядність для задання адрес вершин трикутників – 12; вихідні дані – кінцеве зображення, зафарбоване згідно методу Гуро в поєднанні з моделлю освітлення Бліна.

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 моделей рендерингу	07.10.2019 – 27.10-2019
2	Виведення математичних залежностей	28.10.2019 – 4.11.2019
3	Розробка методів розпаралелення рендерингу Гуро	5.11.2019 – 15.11.2019
4	Розробка програмних засобів	16.11.2019 – 30.11.2019
5	Економічна частина	4.11.2019 – 14.11.2019

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. Ілюстративний матеріал

ІЛЮСТРАТИВНИЙ МАТЕРІАЛ ДО ЗАХИСТУ МАГІСТЕРСЬКОЇ
КВАЛІФІКАЦІЙНОЇ РОБОТИ

Завідувач кафедри ПЗ, д. т. н., професор _____ О. Н. Романюк

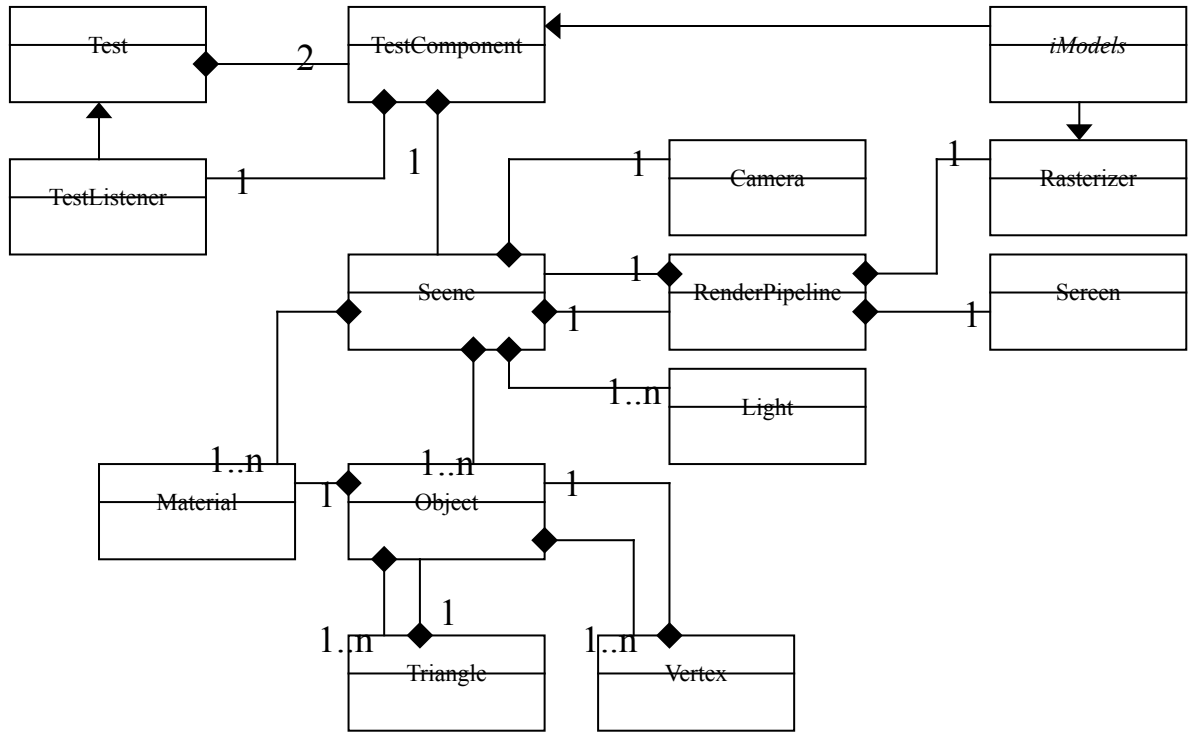
Науковий керівник, д. т. н., проф кафедри ПЗ _____ О. Н. Романюк

Рецензент, к.т.н, доцент кафедри ПЗ _____ В. В. Колодний

Нормоконтроль, д. т. н., проф. кафедри ПЗ _____ В. В. Колодний

Виконавець, студент групи 2ПІ-18м _____ О. Г. Мальований

Додаток В. Модель взаємозв'язку програмних модулів



Додаток Г. Час формування тестових фігур з використання методу Гуро

Метод Гуро	Базовий метод	Тестові фігури					
		Сфера			Чайник		Робот
		Час Зафарбовування	Кількість кадрів в секунду	NMSE	Зафарбовування,	Кількість кадрів	зафарбовування,
4	8						
44	34						
0,00058	–						
10	17						
32	27						
0,00086	–						
15	22						
30	25						
0,00095	–						

Додаток Д. Модуль “Gouraud Rasterizer”. *java*

```
public class GouraudRasterizer 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;

    public GouraudRasterizer()
    {
    }

    protected void initRender()
    {
        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);  
}
```

```
protected void prerenderFirstPart()
```

```
{  
    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;
    }
}

protected void prerenderSecondPart()
{
    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;
```

```
}
```

```
}
```

```
protected void prerenderLine()
```

```
{
```

```
renderLine();
```

```
lir += liir;
```

```
lig += liig;
```

```
lib += liib;
```

```
rir += riir;
```

```
rig += riig;
```

```
rib += riib;
```

```
}
```



```

public String getRasterizerName()
{
    return "Gouraud shading";
}

protected void renderLine()
{
    int cdx = xR - xL;
    if (cdx == 0) return;

    float iir = (rir - lir) / cdx;
    float iig = (rig - lig) / cdx;
    float iib = (rib - lib) / cdx;

    float cir = lir;
    float cig = lig;
    float cib = lib;

    for (x=xL;x<xR;x++)
    {
        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;
        }
        cir += iir;
    }
}

```

```

        cig += iig;
        cib += iib;
    }
}

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(0xFFFFFFFF, (int)(angle*255));
    c = idx3d_Color.add(c, c2);
    return c;
}

protected void beforeRenderTriangle()
{
}
}

```

Додаток Е. Лістинг головного модуля програми

Main.java

```
package phong;
import java.awt.event.ComponentAdapter;
import java.awt.event.ComponentEvent;
import java.awt.event.WindowAdapter;
import javax.swing.JPanel;
import phong.Ramka;
public class Main {
    public static void main(String[] args) {

        final ObjectCenter OC = new ObjectCenter();

        final Ramka ramka = new Ramka();
        final JPanel panel = ramka.getGraphicPanelContainer();

        GraphicPanel gP = new GraphicPanel(OC, ramka, panel.getWidth(),
panel.getHeight() );
        panel.add( gP );

        ramka.setGraphicPanel(gP);
        ramka.setLocationRelativeTo(null);
        ramka.setVisible(true);

        final Animator animator = new Animator(OC, gP);
        Thread thread = new Thread(animator);
        thread.start();
        ramka.setAnimator( animator );
```

```

ramka.addComponentListener( new ComponentAdapter()
    {
        public void componentResized(ComponentEvent e)
        {
            System.out.println("Okienko zmienilo rozmiar");
            panel.removeAll();
            GraphicPanel gP = new GraphicPanel(OC,
ramka, panel.getWidth(), panel.getHeight() );
            panel.add( gP );
            animator.setGraphicPanel( gP );
            ramka.setGraphicPanel( gP );
        }
    }
);
}
}

```

Renderer.java

```

package graph;
import geom.*;
import java.awt.Color;
import java.awt.Graphics2D;
import java.awt.geom.Ellipse2D;
public class Renderer
{
    Vec3d observer;
    boolean flipX = false;

```

```

boolean flipY = false;
double kRatio = 1;
double ambientLightFactor = 0.5d;
double diffuseLightFactor = 0.5d;
double specularLightFactor = 0.5d;

int n = 100;
Light[] lights;
int lnum;
final int maxLNum = 10;

int maxX;
int maxY;

public Renderer(int _maxX, int _maxY)
{
    observer = new Vec3d(0,0,-600);
    maxX = _maxX;
    maxY = _maxY;
    lights = new Light[maxLNum];
    lnum = 0;
        System.out.println("Stworzyłem renderer  maxX="+maxX+"
maxY="+maxY);
}
public void setAmbientLightFactor(double alf)
{
    ambientLightFactor = alf;
}
public void setDiffuseLightFactor(double dlf)
{

```

```

    diffuseLightFactor = dlf;
}
public void setSpecularLightFactor(double slf)
{
    specularLightFactor = slf;
}
public void setN(int n)
{
    this.n = n;
}
public void setKRatio(double k)
{
    kRatio = k;
}

public void renderWireFace(Graphics2D g2d, Body body, Face face)
{
    face.computeNormal();
    face.computeCenter();
    Vec3d followFaceCenter = new Vec3d( face.getCenter() );
    followFaceCenter.sub( observer );
    followFaceCenter.normalize();

    double dotProd = face.getNormal().getScalarProduct( followFaceCenter );
    if ( dotProd >= 0 && !face.isDoubleSide() )
        return;

    double x0 = getScreenX( body.vertices[face.getVertexIndex(0)] );
    double y0 = getScreenY( body.vertices[face.getVertexIndex(0)] );
    double x1 = getScreenX( body.vertices[face.getVertexIndex(1)] );

```

```

double y1 = getScreenY( body.vertices[face.getVertexIndex(1)] );
double x2 = getScreenX( body.vertices[face.getVertexIndex(2)] );
double y2 = getScreenY( body.vertices[face.getVertexIndex(2)] );
double red = 0, green = 0, blue = 0;

g2d.drawLine( (int)x0, (int)y0, (int)x1, (int)y1 );
g2d.drawLine( (int)x2, (int)y2, (int)x1, (int)y1 );
g2d.drawLine( (int)x0, (int)y0, (int)x2, (int)y2 );
}
public void renderSolidFace(Graphics2D g2d, Body body, Face face)
{
    face.computeNormal();
    face.computeCenter();
    Vec3d followFaceCenter = new Vec3d( face.getCenter() );
    followFaceCenter.sub( observer );
    followFaceCenter.normalize();

    double dotProd = face.getNormal().getScalarProduct( followFaceCenter );
    if ( dotProd >= 0 && !face.isDoubleSide() )
        return;

    double x0 = getScreenX( body.vertices[face.getVertexIndex(0)] );
    double y0 = getScreenY( body.vertices[face.getVertexIndex(0)] );
    double x1 = getScreenX( body.vertices[face.getVertexIndex(1)] );
    double y1 = getScreenY( body.vertices[face.getVertexIndex(1)] );
    double x2 = getScreenX( body.vertices[face.getVertexIndex(2)] );
    double y2 = getScreenY( body.vertices[face.getVertexIndex(2)] );

    int[] xCoords = {(int)x0,(int)x1,(int)x2};
    int[] yCoords = {(int)y0,(int)y1,(int)y2};

```

```

Vec3d obsNormal = new Vec3d( 0.0, 0.0, 1.0 );

double red = 0, green = 0, blue = 0;

for( int i=0; i<lnum; i++)
    if ( lights[i].isEnabled() )
        {
            Color color = lights[i].getColor();

            Vec3d tmp = new Vec3d( face.getCenter() );
            tmp.sub( lights[i].getPosition() );
            tmp.normalize();

            dotProd = face.getNormal().getScalarProduct( tmp );

            double beta = Math.acos( face.getNormal().getScalarProduct(tmp) );
            Vec3d tmp2 = face.getNormal().getCrossProduct(tmp);

            Vec3d R = new Vec3d( tmp );
            R.rotateAV(tmp2, beta*2);

            double dotProd2 = Math.pow( obsNormal.getScalarProduct( R ), n );
            if ( dotProd<0 )
                {
                    red += color.getRed() * Math.abs( dotProd ) * diffuseLightFactor
                        + color.getRed() * Math.abs( dotProd2 ) * specularLightFactor;
                    green += color.getGreen() * Math.abs( dotProd ) *
diffuseLightFactor

```



```

        + color.getGreen() * Math.abs( dotProd2 ) *
specularLightFactor;
        blue += color.getBlue() * Math.abs( dotProd ) *
diffuseLightFactor
        + color.getBlue() * Math.abs( dotProd2 ) *
specularLightFactor;
    }
}

```

```

Color color = face.getColor();
red += color.getRed() * ambientLightFactor;
green += color.getGreen() * ambientLightFactor;
blue += color.getBlue() * ambientLightFactor;
if ( red>255 ) red = 255;
if ( green>255 ) green = 255;
if ( blue>255 ) blue = 255;
color = new Color( (int)red, (int)green, (int)blue );
g2d.setColor( color );
g2d.fillPolygon(xCoords, yCoords, 3);
}
public void renderVertex(Graphics2D g2d, Vertd vertd)
{
    g2d.draw( new Ellipse2D.Double(getScreenX(vertd), getScreenY(vertd),
0, 0) );
}
public void renderVertexBody(Graphics2D g2d, Body body)
{
    for(int i=0; i<body.vnum; i++)
        renderVertex( g2d, body.vertices[i] );
}

```

```

public void renderWireBody(Graphics2D g2d, Body body)
{
    for(int i=0; i<body.fnum; i++)
        renderWireFace( g2d, body, body.faces[i] );
}

public void renderSolidBody(Graphics2D g2d, Body body)
{
    for(int i=0; i<body.fnum; i++)
        renderSolidFace( g2d, body, body.faces[i] );
}

private double dostosuj_X(double x)
{
    if( !flipX )
        return x+maxX/2;
    else
        return (x+maxX/2)*-1 +maxX;
}

private double dostosuj_Y(double y)
{
    if( !flipY )
        return ( y*kRatio+maxY/2 )*-1 +maxY ;
    else
        return y+maxY/2;
}

public double getScreenX(Vec3d vec3d)
{
    double d = observer.z * -1;
    double xp = vec3d.x;
    double yp = vec3d.y;
    double zp = vec3d.z;
}

```

```

    double xe = (xp*d)/(zp+d);
    return dostosuj_X( xe );
}

public double getScreenY(Vec3d vec3d)
{
    double d = observer.z * -1;
    double xp = vec3d.x;
    double yp = vec3d.y;
    double zp = vec3d.z;

    double ye = (yp*d)/(zp+d);
    return dostosuj_Y( ye );
}

    public void addLight(Light light)
{
    if ( lnum<maxLNum )
    {
        lights[lnum] = light;
        lnum++;
    }
}

public Light getLight(int n)
{
    if ( n>=0 && n<lnum )
    {
        return lights[n];
    }
    else return null;
}
}

```

Додаток Є. Графічний матеріал

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Мальований Олександр Григорович

Розробка методу та програмних засобів для розпаралелення рендерингу Гуро

Науковий керівник
д.т.н., проф. Романюк О.Н.

Вінниця – 2019

1

Галузі застосування та перспективи розвитку графічних засобів

 <p>У 2018 році реалізовано понад 450 млн. графічних процесорів</p>	 <p>. У найближчі 5 років сумарний продаж ігор для мобільних телефонів збільшиться у 18 разів, а прибуток від їх реалізації досягне 17,5 млрд. дол.</p>	 <p>У минулому році продано 35 млн. кишенькових комп'ютерів. До 2020 р. щорічний продаж нетбуків зросте до 139 млн. одиниць</p>
 <p>У 2018 році реалізовано понад 3 млн. графічних станцій</p>	 <p>Оборот у галузі ігрових консолей склав у 2018 році 17,94 млрд. дол</p>	 <p>Відеокарти</p>
 <p>У 2018 році у світі реалізовано 302 млн. ноутбуків, зокрема, в Україні – 720 тис. шт.</p>	 <p>Архітектури AMD Fusion, Lagabee</p>	 <p>У бортових системах прогнозується широке використання комп'ютерної графіки</p>

2

Мета та завдання досліджень

Метою магістерської кваліфікаційної роботи є підвищення продуктивності формування графічних сцен за рахунок розпаралелення рендерингу за методом Гуро.

Основними задачами дослідження є:

1. Провести аналіз існуючих методів і засобів зафарбовування поверхонь тривимірних об'єктів.
2. Виведення аналітичної залежності приросту інтенсивностей кольору для визначення інтенсивностей кольору ділянки поверхні, обмеженої трикутником;
3. Розробити методи підвищення продуктивності рендерингу Гуро за рахунок розпаралелення. Розробити програмні та апаратні компоненти на основі запропонованих методів;
4. Провести експериментальні дослідження розроблених засобів зафарбовування.

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

Предмет дослідження – методи та засоби формування тривимірних графічних об'єктів.

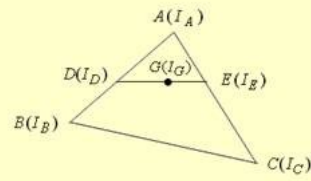
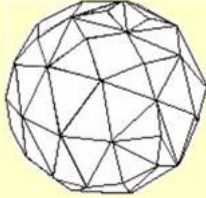
Методи дослідження. У процесі досліджень використовувались: теорія чисел та чисельних методів, методи аналітичної геометрії для розробки методів рендерингу тривимірних об'єктів; комп'ютерне моделювання для аналізу та перевірки отриманих теоретичних положень.

Приклади зафарбовування сфери

Метод однотонного зафарбовування

Метод зафарбовування за Гуро

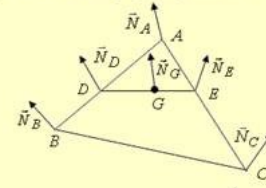
Метод зафарбовування за Фонгом



$$I_D = I_A \frac{y_D - y_A}{y_B - y_A} + I_B \left(1 - \frac{y_D - y_A}{y_B - y_A} \right)$$

$$I_E = I_A \frac{x_E - x_A}{x_C - x_A} + I_C \left(1 - \frac{x_E - x_A}{x_C - x_A} \right)$$

$$I_G = I_D \frac{x_G - x_D}{x_B - x_D} + I_E \left(1 - \frac{x_G - x_D}{x_B - x_D} \right)$$



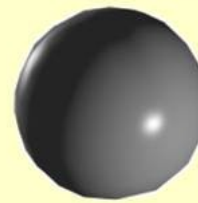
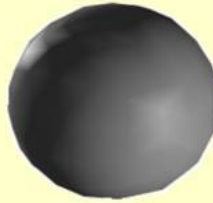
$$x_G \vec{i} = x_D \frac{x_G - x_D}{x_B - x_D} \vec{i} + x_E \left(1 - \frac{x_G - x_D}{x_B - x_D} \right) \vec{i}$$

$$y_G \vec{j} = y_D \frac{x_G - x_D}{x_B - x_D} \vec{j} + y_E \left(1 - \frac{x_G - x_D}{x_B - x_D} \right) \vec{j}$$

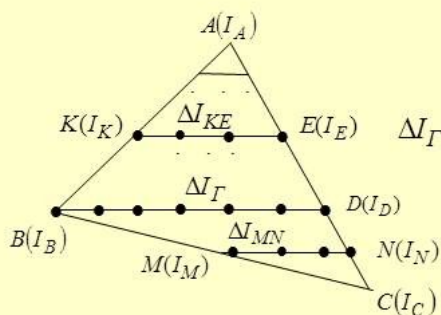
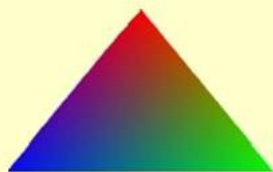
$$z_G \vec{k} = z_D \frac{x_G - x_D}{x_B - x_D} \vec{k} + z_E \left(1 - \frac{x_G - x_D}{x_B - x_D} \right) \vec{k}$$

$$\vec{N}_G = \frac{x_G \vec{i} + y_G \vec{j} + z_G \vec{k}}{\sqrt{x_G^2 + y_G^2 + z_G^2}}$$

$$I_G = f(\vec{N}_G, \vec{L}, \vec{V}, n, k_s, k_d, I_i^x)$$



Властивість сталості приросту інтенсивності кольору

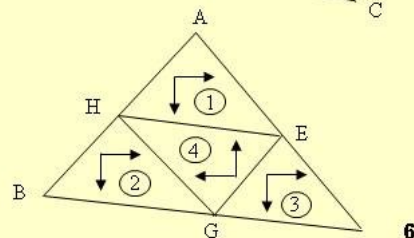
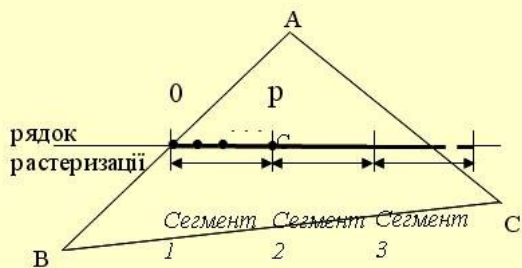
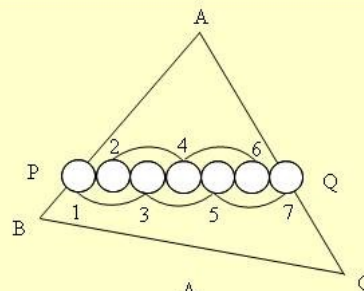
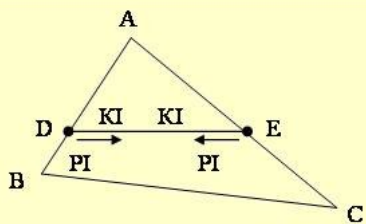


$$\Delta I_\epsilon, \Delta I_\epsilon - const$$

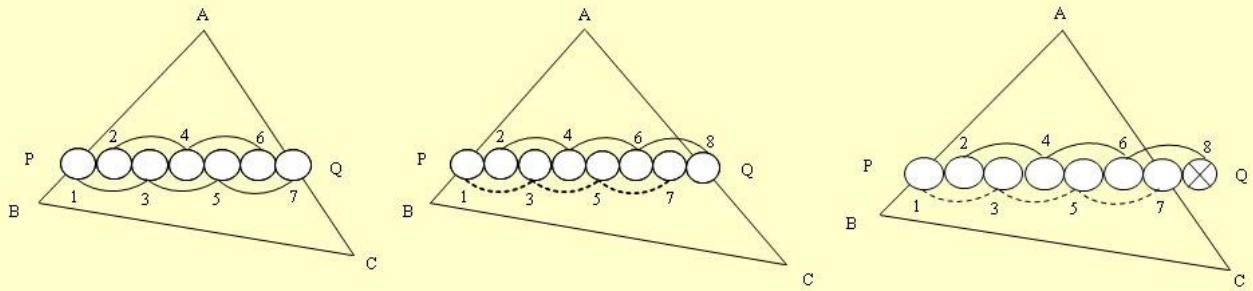
$$\Delta I_{KE} = \Delta I_{MN} = \Delta I_\Gamma - const$$

$$\Delta I_\Gamma = \frac{(I_C - I_A) \cdot \Delta y_{BC} - (I_C - I_B) \cdot \Delta y_{AC}}{\Delta x_{AC} \cdot \Delta y_{BC} - \Delta x_{BC} \cdot \Delta y_{AC}}$$

$$\Delta I_B = (I_B - I_C + \Delta I_\Gamma \Delta x_{BC}) / \Delta y_{BC}$$



Метод підвищення продуктивності зафарбовування Гуро, оснований на незалежному формуванні парних і непарних точок ряду rasterизації

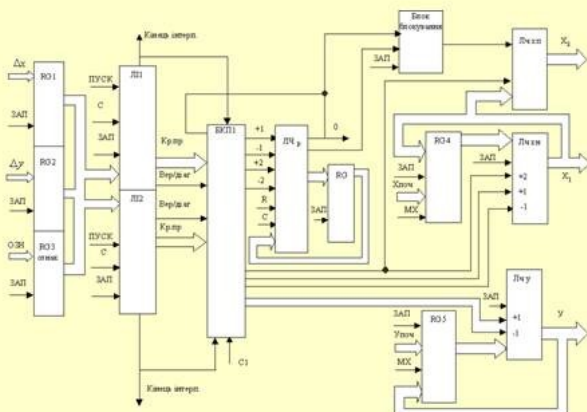


$$I_2 = I_{PB} + \Delta I_{\Gamma}, \quad I_4 = I_2 + 2\Delta I_P, \quad I_6 = I_4 + 2\Delta I_P, \dots, \quad I_{2w} = I_{2w-2} + 2\Delta I_P, \dots;$$

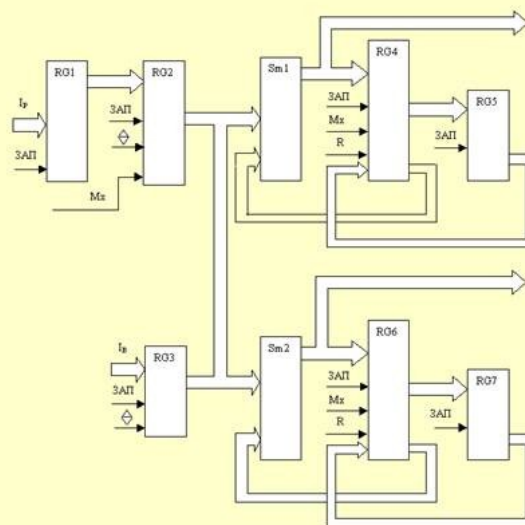
$$I_1 = I_{PB}, \quad I_3 = I_1 + 2\Delta I_P, \quad I_5 = I_3 + 2\Delta I_P, \quad I_7 = I_5 + 2\Delta I_P, \dots, \quad I_{2w+1} = I_{2w-1} + 2\Delta I_P, \dots$$

$$\delta_K = \delta_{\text{ноч.}} + z \cdot 2^{-(V+1)}, \quad \delta_{KH} = \delta_{\text{ноч.}} + \lfloor z/2 \rfloor \cdot 2^{-(V+2)}, \quad \Delta\delta = \delta_K - \delta_{KH} = 3/4 \cdot z \cdot 2^{-(V+1)}.$$

Структурні схеми пристроїв для зафарбовування парних і непарних точок рядка растеризації

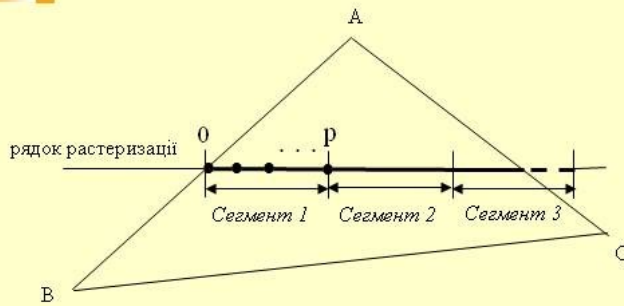


Структурна схема адресного блоку



Структурна схема блоку визначення інтенсивностей кольору внутрішніх точок трикутника

Сегментна реалізація зафарбовування за Гуро



Розрядність сегмента	w
2	2
4	4
8	8
16	14
32	22
64	32

Сегмент 1

I_0	$I_0 + \Delta I_\Gamma$	$I_0 + 2\Delta I_\Gamma$...	$I_0 + (k-1)\Delta I_\Gamma$
-------	-------------------------	--------------------------	-----	------------------------------

$$I_i = I_0 + i \cdot \Delta I_\Gamma.$$

$$k = \log_2 p + 1,$$

$$\frac{(k-1)!}{2((k-1)-2)!} = \frac{(k-1) \cdot (k-2) \cdot (k-3)!}{2(k-3)!} = \frac{(k-1)(k-2)}{2},$$

$$\frac{(k-2)!}{2((k-2)-2)!} = \frac{(k-2) \cdot (k-3) \cdot (k-4)!}{2(k-4)!} = \frac{(k-2) \cdot (k-3)}{2},$$

$$w = k + (k-2) \cdot \frac{(k-1) + (k-3)}{2}.$$

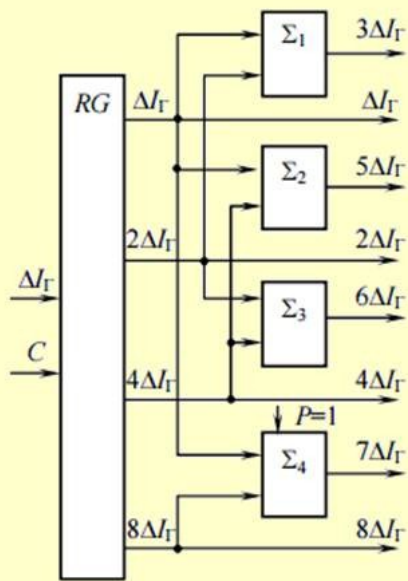
Визначення приростів інтенсивностей кольору

ΔI_Γ	$2\Delta I_\Gamma$	$3\Delta I_\Gamma$...	$(k-1)\Delta I_\Gamma$	$k\Delta I_\Gamma$
-------------------	--------------------	--------------------	-----	------------------------	--------------------

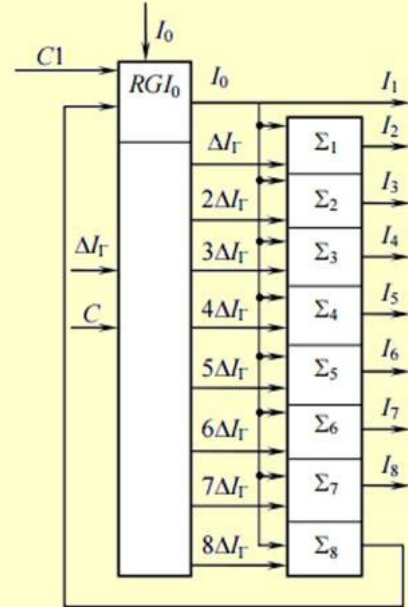
Сегмент 2

$I_0 + k\Delta I_\Gamma$...	$I_0 + 2(k-2)\Delta I_\Gamma$	$I_0 + 2(k-1)\Delta I_\Gamma$
--------------------------	-----	-------------------------------	-------------------------------

Структурна схема пристрою для сегментного зафарбовування Гуро

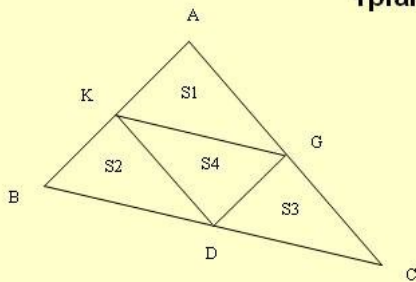


Структурна схема пристрою для визначення приростів інтенсивностей кольору



Структурна схема пристрою для визначення інтенсивностей кольору

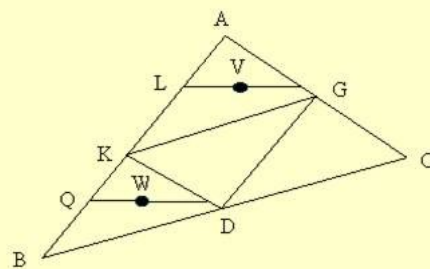
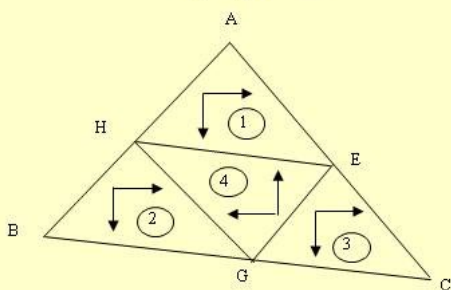
Метод підвищення продуктивності зафарбовування за Гуро з використанням тріангуляції Серпінського



$$AK = KB, \quad AG = GC, \quad BD = DC.$$

$$\Delta AKG = \Delta KBD = \Delta GDC = \Delta DGK.$$

Напрямки зафарбовування складових трикутників



$$I_V = I_L + \Delta I_{\Gamma} LV, \quad \text{де } I_L = I_A + \Delta I_{AK} \text{БП}_{AL};$$

$$I_W = I_Q + \Delta I_{\Gamma} QW, \quad \text{де } I_Q = I_K + \Delta I_{KQ} \text{БП}_{KQ}.$$

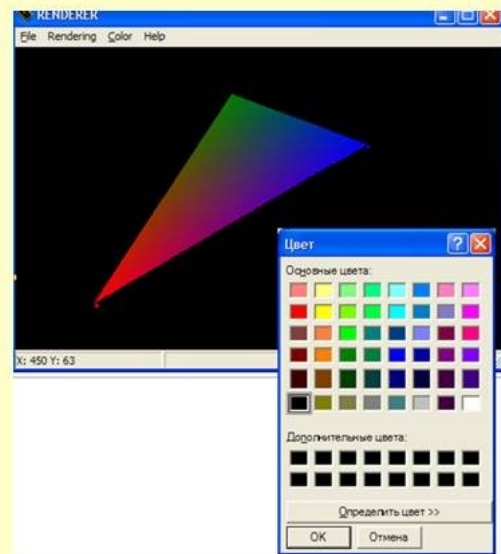
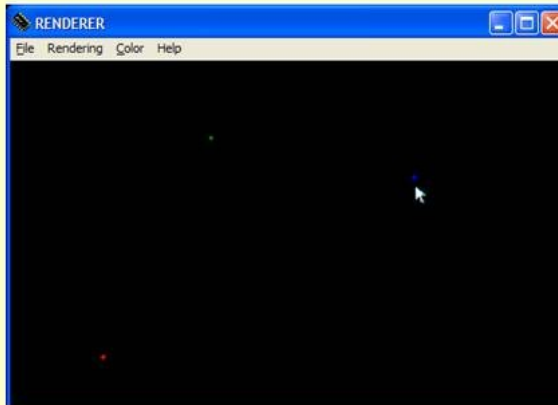
$$I_V = I_A + \Delta I_{AK} \text{БП}_{AL} + \Delta I_{\Gamma} LV,$$

$$I_W = I_K + \Delta I_{KQ} \text{БП}_{KQ} + \Delta I_{\Gamma} QW.$$

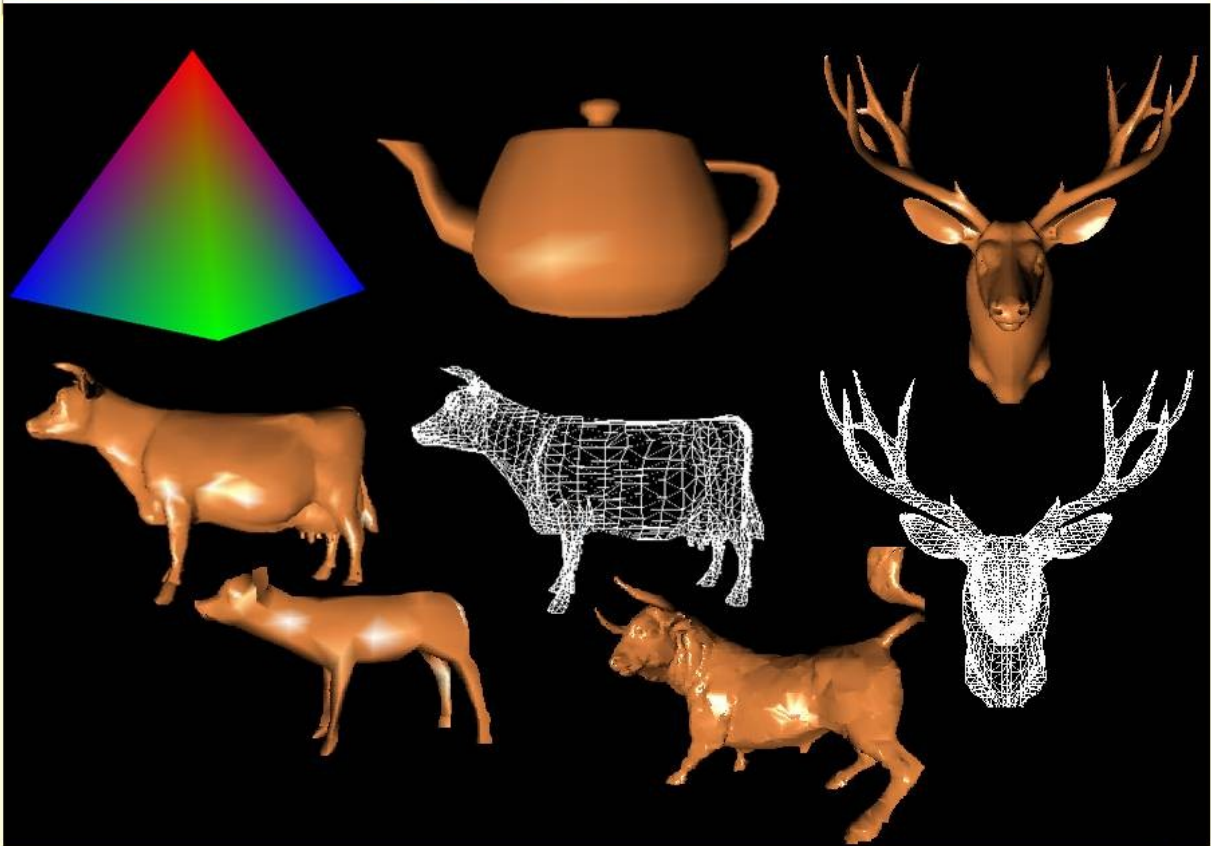
Оскільки $LV = QW$, $\text{БП}_{AL} = \text{БП}_{KQ}$, то

$$I_V - I_W = I_A - I_K.$$

Інтерфейс програми для зафарбовування з використанням триангуляції Серпінського



Приклади зафарбовування фігур



Наукова новизна одержаних результатів

Уперше встановлено залежність приросту інтенсивностей кольору для точок поверхні, обмеженої трикутником, що дає можливість підвищити продуктивність зафарбовування та спростити апаратну реалізацію пристроїв зафарбовування.

Подальшого розвитку отримав метод зафарбовування Гуро, який відрізняється від відомого визначення в кожному такті не однієї, а відразу кількох інтенсивностей точок трикутника, що дозволяє підвищити продуктивність формування графічних сцен.

Практична цінність отриманих результатів. Практична цінність одержаних результатів полягає в тому, що на основі отриманих в магістерській кваліфікаційній роботі теоретичних положень запропоновано алгоритми та розроблено програмні засоби зафарбовування у комп'ютерних системах візуалізації тривимірних зображень. Розроблено структурні схеми рендерів.

Наукова новизна одержаних результатів

Подальшого розвитку отримав метод зафарбовування з використанням сферично-кутової інтерполяції, за яким нормалізація векторів виконується без трудомісткої процедури визначення кутів між векторами, що дозволяє зменшити обчислювальні витрати в середньому на 15 % та спростити апаратну реалізацію.

Запропоновано метод усунення артефактів при растеризації ребер суміжних полігонів на етапі формування зображення, який полягає в локалізації точок, не вибраних при растеризації, та довизначення їх параметрів, що виключає необхідність постоброблювання зображень.

Практичне значення одержаних результатів полягає в тому, що на основі отриманих теоретичних положень розроблено комплекс програмних і апаратних засобів, зокрема:

- спеціалізований програмний модуль для тестування методів рендерингу, який дозволяє отримати порівняльні оцінки за точністю й продуктивністю методів;
- високопродуктивні програмні та апаратні засоби для зафарбовування, текстурування, контурного формоутворення та антиаліазингу;
- структурні схеми пристроїв для формування тривимірних зображень, що є основою для виготовлення конкурентоспроможних зразків цієї продукції;
- швидкодіючі графічні акселератори для систем комп'ютерної графіки.

За темою магістерської кваліфікаційної роботи опубліковано 5 наукових праць, з них 3 авторських права на комп'ютерні програми.

Перемоги у міжнародних конкурсах з комп'ютерної графіки
“IT-Архангельск”, “Творчество без границ”

Авторські свідоцтва

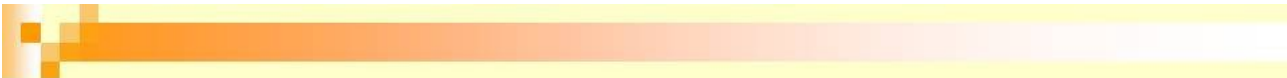


Перемоги у конкурсах



Авторські свідоцтва





Дякую за увагу !

