

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет інформаційних технологій та комп'ютерної інженерії
(повне найменування інституту, назва факультету (відділення))

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

Пояснювальна записка

до магістерської кваліфікаційної роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему **«РОЗРОБКА МЕТОДІВ І ЗАСОБІВ АНТИАЛІЗИНГУ
ЗОБРАЖЕНЬ КРИВИХ ДРУГОГО ПОРЯДКУ»**

Виконав: студент II курсу, групи III –18м
напряму підготовки
121 – Інженерія програмного забезпечення
(шифр і назва напряму підготовки, спеціальності)

Бур'ян А.С.
(прізвище та ініціали)

Керівник д.т.н., проф.. Романюк О.Н.
(прізвище та ініціали)

Рецензент к.т.н., доц. Колесницький О.К.
(прізвище та ініціали)

Вінниця – 2020 року

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.
“29” жовтня 2019 року

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

Бур'яну Андрію Сергійовичу

1. Тема роботи – Розробка методів і засобів антиаліайзингу зображень кривих другого порядку

Керівник роботи: Романюк Олександр Никифорович, д.т.н., завідувач кафедри ПЗ, затверджені наказом ВНТУ від “06” березня 2020 року № 76.

2. Строк подання студентом роботи - 5 червня 2020 р.

3. Вихідні дані до роботи: тип антиаліайзингу – контурний; графічний режим – TrueColor; метод формування крокової траєкторії – метод оцінювальної функції; типи кривих – коло, еліпс, гіпербола.

4. Зміст розрахунково-пояснювальної записки: аналіз методів і засобів антиаліайзингу графічних об'єктів; розробка високопродуктивних методів антиаліайзингу для траєкторій кіл; розробка методів антиаліайзингу меж еліпсів та гіпербол; програмна реалізація антиаліайзингу кривих другого порядку; економічна частина.

5. Перелік графічного матеріалу: контурний антиаліайзинг; метод додаткової вибірки; граф-схеми розроблених алгоритмів; інтерфейс користувача.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Романюк О. Н., д.т.н., професор 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 методів і засобів антиаліайзингу графічних об'єктів	10.01.2020	Вик.
2	Розробка високопродуктивних методів антиаліайзингу для траєкторій кіл	5.02.2020	Вик.
3	Розробка методів антиаліайзингу меж еліпсів та гіпербол	5.03.2020	Вик.
4	Програмна реалізація антиаліайзингу кривих другого порядку	15.04.2020	Вик.
5	Економічна частина	30.05.2020	Вик.

Студент _____ Бур'ян А.С.

Керівник магістерської кваліфікаційної роботи _____ Романюк О.Н.

Рецензент магістерської кваліфікаційної роботи _____ Колесницький О.К.

АННОТАЦІЯ

Проведено аналіз існуючих методів антиаліайзингу. Обґрунтовано вибір методі оцінювальної функції як базового для розробки методів і засобів контурного антиаліайзингу

Вперше запропоновано метод антиаліайзингу зображень траєкторій кіл, особливість якого полягає у розбитті піксела на субпікселі з подальшим знаходженням для кожного з них знаків оцінювальних функцій, по яких розраховують площу покриття піксела траєкторією, що дозволяє підвищити продуктивність згладження;

Отримано нові аналітичні залежності для розрахунку площі покриття піксела траєкторією, що дозволило модифікувати табличний метод антиаліайзингу траєкторій кіл, що дозволило зменшити об'єми постійної пам'яті,;

Подальшого розвитку отримали методи антиаліайзингу кривих другого порядку, в яких, на відміну від відомих, використано нові оцінювальні функції, що дозволило підвищити продуктивність згладження.

Розроблені алгоритми та програмні засоби для контурного антиаліайзингу зображень кривих другого порядку.

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

ABSTRACT

The analysis of existing methods of antialiasing is carried out. The choice of the method of evaluation function as a basis for the development of methods and means of contour antialiasing is substantiated

For the first time, a method of antialiasing images of circle trajectories was proposed, the feature of which is to divide a pixel into subpixels with subsequent finding for each of them signs of estimation functions, which calculate the pixel coverage area by a trajectory, which increases smoothing productivity;

New analytical dependences for calculation of the area of coverage of a pixel of a trajectory by a trajectory are received , which allowed to modify the tabular method of anti-aliasing of the trajectories of circles, which allowed to reduce the amount of permanent memory.

The methods of antialiasing of the second order curves were further developed, in which, in contrast to the known ones, new estimation functions were used, which allowed to increase the smoothing productivity.

Algorithms and software for contour anti-aliasing of second-order curve images 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.....	
1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ АНТИАЛІАЙЗИНГУ ГРАФІЧНИХ ОБ'ЄКТІВ.....	
1.1 Методи надлишкової вибірки.....	
1.2 Методи антиаліайзингу для границь кіл та еліпсів.....	
1.3 Методи антиаліайзингу для тривимірних зображень.....	
Висновки.....	
2 РОЗРОБКА ВИСОКОПРОДУКТИВНИХ МЕТОДІВ АНТИАЛІАЙЗИНГУ ДЛЯ ТРАЄКТОРІЇ КІЛ.....	
2.1 Метод додаткових оцінювальних функцій для антиаліайзингу меж кіл.....	
2.2 Метод модифікованої оцінювальної функції для антиаліайзингу меж кіл.....	
2.3 Модифікація табличного методу антиаліайзингу меж кіл.....	
Висновки.....	
3 РОЗРОБКА МЕТОДІВ АНТИАЛІАЙЗИНГУ МЕЖ ЕЛІПСІВ ТА ГІПЕРБОЛ.....	
3.1 Антиаліайзинг меж еліпсів.....	
3.2 Антиаліайзинг меж гіперболи.....	
Висновки.....	
4. ПРОГРАМНА РЕАЛІЗАЦІЯ АНТИАЛІАЙЗИНГУ КРИВИХ ПОРЯДКУ	
4.1 Засоби для антиаліайзингу зображень траєкторій кіл	
4.2 Засоби для Антиаліайзингу зображень траєкторій еліпсів і гіпербол.....	
Висновки.....	
5.Економічний розділ.....	
Література.....	
ДОДАТКИ	
Додаток А. Технічне завдання.....	
Додаток Б. Ілюстративний матеріал	
Додаток С. Лістинги	

ВСТУП

Актуальність теми. Більшість сучасних пристроїв відображення інформації використовує растровий принцип формування зображення. Особливість даного принципу полягає в тому, що зображення представляється у вигляді матриці точок, кожна з яких має фіксоване положення на екрані та інтенсивність кольору. При формуванні зображення на стадії растеризації графічні примітиви переводяться в растрову форму. У результаті перетворень виникають спотворення, обумовлені недостатньою роздільною здатністю пристроїв відображення. На зображеннях з'являються артефакти, одним із проявів яких є яскраво виражені сходинки або зубці на краях об'єктів. Даний ефект отримав назву ступінчастого ефекту чи ефекту *аліайзингу*. Ефект аліайзингу суттєво погіршує якість сформованого зображення. Тому актуальними є методи згладження крокових траєкторій.

Проведені дослідження [] показали, що при використанні 17" монітора і розміщенні спостерігача на відстані 65 см від екрана для повного усунення ефекту аліайзингу потрібен монітор із роздільною здатністю як мінімум 4000x4000 пікселів, а для людей із рівнем зору вище середнього - взагалі 8000x8000 пікселів. Сучасний рівень технологій поки що не в змозі забезпечити таку роздільну здатність, тому для забезпечення реалістичності синтезованих зображень у системах комп'ютерної графіки використовують спеціальні методи та засоби усунення ступінчастого ефекту.

Криві другого порядку широко використовуються в системах комп'ютерної графіки як базові примітиви. Тому доцільно розробити для них ефективні методи та алгоритми.

Оскільки антиаліайзинг у системах комп'ютерної графіки вимагає суттєвих обчислювальних затрат, то актуальною задачею є розробка високопродуктивних методів, алгоритмів і програм для його усунення.

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

Мета і завдання дослідження. Метою роботи є підвищення продуктивності антиалайзингу кривих другого порядку за рахунок розробки та використання нових методів і засобів

Основними **задачами дослідження** є:

- аналіз методів і засобів антиалайзингу контурів;
- спрощення аналітичних залежностей для задач антиалайзингу;
- розробка методів підвищення продуктивності згладжування траєкторій кіл, еліпсів і гіпербол;
- розробка алгоритмів і програм усунення алаізингу контурів кривих другого порядку;
- комп'ютерне моделювання для перевірки отриманих результатів.

Об'єкт дослідження – процес антиалайзингу зображень кривих другого порядку

Предмет дослідження – методи і засоби антиалайзингу зображень кривих другого порядку.

Методи дослідження У процесі дослідження застосовувалася: теорія чисел та чисельних методів, математичний апарат апроксимації функцій; методи аналітичної геометрії для розробки методів згладження зображень кривих другого порядку; теорія алгоритмів і програмного забезпечення для розробки програм; комп'ютерне моделювання для перевірки отриманих результатів.

Наукова новизна одержаних результатів:

- вперше запропоновано метод антиалайзингу зображень траєкторій кіл, особливість якого полягає у розбитті піксела на субпікселі з подальшим знаходженням для кожного з них знаків оцінювальних функцій, по яких розраховують площу покриття піксела траєкторією, що дозволяє підвищити продуктивність згладження;

- отримано нові аналітичні залежності для розрахунку площі покриття пікселя траєкторією, що дозволило модифікувати табличний метод антиаліайзингу траєкторій кіл, що дозволило зменшити об'єми постійної пам'яті;

- подальшого розвитку отримали методи антиаліайзингу кривих другого порядку, в яких, на відміну від відомих, використано нові оцінювальні функції, що дозволило підвищити продуктивність згладження.

Достовірність отриманих результатів підтверджується збігом результатів теоретичних досліджень із результатами комп'ютерного моделювання.

Практичне значення одержаних результатів полягає у тому, що на основі проведених теоретичних досліджень і отриманих наукових результатів розроблено алгоритми та програми для антиаліайзингу кривих другого порядку.

Особистий внесок здобувача у роботах, виконаних у співавторстві, є таким: [18] – аналіз методів; [19] – формули для розрахунку площі покриття.

Апробація результатів дисертації. Основні положення дисертаційної роботи доповідалися та обговорювалися на

- XX Всеукраїнській науково-технічній конференції молодих вчених, аспірантів та студентів “Стан, досягнення і перспективи інформаційних систем і технологій”, Одеса, 2020.

- IX Міжнародній науково-практичній конференції молодих вчених та студентів «Молодь у світі сучасних технологій», МССТ-2020, Херсон, 2020.

Публікації. За результатами виконаних досліджень опубліковано 2 наукові праці у збірниках матеріалів міжнародних конференцій.

1. АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ АНТИАЛІАЙЗИНГУ ГРАФІЧНИХ ОБ'ЄКТІВ

Зменшення ефекту аліайзингу може бути досягнуто за рахунок збільшення роздільної здатності пристрою відображення [1, 2]. Основний недолік такого підходу полягає в тому, що він потребує використання великих масивів пам'яті та призводить до зниження швидкодії. Крім того, існують фізичні обмеження [1] щодо збільшення роздільної здатності пристроїв відображення. Проведені дослідження [2] показали, що при використанні 17" монітора і розміщенні спостерігача на відстані 65 см від екрана для повного усунення ефекту аліайзингу потрібен монітор із роздільною здатністю як мінімум 4000x4000 пікселів, а для людей із рівнем зору вище середнього - взагалі 8000x8000 пікселів. Сучасний рівень технологій поки що не в змозі забезпечити таку роздільну здатність.

Для забезпечення належної якості зображень використовуються спеціальні методи антиаліайзингу . На відміну від звичайних методів растеризації, які розраховують інтенсивність кольору лише в точці, що розташована в центрі піксела [4], методи антиаліайзингу обчислюють колір з урахуванням зони, яка оточує потокову точку . При цьому в графічному конвеєрі на стадіях растеризації, видалення невидимих поверхонь та пост-обробки кадру виконуються додаткові обчислення [5].

Усі методи антиаліайзингу можна умовно поділити на методи надлишкової вибірки та аналітичні [5, 6].

1.1. Методи надлишкової вибірки

Одним із найбільш поширених методів надлишкової вибірки є метод, який оснований на збільшенні дискретизації. Суть методу полягає в тому, що зображення сцени обчислюється з роздільною здатністю більшою ніж у пристрою відображення, а перед відображенням на екрані зменшують

шляхом усереднення [7]. Даний метод має просту апаратну реалізацію та може бути легко інтегрований в існуючі системи комп'ютерної графіки. Недоліком підходу є його низька продуктивність. При збільшенні кількості пікселів в n разів, обчислювальна складність збільшується в n^2 разів [6]. Метод передбачає використання додаткової пам'яті та підвищення пропускної здатності шини пам'яті.

У більшості існуючих методів растеризації значення інтенсивностей кольору обчислюють лише в центрах пікселів. У методі додаткових вибірок [11] розраховують інтенсивність кольору в декількох допоміжних точках всередині піксела. Значення цих підвибірок потім усереднюють для отримання результуючої інтенсивності кольору всього пікселі. Якість згладжування залежить від кількості додаткових вибірок та від їхнього розташування всередині піксела. Збільшення кількості точок вибірки покращує якість зображення, але одночасно підвищує потребу в додатковій пам'яті та призводить до зниження швидкодії [8].

Існують різні варіанти розташування додаткових вибірок усередині піксела.

Найпростішим є випадок, коли додаткова дискретизація виконується у формі впорядкованої решітки [7].

Таке розташування не забезпечує максимальної якості згладжування. Можна використати зсунуту решітку. Це покращить згладженість траєкторії.

М. Діпе та Е. Уолт [10] запропонували стохастично розташувати додаткові точки вибірки. Даний підхід використовує ту особливість, що в цьому випадку людське око розрізняє "шум". В цьому випадку, аліайзинг замінюється іншим артефактом, до якого людське око менш чутливе.

Основним недоліком такого підходу є великі обчислювальні витрати, обумовлені тим, що потрібно не менше 16 випадково розподілених підвибірок на піксел для досягнення потрібного рівня "неперервного шуму".

П. Хаеберлі та К. Акелей [10] запропонували для зменшення кількості додаткової пам'яті застосувати нагромаджувальний буфер. При використанні n додаткових точок для обчислення інтенсивності кольору пікселя рендеринг сцени здійснюється за n проходів. Після кожного проходу відбувається зсув сітки точок додаткової вибірки. Результати розрахунку інтенсивності кольору пікселя підсумовуються в нагромаджувальному буфері та усереднюються перед видачею зображення на екран. Основний недолік підходу – низька швидкодія, обумовлена необхідністю розраховувати зображення сцени декілька разів.

У роботах [11, 14-16] для зменшення кількості додаткових обчислень пропонується при визначенні інтенсивності кольору пікселя використати додаткові вибірки, що були розраховані для сусідніх пікселів. Даний підхід використовується в графічних акселераторах фірми Nvidia. Існує два варіанти розташування додаткових точок усередині пікселя. Перший варіант [11] (рис. 1.1а) отримав назву Quincunx. Його апаратно реалізовано у графічних акселераторах GeForce-3 Titanium. На рис. 1.1 зображено інший варіант, який забезпечує кращу якість згладжування [11]. Таке розташування додаткових точок вибірки використовується в усіх акселераторах фірми Nvidia, починаючи із серії GeForce-4 Titanium. Основним недоліком розглянутого підходу є розмитість границь об'єктів на кінцевому зображенні.

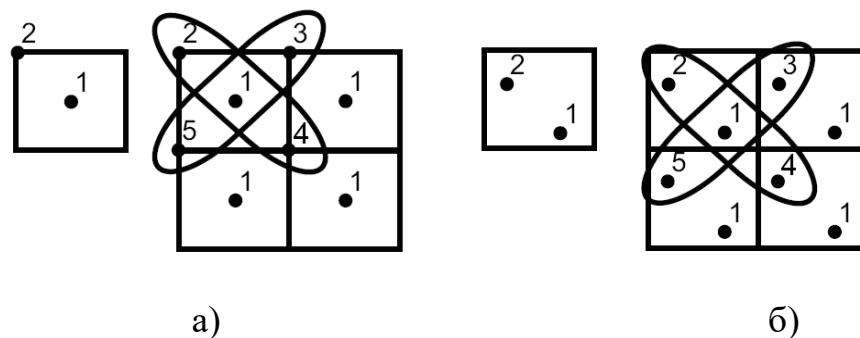


Рис. 1.1. Вибірki пікселів
в графічних акселераторах фірми Nvidia:
а) схема Quincunx; б) схема Accuview.

1.2 Методи антиаліайзингу для границь кіл та еліпсів

У роботі [7, 11] запропонована модифікація методу Гупти-Спроула для усунення аліайзингу границі кола. При цьому використовується конусна модель піксела. Суть методу полягає в тому, що знаходиться відстань від кола до центра піксела, яку використовують як індекс у таблиці заздалегідь розрахованих значень фільтра. Встановлено, що для кіл із радіусом більшим 10, дугу кола всередині піксела можна апроксимувати прямою (похибка складатиме не більше 0,4% [11]). При цьому можна використовувати ті ж таблиці, що і для згладжування прямих. Для всіх кіл меншого радіусу потрібно будувати окремі таблиці.

Основним недоліком даного підходу є необхідність виконання довгих операцій у циклі інтерполювання та застосування чисел із плаваючою комою.

Більшість існуючих методів антиаліайзингу границь кіл та еліпсів [11] використовують квадратну модель піксела, тому що вона має порівняно малу обчислювальну складність.

М. Капеля [11] запропонував алгоритм антиаліайзингу границь еліпсів, який використовує знаходження квадратних коренів за методом Ньютона [16]. Недолік методу полягає у наявності довготривалої операції ділення.

Згідно алгоритмів антиаліайзингу [11] використовуються прості з обчислювальної точки зору вирази. Алгоритм має просту апаратну реалізацію, але не забезпечує високу якість антиаліайзингу, оскільки використовує тільки чотири рівні інтенсивності кольору.

Д. Філд [14] запропонував метод прогнозувань та поправок для згладження траєкторій кіл. Інтенсивність кольору піксела знаходять за виразом:

$$I_p = I_M \cdot \alpha,$$

де α - площа покриття піксела колом.

При визначенні площі перерізу піксела та кола враховують, що в другому октанті границя кола перетинає один або два суміжні пікселі по

вертикалі. При визначенні площі α дуга кола апроксимується прямою (рис. 1.2.)

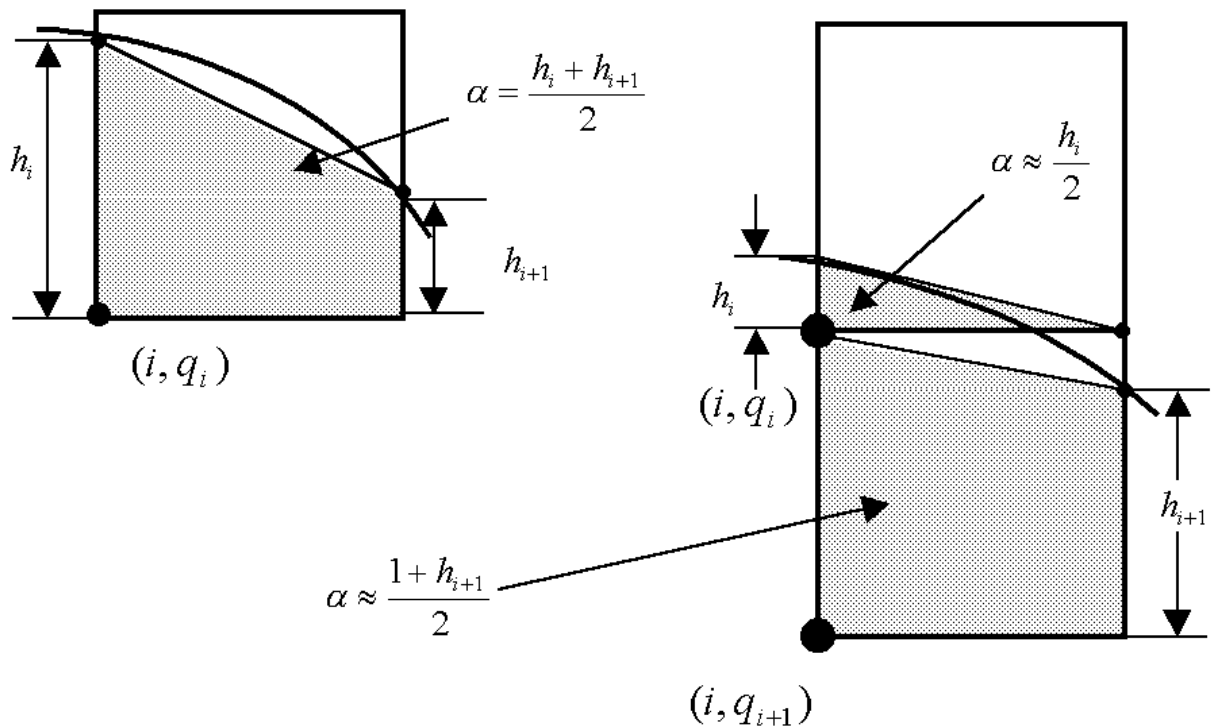


Рис. 1.2. Обчислення площі покриття за методом Філда

Для випадку, коли границя кола перетинає лише один піксел, розрахунок площі покриття проводиться за формулою:

$$\alpha \approx \frac{h_i + h_{i+1}}{2},$$

де $h_i = \sqrt{R^2 - i^2} - q_i$; R - радіус кола; (i, q_i) - координати пікселя.

Для спрощення обчислень використовують такі вирази:

$$\alpha_{\text{верх}} \approx \frac{h_i}{2}, \quad \alpha_{\text{низ}} \approx \frac{1 + h_{i+1}}{2},$$

де $h_i = \sqrt{R^2 - i^2} - q_i$; R - радіус кола; (i, q_i) - координати пікселя.

Для знаходження $h_i = \sqrt{R^2 - i^2} - q_i$ застосовується метод прогнозувань і поправок [16].

Основний цикл інтерполювання містить лише операції типу додавання.

Алгоритм Філда не містить „довгих” операцій у циклі інтерполювання, використовує лише цілочисельну арифметику та забезпечує достатньо високу якість згладжування.

Суттєвий недолік алгоритму полягає в усуненні ефекту аліазингу лише для кола, який формують на однотонному фоні, що складає тільки окремий випадок.

Згідно методу Ксяолін Ву [11, 14, 15] інтенсивність кольору дуги кола розподіляється між двома суміжними пікселями, що знаходяться безпосередньо над та під ідеальною дугою, пропорційно до відстані від центра пікселя до точки, у якій дуга перетинає координатну решітку (рис. 1.3).

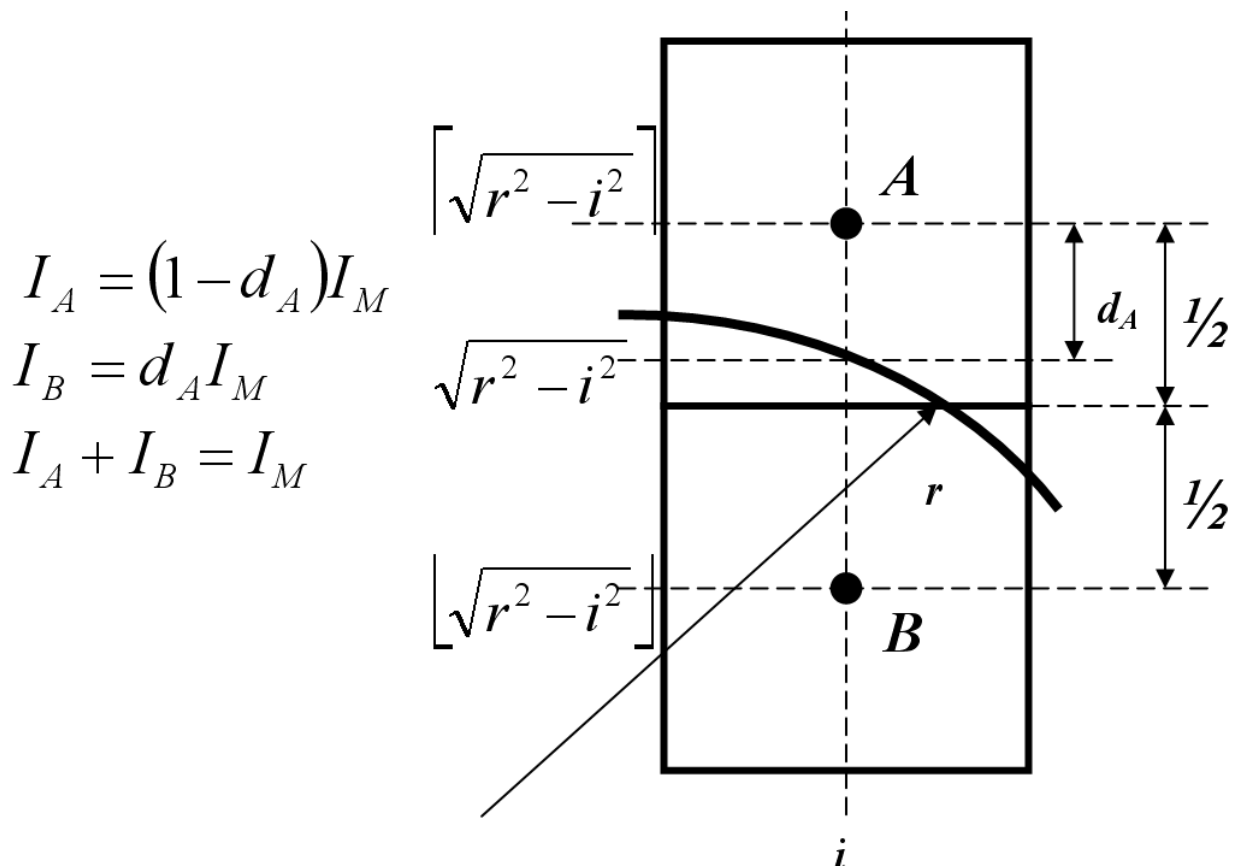


Рис. 1.3. Визначення інтенсивності кольору пікселей за методом Ву

Сума інтенсивностей цих двох пікселів завжди дорівнює максимальній. Інтенсивності кольору пікселів визначаються за формулами:

$$I(\lfloor \sqrt{r^2 - j^2} \rfloor, j) = D(r, j); \quad I(\lfloor \sqrt{r^2 - j^2} \rfloor, j) = \overline{D(r, j)},$$

$$D(r, j) = \lfloor (2^m - 1) \left(\sqrt{r^2 - j^2} - \sqrt{r^2 - j^2} \right) + 0.5 \rfloor$$

де r - радіус кола; $1 \leq j \leq \frac{r}{\sqrt{2}}$; 2^m - кількість рівнів інтенсивності.

Обчислювальна складність методу Ву менша порівняно з методом Філда, однак він потребує використання блоків постійної пам'яті для зберігання таблиці функції $D(r, j)$, що ускладнює апаратну реалізацію пристрою на основі БМК і ПЛІС.

Алгоритми антиаліазингу, наведені в роботах [14], передбачають, що інтенсивність кольору фонового зображення є сталою. Для більш загального випадку, коли інтенсивність кольору фонового зображення, на яке накладається коло, не є сталою та може змінюватись довільно, у циклі інтерполювання необхідне виконання операції ділення та множення, це призводить до ускладнення обчислювального процесу та зменшення швидкодії.

Таким чином, існує необхідність розробки нових підходів до антиаліазингу границь кіл та еліпсів, які б враховували можливість зміни інтенсивностей кольору еліпса і фонового зображення, та характеризувалися простотою обчислювального процесу.

1.3 Методи антиаліазингу для тривимірних зображень

Генерація тривимірних графічних зображень є складною обчислювальною задачею, тому на практиці поверхні об'єктів апроксимують полігональною мережею [12, 13]. Найбільш поширене представлення графічних об'єктів мережею трикутників, оскільки [12, 13]: трикутник є найпростішим полігоном, вершини якого однозначно задають грань; будь-

яку поверхню можна гарантовано розбити на трикутники; обчислювальна складність алгоритмів розбиття областей на трикутники істотно менша ніж при використанні інших полігонів; реалізація процедур рендерингу найпростіша для області, обмеженої трикутником; для трикутника легко визначити три його найближчих сусідів, що мають з ним спільні грані.

У більшість існуючих методів антиаліазингу тривимірних зображень трикутник є базовим примітивом [14]. Найбільше розповсюдження отримали такі методи зафарбовування полігональних областей, обмежених трикутником: однотонне зафарбовування [14]; зафарбовування за Гуро [12-13]; зафарбовування за Фонгом [12-14].

При використанні зафарбовування за Гуро [12, 13] область, обмежена багатокутником, зафарбовується шляхом лінійної інтерполяції інтенсивностей кольору вздовж ребер, а потім і між ребрами вздовж кожного рядка растеризації. Лінійність зміни інтенсивності кольору вздовж ребра може бути використана для підвищення продуктивності антиаліазингу.

При згладжуванні тривимірних зображень, окрім площі частини піксела, яка покривається полігоном, на інтенсивність кольору піксела впливає розташування полігонів по відношенню до спостерігача. Тому для антиаліазингу тривимірних трикутників необхідно видалення невидимих поверхонь на рівні піксела.

Е. Катмул [11] запропонував точний метод усунення ступінчастого ефекту та видалення невидимих поверхонь, оснований на методі скануючого рядка. Для усунення аліазингу використовується квадратна модель піксела. Всі полігони, які покривають піксел, відсікаються до меж піксела. Визначається видимість кожного зі сформованих фрагментів за допомогою сортування по глибині. Значення інтенсивності кольору кожного видимого фрагмента множиться на його площу та додається до результуючої інтенсивності кольору піксела. Даний метод потребує сортування полігонів перед кожним рендерингом сцени та використовує відносно складу

процедуру відсікання, що обумовлює його велику обчислювальну складність [6].

У методі А-буфера [11] замість точного визначення площі покриття використовуються бітові маски. Для визначення площі фрагментів полігонів піксел задається бітовою маскою розмірами 4 на 8. Площі субпікселних фрагментів обчислюються за допомогою булевої операції нерівнозначності. Формування фінальної маски фрагмента полігона з бітових масок його ребер зображено на рис. 1.4.

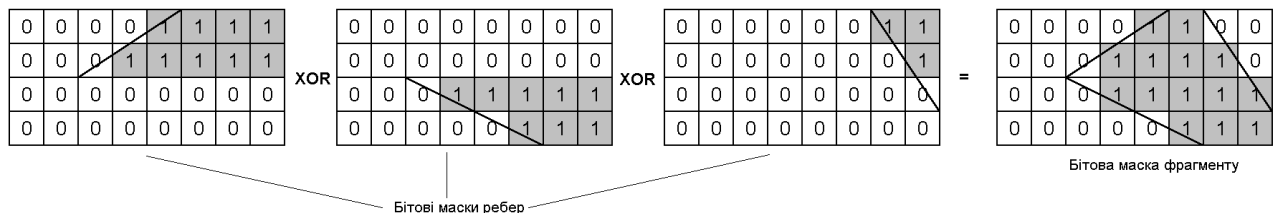


Рис.1.4. Утворення маски фрагмента полігона з бітових масок його ребер

За методом А-буфера в пам'яті зберігаються пов'язані списки фрагментів полігонів всередині кожного піксела. Фрагменти сортуються за значенням z-координати для підтримки правильного відображення прозорих поверхонь. При перекритті по z-координаті різних фрагментів та належності їх до одного об'єкту, вони об'єднуються в один елемент.

Для кожного фрагмента окрім мінімального значення z-координати зберігається також максимальне значення глибини. Для фрагментів, що перетинаються (рис 1.12), розрахунок видимої частини виконується за формулою [11]:

$$Vis_{front} = \frac{Z_{max_next} - Z_{min_front}}{(Z_{max} - Z_{min})_{front} + (Z_{max} - Z_{min})_{next}},$$

де Vis_{front} – коефіцієнт, що визначає, яка частина найближчого до спостерігача фрагмента не закрита від спостерігача; $Z_{max_front}, Z_{min_front}$ - відповідно максимальне та мінімальне значення z-координати ближнього фрагмента; $Z_{max_next}, Z_{min_next}$ - відповідно максимальне та мінімальне значення z-координати наступного фрагмента.

Кінцева інтенсивність кольору області перетину знаходиться за виразами [11]:

$$\alpha = Vis_{front} \cdot Opacity_{front} + (1 - Vis_{front}) \cdot Opacity_{next},$$

$$I = \alpha \cdot I_{front} + (1 - \alpha) \cdot I_{next},$$

де $Opacity_{front}$ - коефіцієнт прозорості фронтального фрагмента; $Opacity_{next}$ - коефіцієнт прозорості наступного фрагмента; I_{front} - інтенсивність кольору фронтального фрагмента; I_{next} - інтенсивність кольору наступного фрагмента.

Формула (1.5) справедлива для випадку, коли фрагменти, які перекриваються по z-координаті, нахилені протилежно. Це складає лише окремий випадок, тому метод А-буфера може генерувати зображення з артефактами у місцях перетину поверхонь [11].

Для отримання результуючого кольору пікселя розглядаються всі фрагменти, починаючи з найближчого до спостерігача. Використовуючи субпіксельні бітові маски визначається внесок кожного фрагмента у кінцевій інтенсивності кольору пікселя.

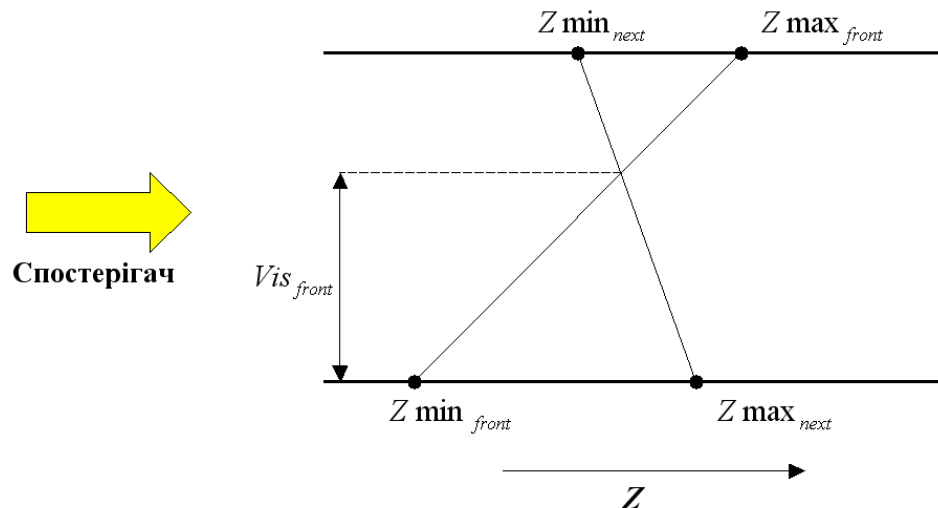


Рис. 1.5. Визначення видимої частини для фрагментів, що перетинаються

У методі А-буфера важко спрогнозувати необхідні об'єми пам'яті, адже не можливо заздалегідь визначити, до яких розмірів може зрости список фрагментів полігонів [14]. Крім того, обчислювальні витрати на формування

одного пікселя зображення залежать від складності сцени і не можуть бути передбачені. Вказані недоліки ускладнюють апаратну реалізацію методу А-буфера.

А. Шилінг [11] запропонував для підвищення якості згладжування окрім бітової маски зберігати точне значення площі пікселя, що покривається полігоном. Однак кількість випадків, для яких можна застосувати точне значення площі, є обмеженою.

У роботі [11] запропоновано використовувати лише фіксований об'єм пам'яті для зберігання фрагментів полігонів, що покривають піксел. Фрагменти, для збереження яких не вистачає місця в пам'яті, відкидаються [99]. Даний підхід дозволяє спростити апаратну реалізацію методу А-буфера, однак призводить до появи артефактів при візуалізації сцен зі складною геометрією. Інший підхід полягає у тому, що для зберігання надлишкових фрагментів додатково використовується системна пам'ять. Основним недоліком даного підходу є збільшене навантаження на системну шину, обумовлене необхідністю передачі надлишкових фрагментів.

С. Уінер [11, 14] запропонував модифікацію методу А-буфера, яка полягає у тому, що для випадку, коли в пам'яті не вистачає місця для зберігання фрагментів полігонів, виконуються додаткові проходи для комбінування надлишкових фрагментів. Основним недоліком такого підходу є зниження швидкодії формування зображення, обумовлене необхідністю виконувати додаткові проходи для розрахунку геометрії сцени.

Аналіз показав, що основні роботи щодо вдосконалення методу А-буфера направлені на розробку нових підходів до зменшення об'ємів пам'яті, необхідних для зберігання фрагментів полігонів.

1.4. Визначення напрямку та постановка задач дослідження

Проведений аналіз показав, що на даному етапі розвитку комп'ютерної графіки для усунення ефекту аліайзингу двовимірних зображень найбільше поширення отримали методи крайового антиаліайзингу. Оскільки обчислювальна складність та якість сформованого зображення за цими методами визначається моделлю піксела, то один із напрямків модифікації даних методів полягає в розробці нових моделей пікселів, які б забезпечували прийнятну якість та характеризувались простотою обчислювального процесу. Другий напрямок модифікацій направлений на підвищення швидкодії за рахунок вдосконалення обчислювального .

Головними напрямками та задачами дослідження є:

- аналіз існуючих методів і засобів згладжування контурів об'єктів з метою визначення напрямків до підвищення їх швидкодії;
- отримання більш простих аналітичних залежностей по визначенню геометричних властивостей векторів та кривих другого порядку, що використовуються в процедурах згладжування;
- розробка методів підвищення продуктивності згладжування контурів тривимірних об'єктів;
- розробка методів підвищення продуктивності антиаліайзингу та кривих другого порядку;
- розробка алгоритмів та структур для згладжування контурів графічних об'єктів.

2. РОЗРОБКА ВИСОКОПРОДУКТИВНИХ МЕТОДІВ АНТИАЛІАЙЗИНГУ ДЛЯ ТРАЄКТОРІЇ КІЛ

2.1 Метод додаткових оцінювальних функцій для антиаліаїзингу меж кіл

Пропонується новий метод антиаліаїзингу з використанням додаткових ОФ. Виконаємо субпікселізацію дискретної решітки. (рис. 2.1).

В кожній отриманій точці визначимо знак оцінювальної функції. Відомо, що якщо ОФ більше нуля, то точка знаходиться нижче примітива. В протилежному випадку ОФ додатна. Позначимо знак ОФ в точці, отриманої при субпікселізації $P_{i,j}$. Якщо ОФ додатня, то $P_{i,j} = 1$. В протилежному випадку $P_{i,j} = 0$ Площу покриття піксела можна знайти через суму $P_{i,j}$ за формулою.

$$S_{\text{покр.}} \approx S_a = \frac{\sum_{i=1}^H \sum_{j=1}^V P_{i,j}}{H \cdot V}.$$

Знайдемо абсолютну похибку:

$$\Delta S = |S_{\text{покр.}} - S_a|.$$

Виразимо похибок через H і V .

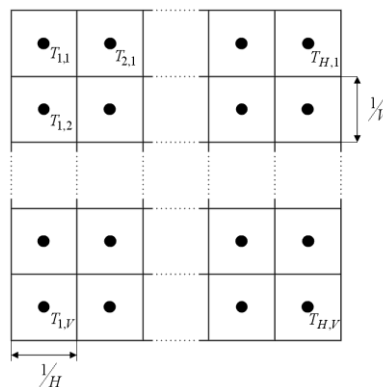


Рис. 2.1. Розташування допоміжних точок всередині піксела

Розробимо алгоритм антиаліаїзингу границі кола з використанням чотирьох точок субпікселізації.

Цк дасть можливість врахувати 5 дискретних: 0; 1/4; 1/2; 3/4; 1. Ці значення відповідають таким умовам: “0” – увесь піксел знаходиться за межами границі кола; “1/4” – четверта частина піксела знаходиться всередині кола; “1/2” – половина піксела знаходиться всередині кола; “3/4” – три четвертих піксела знаходиться всередині кола; “1” – увесь піксел знаходиться всередині кола.

Використаємо 4 допоміжні точки: А, В, С, D (рис. 2.2). Залежно від числа допоміжних точок, які знаходяться всередині кола, площа перетину апроксимується відповідним значенням.

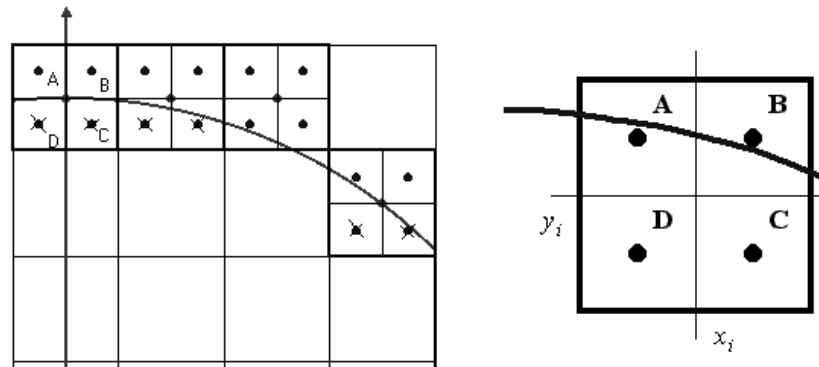


Рис. 2.2. Розташування точок

Скористаємось методом оцінювальної функції. Відомо, що при колівій інтерполяції оцінювальна функція від’ємна для всіх точок, що лежать всередині кола, і обчислюється за формулою[13]:

$$O\Phi_i = x_i^2 + y_i^2 - R^2.$$

Знайдемо значення оцінювальної функції в допоміжних точках. Допоміжна точка А має координати: $x_i^A = x_i - 1/4$; $y_i^A = y_i + 1/4$. Знайдемо значення ОФ

$$\begin{aligned} O\Phi_i^A &= (x_i - 1/4)^2 + (y_i + 1/4)^2 - R^2 = (x_i^2 + y_i^2) - R^2 - \frac{x_i}{2} + \frac{y_i}{2} + \frac{1}{8} = \\ &= O\Phi_i - \frac{x_i}{2} + \frac{y_i}{2} + \frac{1}{8} \approx O\Phi_i - \left(\frac{x_i - y_i}{2} \right); \end{aligned}$$

$$O\Phi_i^B = O\Phi_i + \frac{x_i}{2} + \frac{y_i}{2} + \frac{1}{8} \approx O\Phi_i + \frac{x_i}{2} + \frac{y_i}{2} = O\Phi_i + \left(\frac{x_i + y_i}{2} \right);$$

$$O\Phi_i^C = O\Phi_i + \frac{x_i}{2} - \frac{y_i}{2} + \frac{1}{8} \approx O\Phi_i + \frac{x_i}{2} - \frac{y_i}{2} = O\Phi_i + \left(\frac{x_i - y_i}{2} \right);$$

$$O\Phi_i^D = O\Phi_i - \frac{x_i}{2} - \frac{y_i}{2} + \frac{1}{8} \approx O\Phi_i - \frac{x_i}{2} - \frac{y_i}{2} = O\Phi_i - \left(\frac{x_i + y_i}{2} \right).$$

Зрозумілі, що знаки ОФ можна визначати незалежно. Слід відмітити, що значення $(x_i - y_i)$ використовується у коловому інтерполюванні для визначення переходу між октантами, а отже його не потрібно обчислювати додатково.

Якщо всередину кола попали всі 4 точки, то $I = I_M$; якщо лише 3 точки, то $I = 3 \cdot I_M / 4$; якщо дві, то $I = I_M / 2$; якщо одна – то $I = I_M / 4$. Для підвищення швидкодії значення I_M ; $3 \cdot I_M / 4$; $I_M / 2$; $I_M / 4$ можна обчислювати лише одноразово в циклі підготовки.

Дослідимо точність апроксимації площі покриття, яку забезпечує використання чотирьох додаткових оцінювальних функцій. Абсолютна похибка дорівнює: $\max(\Delta S) = \frac{(V + H)^2}{8V^2H}$. Для нашого випадку $V = H = 2$, тому

$$\max(\Delta S) = \frac{1}{4}.$$

Для оцінювання максимальних абсолютних похибок апроксимації була розроблена програма формування кіл, яка під час інтерполювання розраховувала площу покриття пікселя з використанням методу додаткових оцінювальних функцій та порівнювала отримане значення з точним значенням. Розрахунки було проведено для всіх кіл з радіусами від 2 до 1024. Для кожного кола було також підраховано загальну кількість сформованих точок та кількість точок, для яких абсолютна похибка визначення площі покриття належить діапазонам: $\delta < 1/32$, $1/32 < \delta < 1/16$, $1/16 < \delta < 1/8$, та $\delta < 1/8$.

Точність апроксимації площі покриття може бути підвищена за рахунок використання більшої кількості допоміжних точок. Дослідимо

максимальну абсолютну похибку апроксимації площі покриття при використанні восьми додаткових оцінювальних функцій, які обчислені в допоміжних точках, розташованих так, як показано на рис. 2.3.

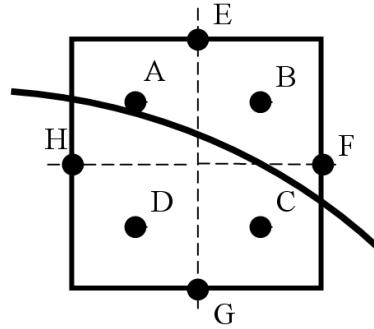


Рис. 2.3. Розміщення додаткових точок для розрахунку ОФ

Оцінювальні функції в даному випадку розраховуються за виразами:

$$O\Phi_i^E = x_i^2 + (y_i + 1/2)^2 - R^2 = (x_i^2 + y_i^2) - R^2 + y_i + \frac{1}{4} = O\Phi_i + y_i + \frac{1}{4};$$

$$O\Phi_i^F = (x_i + 1/2)^2 + y_i^2 - R^2 = (x_i^2 + y_i^2) - R^2 + x_i + \frac{1}{4} = O\Phi_i + x_i + \frac{1}{4};$$

$$O\Phi_i^G = x_i^2 + (y_i - 1/2)^2 - R^2 = (x_i^2 + y_i^2) - R^2 - y_i + \frac{1}{4} = O\Phi_i - y_i + \frac{1}{4};$$

$$O\Phi_i^H = (x_i - 1/2)^2 + y_i^2 - R^2 = (x_i^2 + y_i^2) - R^2 - x_i + \frac{1}{4} = O\Phi_i - x_i + \frac{1}{4}.$$

По знаках наведених оцінювальних функцій легко знайти площу перекриття пікселя і виконати антиаліайзинг крокових траєкторії

2.2 Метод модифікованої оцінювальної функції для антиаліайзингу меж кіл

Пропонується для задач антиаліайзингу використати модифіковану оцінювальну функцію.

У більшості алгоритмів колової інтерполяції використовується така оцінювальна функція: $OF = x^2 + y^2 - R^2$.

Пропонується використати модифіковану оцінювальну функція, яка визначається за формулою:

$$OF' = \frac{2^q}{2R} x^2 + \frac{2^q}{2R} y^2 - \frac{2^q}{2R} R^2. \quad (2.1)$$

Знак OF' збігається зі знаком оцінювальної функції OF .

Із урахуванням виразу (2.1) отримуємо формулу для знаходження відстані від центра піксела до дуги кола:

$$d_i = \frac{-OF'}{2^q}. \quad (2.2)$$

Визначимо, як змінюється модифікована оцінювальна функція для різних крокових приростів. У подальшому будемо розглядати випадок належності траєкторії до другого октанту та вважати, що центр кола збігається з початком координат. Формули для інших випадків отримуються аналогічно.

Для другого октанту при виконанні кроку $(x_{i+1} = x_i + 1; y_{i+1} = y_i)$ значення OF визначається за формулою:

$$OF'_{i+1} = \frac{2^q}{2R} (x+1)^2 + \frac{2^q}{2R} y^2 - \frac{2^q R}{2} OF'_i + \frac{2^q}{R} x + \frac{2^q}{2R}. \quad (2.3)$$

Позначимо $g(x) = \frac{2^q}{R} x$.

Останній вираз можна обчислювати рекурсивно за формулою:

$$g(x+1) = g(x) + \frac{2^q}{R}, \text{ враховуючи, що } g(0) = 0.$$

Для кроку $(x_{i+1} = x_i + 1; y_{i+1} = y_i - 1)$, OF розраховується за виразом:

$$OF'_{i+1} = \frac{2^q}{2R} (x+1)^2 + \frac{2^q}{2R} (y-1)^2 - \frac{2^q R}{2} = OF'_i + \frac{2^q}{R} x - \frac{2^q}{R} y + \frac{2^q}{R}.$$

Позначимо $l(y) = \frac{2^q}{R} y$.

Використаємо таку формулу:

$$l(y-1) = l(y) - \frac{2^q}{R}, \text{ при цьому } l(R) = 2^q.$$

Для прикладу, в якості базового алгоритму використаємо алгоритм колової інтерполяції за методом оцінювальної функції з початковим значенням $R-1/4$ [13]. У випадку використання модифікованої оцінювальної функції її початкове значення буде обчислюватись за формулою:

$$OF'_0 = \frac{2^q}{2R} \left(R - \frac{1}{4} \right) = 2^{q-1} - \frac{2^q}{8R}.$$

Початкове значення оцінювальної функції необхідно врахувати при визначенні відстані від центра пікселя до дуги кола. Нехай $OF_m = OF' + OF'_0$. З врахуванням виразу (4.3) отримуємо формулу:

$$OF_m = OF' + 2^{q-1} - \frac{2^q}{8R} = -d \cdot 2^q + 2^{q-1} - \frac{2^q}{8R}.$$

З останнього виразу знаходимо:

$$d = -\frac{OF_m}{2^q} + \frac{1}{2} - \frac{1}{8R}.$$

Якщо оцінювальна функція має додатний знак, то її значення збігається із значенням оцінювальної функції в точці дискретного простору, яка обирається на даному кроці [13]. Для обчислення відстані від центра пікселя до прямої можна використати значення оцінювальної функції, отримане на даному кроці інтерполяції.

Якщо оцінювальна функція має від'ємний знак, то її значення збігається із значенням оцінювальної функції у „прогнозуючій” точці, яка розташована на одну дискрету нижче дійсної. Значення оцінювальної функції в прогнозуючій точці обчислюється за виразом:

$$OF'_{\text{проз.}} = \frac{2^q}{2R}x^2 + \frac{2^q}{2R}(y-1)^2 - \frac{2^q R}{2} + OF'_0 = \frac{2^q}{2R}x^2 + \frac{2^q}{2R}y^2 - \frac{2^q R}{2} + OF' - \frac{2^q}{R}y + \frac{2^q}{2R}.$$

Знайдемо значення оцінювальної функції в дійсній точці:

$$OF'_{\text{дійсн.}} = \frac{2^q}{2R}x^2 + \frac{2^q}{2R}y^2 - \frac{2^q R}{2} + OF'_0 = OF'_{\text{проз.}} + l(y) - \frac{2^q}{2R}.$$

Враховуючи наведені формули можна запропонувати такий алгоритм антиаліайзингу границі кола:

1. Початкові значення:

$$g := 0; l := 2^q; T := \left\lfloor \frac{2^q}{R} + \frac{1}{2} \right\rfloor; OF_m := 2^{q-1} - \left\lfloor \frac{T}{8} \right\rfloor; x := 0; y := R;$$

$$d(0, R) := 0.$$

Дані операції утворюють цикл підготування до подальшого інтерполювання.

2. Якщо $x > y$, то переходимо до кроку 7.

3. Якщо $OF_m \geq 0$, то $OF_m := OF_m + g - l + T$;

$$\text{інакше} \quad OF_m := OF_m + g + T/2;$$

4. Якщо $OF_m \geq 0$, то $x := x+1; y := y-1; g := g+T; l := l-T; OF_{\text{дійсн.}} = OF_m$,

$$\text{інакше} \quad x := x+1; g := g+T; OF_{\text{дійсн.}} = OF_m + l - T/2.$$

5. Розраховуємо відстань від точки (x, y) до дуги кола:

$$d = \frac{-OF_{\text{дійсн.}}}{2^q} + \frac{1}{2} - \frac{1}{8R};$$

За значенням d розраховується інтенсивність кольору пікселя.

6. Перехід до п. 2.

7. Кінець.

У запропонованому підході відсутні довгі операції в циклі інтерполювання та використовується лише цілочисельна арифметика.

Визначимо значення параметра q , при якому накопичення похибки в процесі інтерполяції не вплине на результат обчислення відстані від центра пікселя до дуги кола.

Абсолютна похибка, яка має місце за рахунок округлення параметра T , розраховується за формулою:

$$\Delta T = \frac{2^q}{R} - \left\lfloor \frac{2^q}{R} + \frac{1}{2} \right\rfloor.$$

З останнього виразу видно, що максимальна абсолютна похибка ΔT_{\max} дорівнює $\frac{1}{2}$.

Абсолютна похибка обчислення функції $g(x) = \frac{2^q}{R}x$ на i -му кроці інтерполяції знаходиться за виразом:

$$\Delta g_i = \frac{2^q}{R}i - \left\lfloor \frac{2^q}{R}i \right\rfloor = i \cdot \Delta T.$$

Абсолютна похибка обчислення функції $l(y) = \frac{2^q}{R}y$ дорівнює:

$$\Delta l_j = j \cdot \Delta T.$$

Абсолютна похибка обчислення значення оцінювальної функції на i -му кроці інтерполяції дорівнює:

$$\Delta OF'_i = \sum_{k=1}^i \Delta g_k + \sum_{k=1}^i \Delta l_k + i \cdot \Delta T.$$

Максимальне значення абсолютної похибки можна знайти за виразом:

$$\begin{aligned} \Delta OF'_{\max} &= \sum_{k=1}^N k \cdot \Delta T_{\max} + \sum_{k=1}^M k \cdot \Delta T_{\max} + N \cdot \Delta T_{\max} = \Delta T_{\max} \frac{(N^2 + N)}{2} + \Delta T_{\max} \frac{(M^2 + M)}{2} + \\ &+ N \cdot \Delta T_{\max} = \Delta T_{\max} \frac{(N^2 + 3N + M^2 + M)}{2}, \end{aligned}$$

де N - загальна кількість ітерацій циклу інтерполювання, $N = R/\sqrt{2}$;

M - кількість діагональних кроків, $M = R - R/\sqrt{2}$.

Вираз для максимального значення абсолютної похибки обчислення відстані від центра піксела до дуги кола має вигляд:

$$\Delta d_{\max} = \frac{\Delta OF'_{\max}}{2^q}.$$

Якщо задано, що відстань потрібно обчислити з точністю до 2^{-P} , то має виконуватись нерівність: $\Delta d_{\max} \leq 2^{-(p+1)}$.

Перепишемо останню нерівність із врахуванням попередньо отриманих виразів: $\frac{\Delta T_{\max}}{2} (2R^2 + R\sqrt{2} - R^2\sqrt{2} + R) 2^{-q} \leq 2^{-(p+1)}$.

Після перетворень отримуємо: $q \geq p + \log_2(2R^2 + R\sqrt{2} - R^2\sqrt{2} + R) - 1$.

У таблиці 2.1 наведено значення параметра q , які розраховані для різних максимальних значень радіуса кола та параметра p .

Використання модифікованої оцінювальної функції дозволило при визначенні відстані від центра піксела, виключити з обчислювального процесу довгі операції в циклі інтерполювання, що дозволяє при антиаліайзингу суттєво підвищити швидкодію та спростити апаратну реалізацію.

Таблиця 2.1

Значення параметра q для різних діапазонів значень
радіуса кола та параметра p

R_{\max}	Значення параметра q							
	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$	$p = 7$	$p = 8$
0 – 512	18	19	20	21	22	23	24	25
0 – 1024	20	21	22	23	24	25	26	27
0 – 2048	22	23	24	25	26	27	28	29
0 – 4096	24	25	26	27	28	29	30	31

2.3 Модифікація табличного методу антиаліазингу меж кіл

При обчисленні площі покриття піксела колом найбільш трудомісткою процедурою є розрахунок значення виразу $\sqrt{R^2 - x^2}$. Один з можливих підходів до підвищення продуктивності крайового антиаліазингу границь кіл полягає у використанні табличного методу для розрахунку значення $\sqrt{R^2 - x^2}$. У методі, що був запропонований Ксяоліном Ву [11], для згладжування границь кіл використовується таблиця заздалегідь розрахованих значень функції:

$$D(r, j) = \lfloor (2^m - 1) \left(\sqrt{r^2 - j^2} - \sqrt{r^2 - j^2} \right) + 0.5 \rfloor,$$

де r - радіус кола; $1 \leq j \leq \frac{r}{\sqrt{2}}$.

Якщо R_{\max} - максимальне значення радіуса кола, то для антиаліазингу границі кола необхідно зберігати $\left\lceil \frac{\sqrt{2}}{4} R_{\max}^2 \right\rceil$ значень функції $D(r, j)$ [14].

Використання відносно великих масивів пам'яті для значень радіусів кіл, більших за 2048, обмежує використання даного підходу. Наприклад: для випадку, коли максимальне значення радіусу кола дорівнює 4096, таблиця значень функції $D(r, j)$ займає 5,8 Мбайт пам'яті.

Висока швидкодія згладжування дуг кіл, яка досягається при використанні табличного методу обумовлює актуальність пошуку нових способів зберігання таблиці значень функції $\sqrt{R^2 - x^2}$. Таблично-алгоритмічний підхід [11] дозволяє збільшити крок квантування і таким чином зменшити необхідні об'єми пам'яті, однак при цьому використовується операція множення, яка ускладнює апаратну реалізацію та зменшує швидкодію антиаліазингу кола.

Пропонується новий підхід до обчислення значення функції $\sqrt{R^2 - x^2}$, який дозволяє зменшити необхідний об'єм постійної пам'яті і при цьому не містить „довгих” операцій у обчислювальному процесі.

Позначимо $T = R^2 - x^2$, $g(T) = \sqrt{T}$. Значення T змінюється в діапазоні $T \in [0, 1 \dots R_{\max}^2]$, тому для обчислення площі покриття піксела колом необхідно зберігати R_{\max}^2 значень функції $g(T)$. Розглянемо підхід до зменшення об'єму постійної пам'яті за рахунок апроксимації функції $g(T)$.

Представимо T у вигляді:

$$T = A \cdot 2^p + B, \quad (2.4)$$

де $A = \left\lfloor \frac{T}{2^p} \right\rfloor$, $B = T - A \cdot 2^p$, $p < m$, m - розрядність T .

Розкладемо функцію $g(T)$ в околі точки $T = A \cdot 2^p$ у ряд Тейлора, обмежившись першими двома членами:

$$g(A \cdot 2^p + B) \approx g(A \cdot 2^p) + g'(A \cdot 2^p)B \quad (2.5)$$

Для наближеного обчислення похідної функції $g(T)$ в точці $T = A \cdot 2^p$ використаємо вираз:

$$g'(A \cdot 2^p) \approx \frac{g(A \cdot 2^p + B \cdot 2^p) - g(A \cdot 2^p)}{B \cdot 2^p}.$$

Підставивши значення останнього виразу в формулу (2.5), отримуємо формулу для наближеного розрахунку значення функції $g(T)$ в точці $T = A \cdot 2^p + B$:

$$g(A \cdot 2^p + B) \approx g(A \cdot 2^p) + \frac{g((A+B)2^p) - g(A \cdot 2^p)}{2^p}.$$

Позначимо $G(x) = \sqrt{x \cdot 2^p}$ та перепишемо попередній вираз у вигляді:

$$g(A \cdot 2^p + B) \approx G(A) + \frac{G(A+B) - G(A)}{2^p} \quad (2.6)$$

З останньої формули видно, що для наближеного розрахунку значення функції $g(T)$, можна використовувати таблицю значень функції $G(x)$, $x \in [0, 1 \dots \max(A+B)]$. Максимальне значення, яке може приймати B дорівнює $2^p - 1$ (див. формулу (2.4)). Максимальне значення A дорівнює $2^{m-p} - 1$. Таким чином, необхідно зберігати $2^p + 2^{m-p} - 2$ значень функції $G(x)$.

Для випадку, коли $T < 2^p + 2^{m-p} - 2$ значення функції $g(T)$ пропонується розраховувати за виразом:

$$g(T) = \sqrt{T} = \frac{\sqrt{T \cdot 2^p}}{\sqrt{2^p}} = \frac{G(T)}{\sqrt{2^p}}. \quad (2.7)$$

Оскільки відоме точне значення $G(T)$, то розрахунок значення функції $g(T)$ за формулою (2.7) суттєво зменшує похибку обчислень. Вибір парного значення p дозволяє спростити обчислення за останньою формулою, оскільки при цьому операція ділення на $\sqrt{2^p}$ замінюється операцією зсуву на $p/2$ розрядів в бік молодших розрядів. На основі наведених вище виразів можна запропонувати таку формулу для наближеного розрахунку значення функції $g(T) = \sqrt{T}$ з використанням таблиці значень функції $G(x) = \sqrt{x \cdot 2^p}$:

$$g_{\text{набл}}(T) = \begin{cases} \frac{G(T)}{\sqrt{2^p}}, & \text{якщо } T < 2^p + 2^{m-p} - 2; \\ G(A) + \frac{G(A+B) - G(A)}{2^p}, & \text{якщо } T \geq 2^p + 2^{m-p} - 2. \end{cases} \quad (2.8)$$

Для оцінення максимальних абсолютних похибок апроксимації функції $g(T)$ при розрахунках за виразом (2.8) була розроблена програма, у якій для заданих значень R_{\max} та p розраховувалось значення функції $g(T)$ у точках $T = 0, 1, \dots, R_{\max}^2$ з використанням виразу (2.8) та порівнювалось з точним значенням. Результати обчислень наведено в додатку В у таблиці В.1. Значення p , які є парними і для яких максимальна похибка апроксимації не перевищує 2^{-9} , в таблиці виділені курсивом.

Визначимо кількість двійкових розрядів, необхідну для зберігання значень функції $G(x) = \sqrt{x \cdot 2^p}$. Враховуючи, що максимальне значення x дорівнює $2^p + 2^{m-p} - 2$, то кількість розрядів для зберігання цілої частини значень функції $G(x)$ може бути визначена за формулою:

$$q_y = \left\lfloor \log_2 \left(\sqrt{(2^p + 2^{m-p} - 2)2^p} \right) \right\rfloor + 1. \quad (2.9)$$

Для визначення кількості розрядів q_δ , необхідної для зберігання дробової частини значень функції $G(x)$, було розроблено програму, яка при заданих значеннях R_{\max} , p та q_δ розраховувала значення функції $g(T)$ у точках $T = 0, 1, \dots, R_{\max}^2$ з використанням виразу (2.8) та значень функції $G(x)$ представлених з точністю до q_δ розрядів. Наближене значення порівнювалось з відповідним точним значенням та визначалось максимальне значення абсолютної похибки. Результати розрахунків наведено у таблиці В.2 додатка В.

Можна запропонувати такий алгоритм обчислення значення функції $\sqrt{R^2 - x^2}$:

1. $T = R^2 - x^2$ представляється у вигляді: $T = A \cdot 2^p + B$. Для цього розраховують: $A = \left\lfloor \frac{T}{2^p} \right\rfloor$, $B = T - A \cdot 2^p$.

2. Якщо $T < 2^p + 2^{m-p} - 2$, то з пам'яті зчитується значення $G(T)$ і обчислюється $g(T) = G(T) \cdot 2^{-\frac{p}{2}}$. Після цього виконується перехід до п.5.

3. З пам'яті зчитуються значення $G(A)$ та $G(A+B)$.

4. Розраховується $g(T) = G(A) + \frac{G(A+B) - G(A)}{2^p}$.

5. $\sqrt{R^2 - x^2} = g(T)$.

6. Кінець.

При апаратній реалізації можливе паралельне виконання операцій вибірки з пам'яті та операцій додавання, а також використання монтажного зсуву, що дозволить додатково зменшити час обчислень.

Висновки

1. Запропоновано метод антиаліазингу, який оснований на розбитті піксела на складові та обчислення в них знаків оцінювальних функцій, що дозволило спростити обчислення площі покриття піксела кроковою траєкторією, і, як наслідок, підвищити продуктивність антиаліазингу.

2. Розроблено алгоритм табличної реалізації антиаліазингу. Запропонований алгоритм не містить довгих операцій, що обумовлює його високу швидкодію. При програмній реалізації розглянутого алгоритму для обчислення значення функції $\sqrt{R^2 - x^2}$ у найгіршому випадку потрібно виконати: одну операцію зсуву; одну операцію віднімання; одну операцію порівняння; 2 операції зчитування з пам'яті; операцію віднімання; операцію зсуву та операцію додавання.

3 РОЗРОБКА МЕТОДІВ АНТИАЛІАЙЗИНГУ МЕЖ ЕЛІПСІВ І ГІПЕРБОЛ

3.1 Антиаліайзинг меж еліпсів

У розділі 1 показано, що у більшості існуючих алгоритмів передбачено те, що інтенсивності кольору точок еліпса та фонового зображення є сталими. Для більш загального випадку, коли інтенсивність кольору еліпса та фонового зображення, на яке накладається еліпс, не є сталими і можуть змінюватись довільним чином, у циклі інтерполювання необхідне виконання операції ділення та множення, що призводить до ускладнення обчислювального процесу та обмежує швидкодію. Таким чином, актуальною є розробка нових підходів до антиаліайзингу границь еліпсів, які б враховували можливість зміни інтенсивностей кольору еліпса і фонового зображення, та характеризувалися простотою обчислювального процесу.

З метою спрощення обчислювальної складності для антиаліайзингу границь еліпсів будемо використовувати квадратну модель піксела [11]. Вибір такої моделі обумовлено тим, що для інших моделей необхідно знаходити відстань від центра піксела до траєкторії, що у випадку еліпса призводить до значного збільшення обчислювальної складності розрахунків [14].

Відповідно до обраної моделі піксел розглядається як квадрат із стороною, що дорівнює одній дискреті. Центр квадрата збігається з центром піксела. Інтенсивність кольору піксела будемо встановлювати пропорційно до площі, яка відтинається границею еліпса від піксела. Для визначення інтенсивності кольору використаємо вираз [12]:

$$I_p(i, j) = S \cdot I_M + (1 - S) \cdot I_\phi, \quad (3.1)$$

де S – площа, яка відтинається від піксела (i, j) границею еліпса, $0 \leq S \leq 1$; I_M – інтенсивність кольору, з якою треба відтворити еліпс.

Для випадку кольорового зображення вираз (3.1) розраховується окремо для інтенсивності кожної з RGB компонент кольору.

Запишемо вираз (3.10) у вигляді:

$$I_p(i, j) = S \cdot I_M + (1 - S) \cdot I_\phi = S \cdot (I_M - I_\phi) + I_\phi.$$

Розглянемо процес визначення координат точок траєкторії. Будемо розглядати еліпс, центр якого знаходиться в початку координат. Рівняння еліпсу має такий вигляд [13? 14]:

$$\frac{x^2}{B^2} + \frac{y^2}{A^2} = 1.$$

При формуванні траєкторії розглянемо дві області, розділені точкою, в якій $dy/dx = -1$. У першій області на кожній ітерації циклу інтерполювання виконується крок у напрямку осі x , а в другій – у напрямку осі y (рис. 3.1).

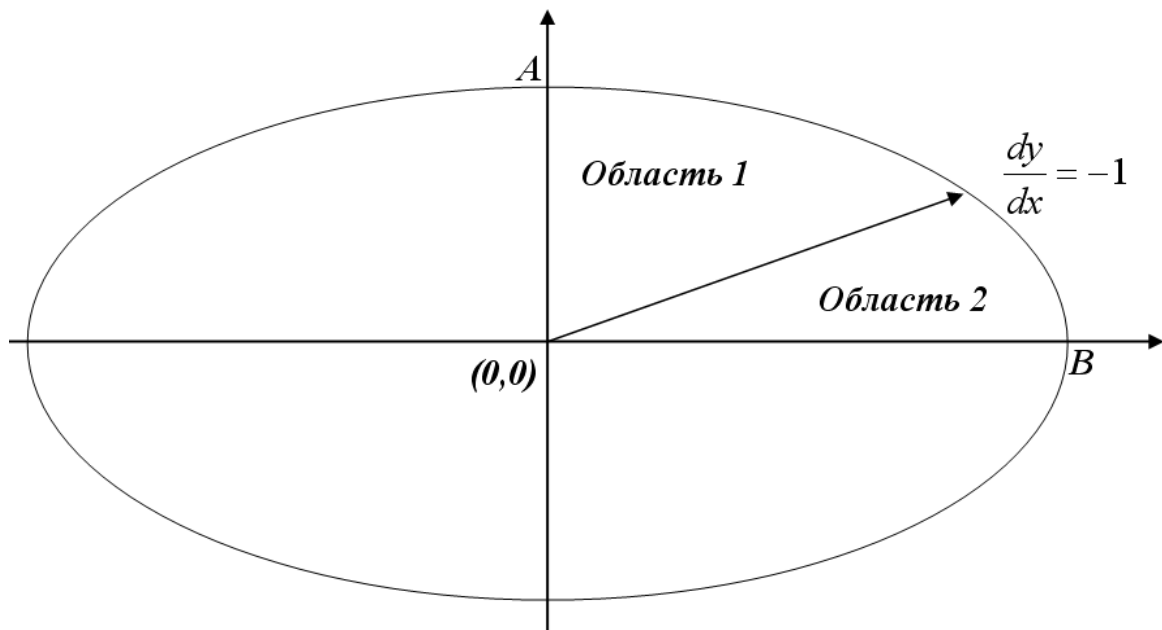


Рис. 3.1. Еліпс

Розглянемо формування крокової траєкторії в області 1. Будемо визначати координати точок за годинниковою стрілкою, починаючи з точки $(0, A)$. Рівняння еліпса для області 1 можна записати у вигляді :

$$f(x) = \sqrt{A^2 - x^2} \frac{A^2}{B^2}.$$

В області 1 у кожному інтерполяційному такті виконується крок у напрямку осі x . Якщо на попередньому кроці було обрано піксел $(i-1, j)$, то наступний піксел обирається з тих, що мають координати (i, j) та $(i, j-1)$. Критерієм для вибору є мінімальна відстань від центра пікселя до точки $(i, f(i))$, у якій ідеальна траєкторія перетинає координатну решітку (рис. 3.2).

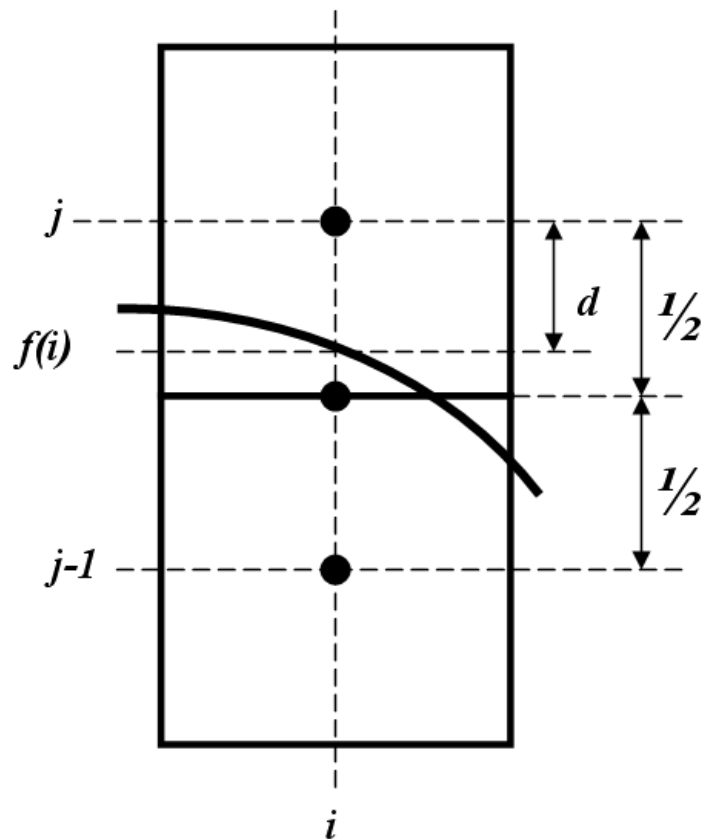


Рис. 3.2. Визначення координат точок траєкторії

Для піксела з координатами (i, j) дану відстань можна знайти за формулою:

$$d(i, j) = f(i) - j. \quad (3.2)$$

Для забезпечення максимальної точності при формуванні крокової траєкторії повинна виконуватись умова: $-0,5 \leq d(i, j) \leq 0,5$. З урахуванням виразу (3.2) отримуємо нерівність: $-0,5 \leq f(i) - j \leq 0,5$, яка після перетворення має вигляд:

$$j - 0,5 \leq f(i) \leq j + 0,5. \quad (3.3)$$

Для визначення типу крокового приросту можна використати оцінювальну функцію вигляду:

$$OF(i, j) = f(i) - (j - 0,5). \quad (3.4)$$

Якщо $OF(i, j) > 0$, то виконується крок у напрямку осі x і обирається піксел (i, j) . В іншому випадку виконується діагональний крок і обирається піксел $(i, j - 1)$. Для обчислення функції $f(i)$ необхідно виконати трудомістку операцію знаходження квадратного кореня. З метою спрощення обчислень значення доданків у виразі (3.13) доцільно замінити їх квадратами. Оскільки $f(i) \geq 0$ та $(j - 0,5) > 0$, то знак функції не зміниться. В результаті отримуємо такий вираз для оцінювальної функції:

$$OF'(i, j) = f^2(i) - (j - 0,5)^2. \quad (3.5)$$

Знайдемо рекурентні вирази для обчислення оцінювальної функції. Розглянемо, як змінюється перший доданок у виразі (3.5) при виконанні елементарного кроку у напрямку осі x :

$$f^2(i+1) = A^2 - (i+1)^2 \frac{A^2}{B^2} = \left(A^2 - i^2 \frac{A^2}{B^2} \right) - 2i \frac{A^2}{B^2} - \frac{A^2}{B^2} = f^2(i) - \frac{A^2}{B^2} (2i+1).$$

Нехай $L(i) = \frac{A^2}{B^2} (2i+1)$, тоді

$$L(i+1) = \frac{A^2}{B^2}(2(i+1)+1) = \frac{A^2}{B^2}(2i+1) + 2\frac{A^2}{B^2} = L(i) + 2\frac{A^2}{B^2}.$$

Із урахуванням останнього виразу отримуємо формулу для рекурсивного визначення першого доданка у виразі (3.5) :

$$f^2(i+1) = f^2(i) - L(i).$$

Розглянемо, як змінюється другий доданок у виразі (3.5) при виконанні діагонального кроку:

$$((j-1)-0,5)^2 = j^2 - 2j + 1 - j + 1 + 0,25 = (j-0,5)^2 - 2(j-1).$$

Абсолютна похибка, яка має місце при обчисленні виразу (3.5) за рахунок округлення операндів до q двійкових розрядів, може розрахована згідно з виразом:

$$\delta_a = A^2 - x^2 \frac{A^2}{B^2} - A^2 + x^2 \left[\frac{A^2}{B^2} 2^q + 0,5 \right] 2^{-q} = 2^{-q} x^2 \left(\left[\frac{A^2}{B^2} 2^q + 0,5 \right] - \frac{A^2}{B^2} 2^q \right).$$

З останнього виразу випливає, що максимальне значення абсолютної похибки δ_a дорівнює:

$$\delta_{a\max} = B^2 2^{-(q+1)}.$$

Якщо задано граничне значення абсолютної похибки обчислення виразу (3.14) $\delta_2 = 2^{-p}$, то мінімальне значення параметра q , яке забезпечить задану величину абсолютної похибки δ_a , можна знайти за виразом:

$$q = p + \log_2(B^2) - 1.$$

У запропонованому алгоритмі відсутні “довгі” операції, що забезпечує його високу швидкодію та простоту апаратної реалізації. Розглянемо можливість використання виразу (3.5) для усунення ефекту аліайзингу границі еліпса.

Перепишемо формулу для S у вигляді:

$$S = \frac{1}{2} + \frac{A^2 - x^2 A^2 / B^2 - y^2 - 0,5 A^2 / B^2}{2y}.$$

З урахуванням виразу (5):

$$S = \frac{f^2(x) - y^2 + y - 0,25 + 0,25 - 0,5A^2/B^2}{2y} = \frac{OF' + 0,25 - 0,5A^2/B^2}{2y}.$$

Із урахуванням виразу (3.5) отримуємо формулу для визначення інтенсивності кольору пікселя:

$$I_p(x, y) = \frac{OF' + 0,25 - 0,5A^2/B^2}{2y} (I_M - I_\Phi) + I_\Phi. \quad (3.6)$$

Розглянемо шляхи заміни операції ділення та множення на більш прості операції. При цьому будемо враховувати, що у переважній більшості сучасних графічних систем для значення інтенсивності кольору виділяється 1 байт по кожному з RGB каналів. Для такого випадку максимальна кількість рівнів інтенсивності дорівнює $2^8 = 256$. Отже, результат операції ділення для нашого випадку достатньо розрахувати з точністю до $1/256$.

Припустимо, що задано вираз:

$$I_p = \frac{x}{M} I_M, \quad (3.7)$$

де $x \leq M$; $I_M < 256$.

Розглянемо застосування методу послідовних наближень для обчислення виразу (3.16). При цьому використаємо принцип дихотомії [16], згідно з яким значення x будемо урівноважувати значеннями $\frac{1}{2}M$, $\frac{1}{4}M$, $\frac{1}{8}M \dots$.

Представимо вираз x/M у вигляді:

$$\frac{x}{M} = \frac{1}{2}a_1 + \frac{1}{4}a_2 + \frac{1}{8}a_3 + \dots + \frac{1}{256}a_8 + \varepsilon, \quad (3.8)$$

де $a_i = \{0,1\}$, $\varepsilon \leq \frac{1}{256}$.

Знайдемо значення I_p за формулою:

$$I_p = \frac{I_M}{2}a_1 + \frac{I_M}{4}a_2 + \frac{I_M}{8}a_3 + \dots + \frac{I_M}{256}a_8 + \varepsilon,$$

$$\text{де } a_i = \{0,1\} \quad \varepsilon \leq \frac{I_M}{256}.$$

При такому підході значення I_p можна знайти з використанням 8 операцій додавання при умові, що коефіцієнти $a_1 \div a_8$ відомі. Для визначення цих коефіцієнтів рівняння (3.17) запишемо у вигляді :

$$x = \frac{M}{2}a_1 + \frac{M}{4}a_2 + \frac{M}{8}a_3 + \dots + \frac{M}{256}a_8 + \varepsilon,$$

$$\text{де } a_i = \{0,1\} ; \quad \varepsilon \leq M/256.$$

Для визначення коефіцієнта a_1 останнє рівняння запишемо у вигляді:

$$x - \frac{M}{2}a_1 = \frac{M}{4}a_2 + \frac{M}{8}a_3 + \dots + \frac{M}{256}a_8 + \varepsilon. \quad (3.9)$$

Враховуючи, що $\varepsilon \leq M/256$, то має місце нерівність:

$$0 \leq \frac{M}{4}a_2 + \frac{M}{8}a_3 + \dots + \frac{M}{256}a_8 + \varepsilon \leq \frac{M}{2}.$$

Із виразу (3.18) отримуємо:

$$0 \leq x - \frac{M}{2}a_1 \leq \frac{M}{2}.$$

Якщо $a_1 = 0$, то має місце нерівність: $0 \leq x \leq \frac{M}{2}$. У випадку, коли $a_1 = 1$, справедливою є нерівність $\frac{M}{2} \leq x \leq M$. З отриманих нерівностей випливає правило для визначення коефіцієнта a_1 . Якщо $x < \frac{M}{2}$, то $a_1 = 0$, інакше $a_1 = 1$.

Правило для визначення коефіцієнта a_2 знаходимо аналогічно.

Для цього позначимо $k_1 = x - \frac{M}{2}a_1$ і запишемо рівняння (3.18) у вигляді :

$$k_1 - \frac{M}{4}a_2 = \frac{M}{8}a_3 + \dots + \frac{M}{256}a_8 + \varepsilon \quad (3.10)$$

Оскільки $\varepsilon \leq M/256$, то справедливою є нерівність:

$$0 \leq \frac{M}{8}a_3 + \dots + \frac{M}{256}a_8 + \varepsilon \leq \frac{M}{4}$$

Із урахуванням виразу (3.10) остання нерівність має такий вигляд:

$$0 \leq k_1 - \frac{M}{4}a_2 \leq \frac{M}{4}. \quad (3.11)$$

$$a_2 = 0 \text{ за умови, що } 0 \leq k_1 \leq \frac{M}{4}. \quad a_2 = 1, \text{ коли } \frac{M}{4} \leq k_1 \leq \frac{M}{2}. \text{ Із}$$

останніх двох нерівностей отримуємо правило для визначення коефіцієнта a_2 .

Якщо $k_1 < \frac{M}{4}$, то $a_2 = 0$, інакше $a_2 = 1$.

Аналогічним чином можна отримати правила і для знаходження всіх восьми коефіцієнтів. У загальному вигляді це правило має такий вигляд :

$$a_i = 0, \text{ якщо } k_{i-1} < \frac{M}{2^i}; \text{ інакше } a_i = 1. \quad k_0 = x \quad ; \quad k_i = k_{i-1} - \frac{M}{2^i}a_i.$$

Запропонований алгоритм містить лише мікрооперації додавання та зсуву, що забезпечує простоту його апаратної реалізації.

Продуктивність підходу можна підвищити за рахунок зменшення кількості ітерацій N , що безумовно вплине на якість. Слід зазначити, що при цьому максимальна похибка не буде перевищувати значення I_M/N .

Із урахуванням наведених вище виразів отримуємо алгоритм антиаліайзingu, який оснований на визначенні координат точок траєкторії еліпса. Розрахунок інтенсивностей кольору точок виконується згідно з виразом (3.15).

Використання нового підходу до визначення інтенсивностей кольору точок дозволило отримати алгоритм антиаліайзingu, який на відміну від алгоритмів [1], не потребує виконання “довгих” операцій для всіх випадків,

включаючи і той, коли інтенсивність кольору еліпса та фонового зображення, на яке накладається еліпс, не є сталими і можуть змінюватись довільним чином у циклі інтерполювання. Це забезпечує алгоритму високу швидкість та простоту апаратної реалізації. У розробленому алгоритмі передбачено можливість корегування якості антиаліазингу за рахунок використання різної кількості рівнів інтенсивностей кольору.

3.2 Антиаліазинг меж гіперболи

Запропонований підхід до антиаліазингу границь еліпсів можна використати і для інших кривих другого порядку. Розглянемо процедуру антиаліазингу для гіперболи.

У подальшому будемо розглядати гіперболу, центр якої знаходиться в початку координат. Канонічне рівняння гіперболи має такий вигляд [11, 12]:

$$\frac{x^2}{B^2} - \frac{y^2}{A^2} = 1,$$

де A, B – півосі гіперболи, паралельні до відповідних координатних осей (рис. 3.3).

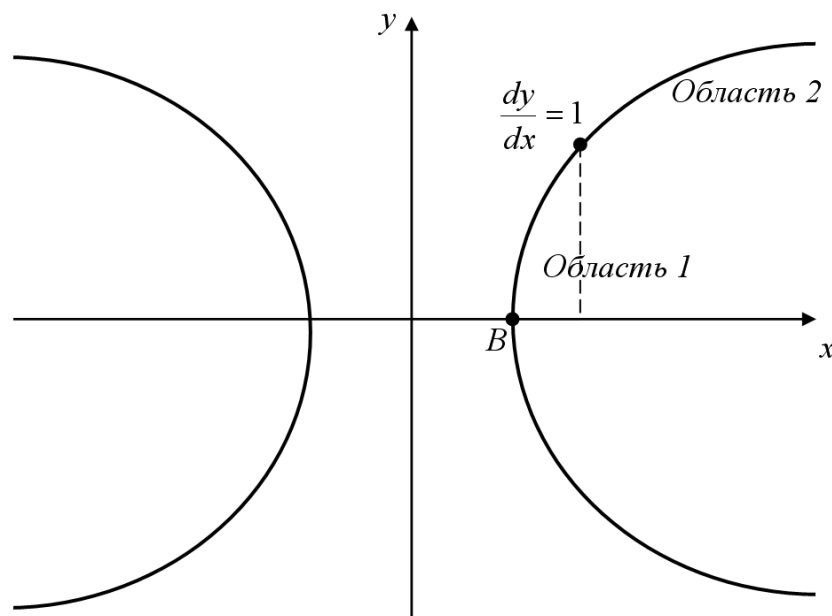


Рис. 3.3. Гіпербола

При формуванні траєкторії розглянемо дві області, розділені точкою, в якій $dy/dx=1$. У першій області при кожній ітерації циклу інтерполювання виконується крок у напрямку осі y , а в другій – у напрямку осі x (рис. 3.3).

Розглянемо формування крокової траєкторії в області 1. Будемо визначати координат точок за годинниковою стрілкою, починаючи з точки $(B,0)$. Рівняння гіперболи для області 1 можна записати у вигляді :

$$f(y) = \sqrt{B^2 + y^2} \frac{B^2}{A^2}.$$

В області 1 у кожному інтерполяційному такті виконується крок у напрямку осі y . Якщо на попередньому кроці було обрано піксел $(i, j-1)$, то наступний піксел обирається з тих, що мають координати (i, j) та $(i+1, j)$. Критерієм для вибору є мінімальна відстань від центра пікселя до точки $(f(j), j)$, у якій ідеальна траєкторія перетинає координатну решітку (рис. 3.4).

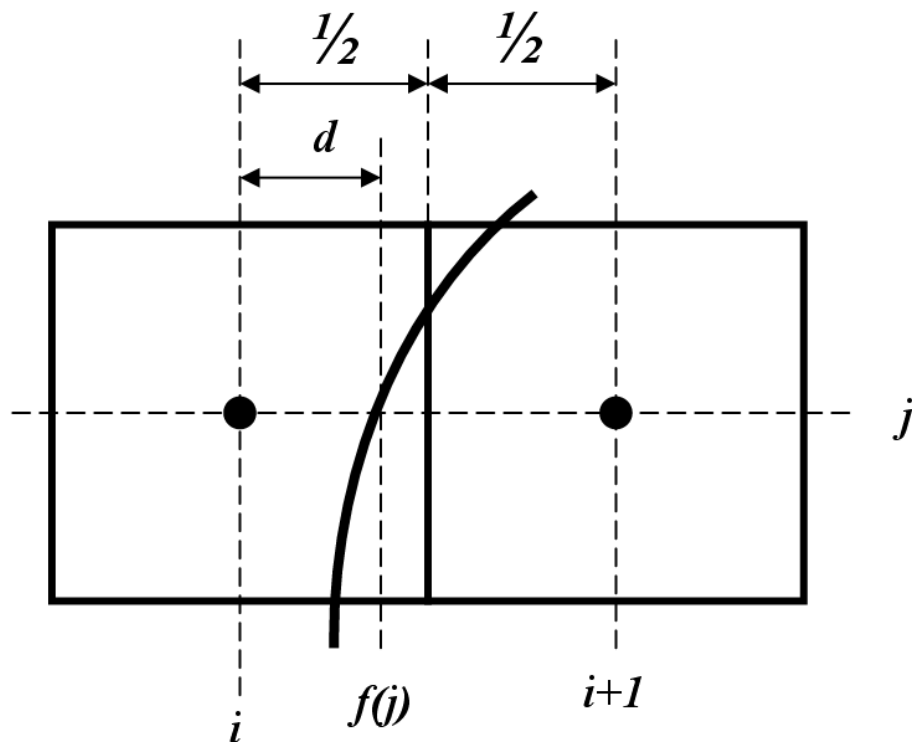


Рис. 3.4 Визначення координат точок траєкторії гіперболи

Для піксела з координатами (i, j) дана відстань може бути знайдена за формулою:

$$d(i, j) = f(j) - i. \quad (3.12)$$

Для забезпечення максимальної точності при формуванні крокової траєкторії повинна виконуватись умова: $-0,5 \leq d(i, j) \leq 0,5$. Із урахуванням виразу (3.21) отримуємо нерівність: $-0,5 \leq f(j) - i \leq 0,5$, яка після перетворення має вигляд:

$$i - 0,5 \leq f(j) \leq i + 0,5. \quad (3.13)$$

Для визначення типу крокового приросту можна використати оцінювальну функцію вигляду:

$$OF(i, j) = (i + 0,5) - f(j). \quad (3.14)$$

Якщо $OF(i, j) > 0$, то виконується крок у напрямку осі y і обирається піксел (i, j) , інакше виконується діагональний крок і обирається піксел $(i+1, j)$. Для обчислення функції $f(j)$ необхідно виконати трудомістку операцію знаходження квадратного кореня. З метою спрощення обчислень значення доданків у виразі (3.14) доцільно замінити їх квадратами. Оскільки $f(j) \geq 0$ та $(i + 0,5) > 0$, то знак функції не зміниться. В результаті отримаємо такий вираз для оцінювальної функції:

$$OF'(i, j) = (i + 0,5)^2 - f^2(j). \quad (3.15)$$

Знайдемо рекурентні вирази для обчислення оцінювальної функції. Розглянемо, як змінюється перший доданок у виразі (3.15) при виконанні елементарного кроку у напрямку осі y :

$$f^2(j+1) = B^2 + (j+1)^2 \frac{B^2}{A^2} = \left(A^2 + j^2 \frac{B^2}{A^2} \right) + 2j \frac{B^2}{A^2} + \frac{B^2}{A^2} = f^2(j) + \frac{B^2}{A^2} (2j+1).$$

Нехай $L(j) = \frac{B^2}{A^2} (2j+1)$, тоді

$$L(j+1) = \frac{B^2}{A^2}(2(j+1)+1) = \frac{B^2}{A^2}(2j+1) + 2\frac{B^2}{A^2} = L(j) + 2\frac{B^2}{A^2}.$$

Із урахуванням останнього виразу отримуємо формулу для рекурсивного визначення першого доданка у виразі (3.15) :

$$f^2(j+1) = f^2(j) + L(j).$$

Розглянемо, як змінюється другий доданок у виразі (3.15) при виконанні діагонального кроку:

$$((i+1)+0,5)^2 = i^2 + 3i + 2,25 = (i+0,5)^2 + 2(i+1).$$

Аналогічно можна отримати алгоритм визначення координат точок траєкторії і для області 2. Запропонований алгоритм не містить “довгих” операцій, що забезпечує його високу швидкість та простоту апаратної реалізації. Розглянемо можливість використання виразу (3.15) для усунення ефекту аліаїзingu границі еліпса.

Знайдемо вирази для розрахунку площі, що відтинається від пікселя границею гіперболи. Доведено, що всередині пікселя дугу можна замінити відрізком прямої, причому відносна похибка у найгіршому випадку складе не більше 0,4% [11]. На рис. 3.5. показано перетин траєкторією гіперболи дискретної координатної решітки.

Із рис. 3.3 видно, що для пікселя (x_i, y_i) площа перетину може бути знайдена за формулою:

$$S_1 = x_i + 1/2 - (f(y_i) - T/2),$$

де T - товщина траєкторії гіперболи.

Для пікселя $(x_i + 1, y_i)$ площа перетину знаходиться за формулою:

$$S_2 = f(y_i) + T/2 - (x_i + 1/2).$$

Розглянемо наближене обчислення функції $f(i)$ з використанням формули Герона.

Для нашого випадку: $a = B^2 + y^2 \frac{B^2}{A^2}$, тому початкове наближення доцільно взяти рівним координаті пікселя: $sqr_0 = y_i$.

Після першої ітерації наближене значення квадратного кореня дорівнює:

$$sqr_1 = \frac{1}{2} \left(sqr_0 + \frac{a^2}{sqr_0} \right) = \frac{1}{2} \left(x_i + \frac{B^2 + y_i^2 B^2 / A^2}{x_i} \right).$$

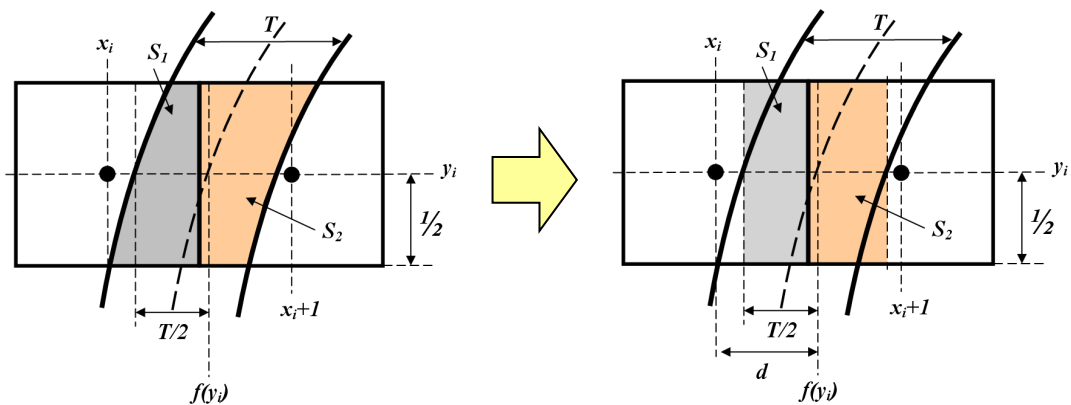


Рис. 3.5. Визначення площі покриття пікселя гіперболою

Перепишемо вираз для знаходження площі покриття пікселя (x_i, y_i) із урахуванням останньої формули:

$$\begin{aligned} S_1 &= x_i + 1/2 + T/2 - sqr_1 = x_i + 1/2 + T/2 - \frac{x_i^2 + B^2 + y_i^2 B^2 / A^2}{2x_i} = \\ &= T/2 + \frac{(x_i + 0,5)^2 - f^2(y_i) - 0,25}{2x_i} = \frac{OF'(x_i, y_i) - 0,25 + Tx_i}{2x_i}. \end{aligned}$$

Інтенсивність кольору пікселя (x_i, y_i) можна визначити за формулою:

$$\begin{aligned} I_p(x, y) &= S_1(I_M - I_{\text{фон}}) + I_{\text{фон}} = \left(\frac{OF' - 0,25 + T \cdot x}{2x} \right) (I_M - I_{\text{фон}}) + I_{\text{фон}} = \\ &= \frac{OF' - 0,25 + T \cdot x}{2x} (I_M - I_{\text{фон}}) + I_{\text{фон}}. \end{aligned}$$

Для обчислення першого доданку у попередньому виразі доцільно використати алгоритм розрахунку інтенсивностей кольору, який було

запропоновано для границь еліпсів. Це дозволить виключити довгі операції з обчислювального процесу.

Висновки

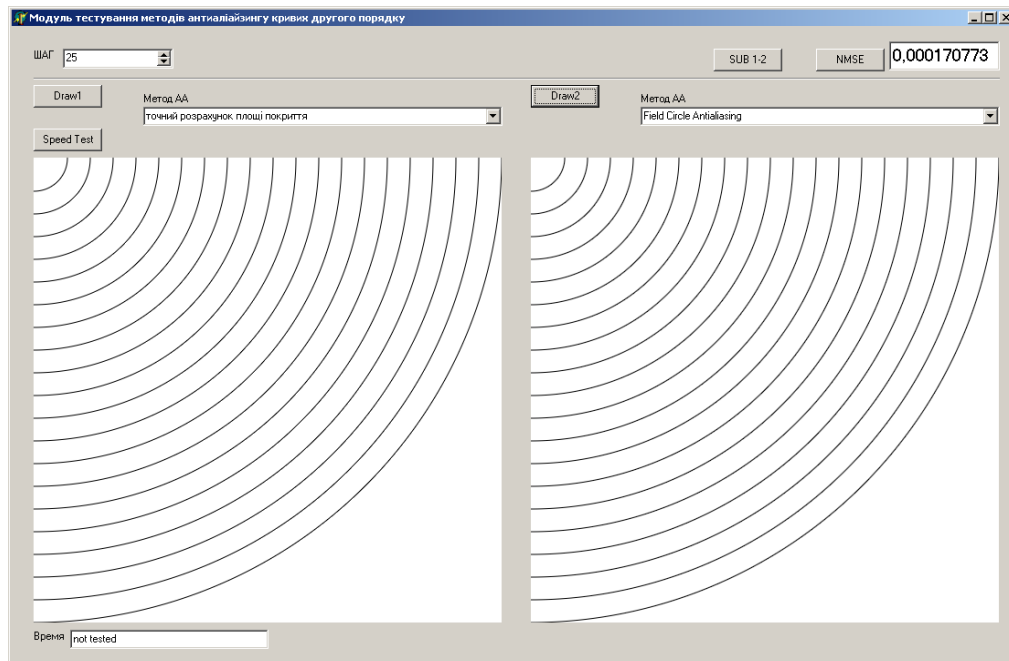
У розділі 4 були запропоновано методи антиаліазингу еліпсів і парабол, які відносять до найпоширеніших примітивів.

Методи передбачають виконання простих з обчислювальної точки зору операцій, а тому мають високу продуктивність і просту апаратну реалізацію.

4. ПРОГРАМНА РЕАЛІЗАЦІЯ АНТИАЛАЙЗИНГУ КРИВИХ ДРУГОГО ПОРЯДКУ

4.1 Засоби для антиаалайзингу зображень траєкторій кіл

У бакалаврській роботі було розроблено кілька методів антиалайзингзображення крокових траєкторій кіл. На рис. зображено зовнішній вигляд інтерфейса. Для формування дуг кіл виділено два поля. Одне з них використовується для формування дуг кіл згідно розроблених методів, а інше - для формування зображень з використанням базових методів. Відмінність між зображеннями визначається за нормованою середньоквадратичною похибкою.



4.1. Зовнішній вигляд інтерфейсу

Для тестування були обрані тестові фігури, які зображено на рис. 4.2 . Вони містять понад 100 тис. трикутників і є загальноприйнятими в комп'ютерній графіці.

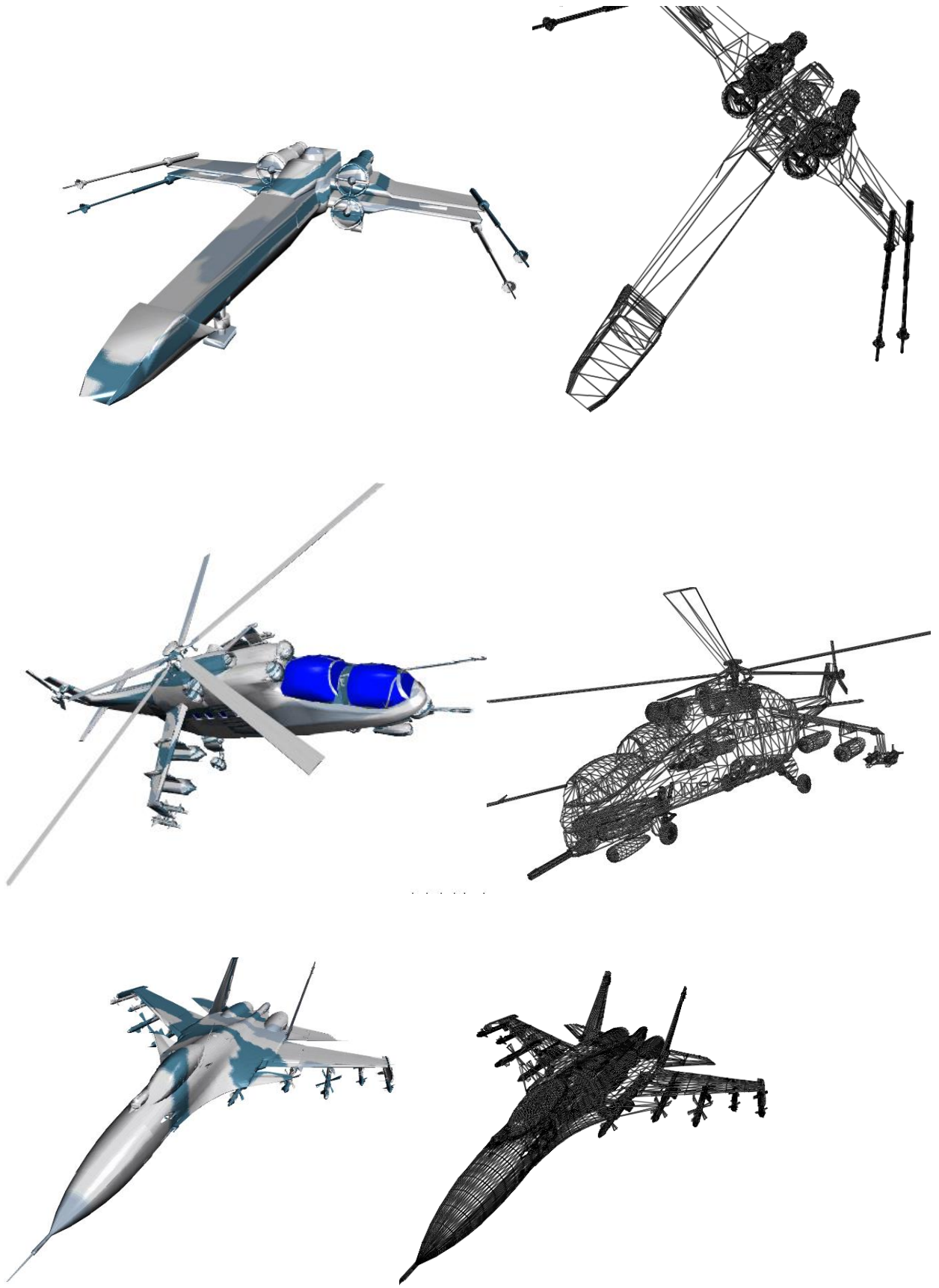


Рис. 4.2. Тестові об'єкти

Приклад формування траєкторій кіл наведено на рис. 4.3

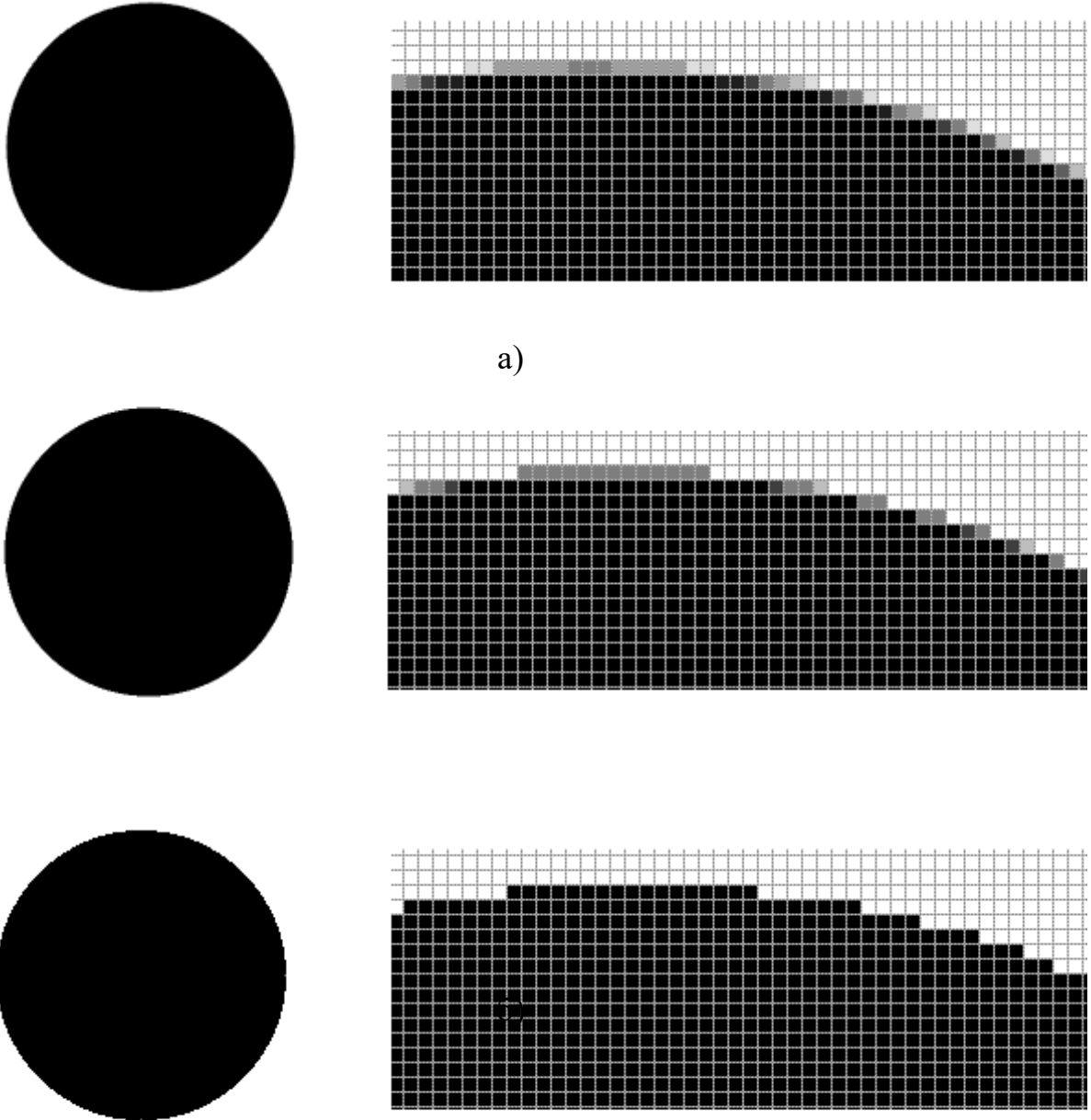


Рис 4.3 . Приклад антиаліазингу границь кіл:

- а) з використанням 8 додаткових ОФ;
- б) з використанням 4 додаткових ОФ;

У табл. 1 наведено результати тестування методів згладжування дуг кіл.

Таблиця 4. 1

Результати тестування методів згладжування дуг кіл

Метод згладжування	Час формування всіх можливих кіл в дискретному просторі 1024x1024, мсек	Коефіцієнт підвищення швидкодії порівняно з базовим методом
Розрахунок площі покриття пікселя за точною формулою	4 35 110	1
Табличний метод (Ву)	13 610	31,9
Філла	64 438	6,8
Модифікованої оцінювальної функції	32 218	13,5
Додаткових оцінювальних функцій (4 ОФ)	21 141	20,6
Додаткових оцінювальних функцій (8 ОФ)	25 341	17,2
Додаткових оцінювальних функцій (4 ОФ, адаптивний)	21 324	20,4
Табличний метод (мод.)	14 357	30,3

Тестування показало, що програмна реалізація табличного методу антиаліазингу границь кіл, розробленого у розділі 2, забезпечує приблизно однакову швидкодію з методом Ву .

Програмна реалізація методу антиаліазингу з використанням чотирьох додаткових оцінювальних функцій має у 1,5 рази більшу швидкодію порівняно з методом модифікованої оцінювальної функції.

4.2 Засоби для антиаліазингу зображень траєкторій еліпсів і гіпербол

Згідно з отриманими у розділі 3 виразами можна запропонувати алгоритм визначення координат точок траєкторії еліпса для області 1, граф-схему якого зображено на рис. 4.4.

На рис. 4.5 наведено приклади формування траєкторій еліпсів.

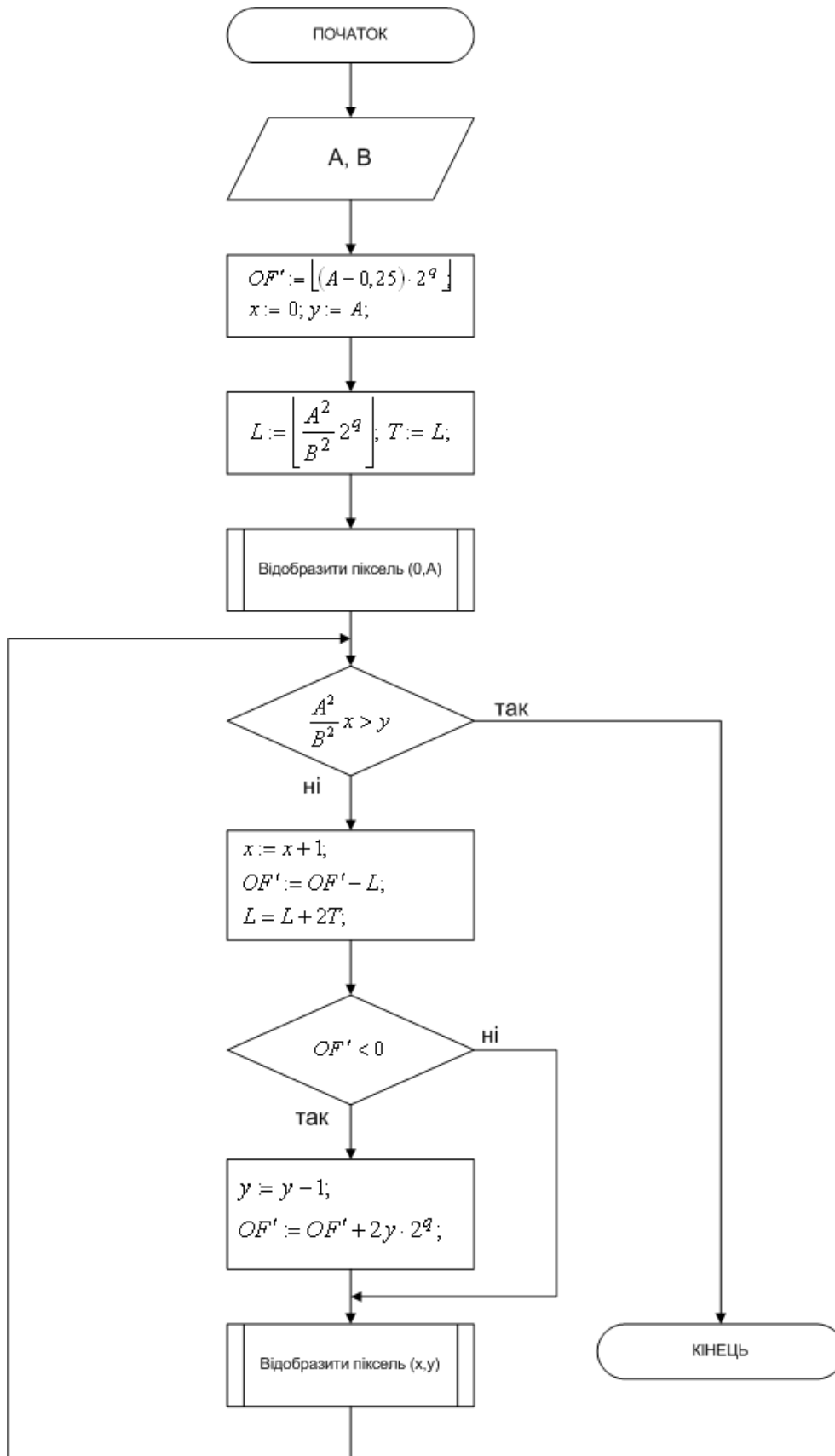


Рис. 4.4. Граф-схема алгоритму визначення координат точок траєкторії еліпса

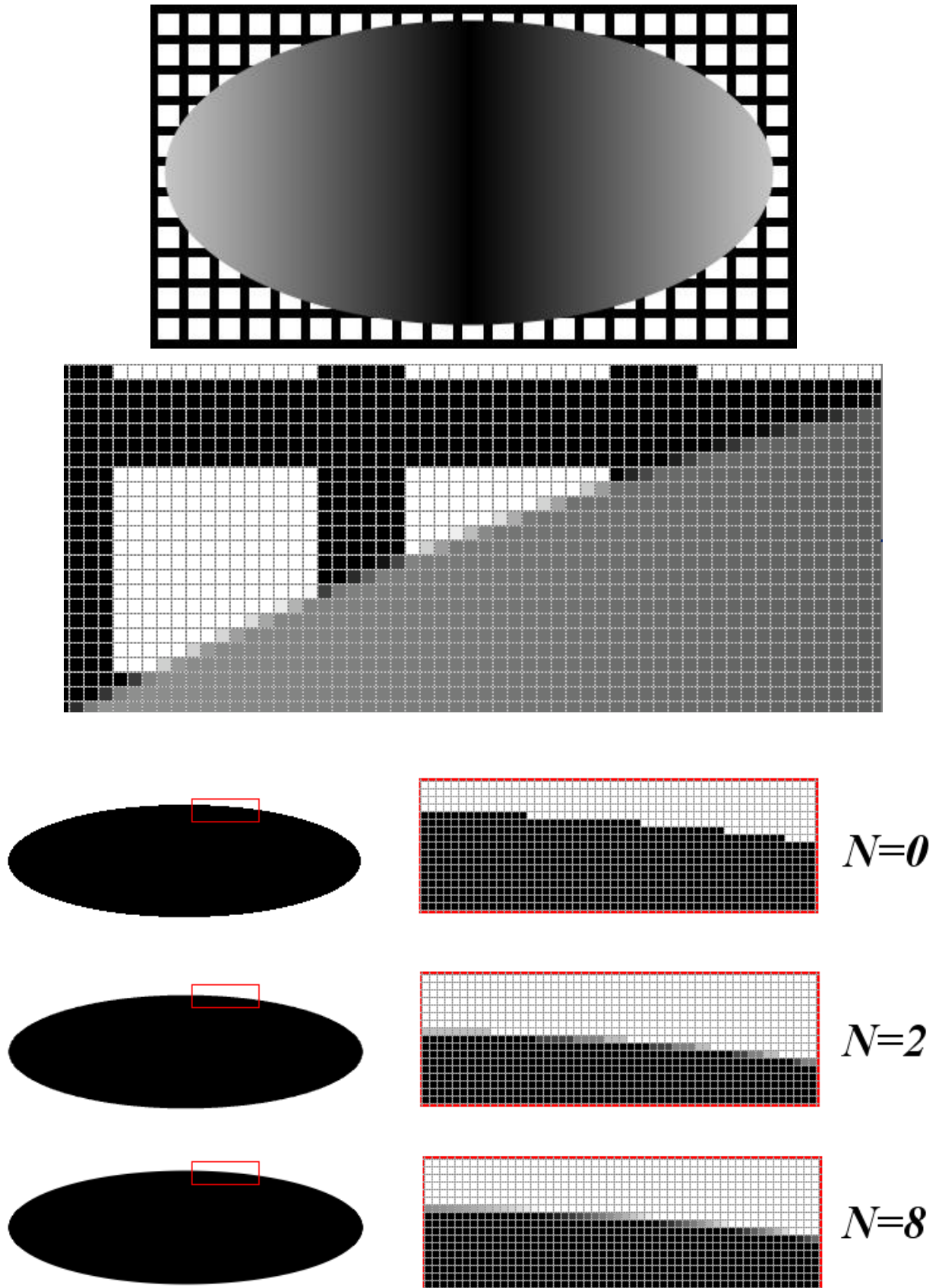


Рис. 4.5 Результат роботи алгоритму антиаліазингу границі еліпса для різних значень N

Згідно з отриманими в розділі 2 виразами можна запропонувати алгоритм визначення координат точок траєкторії гіперболи для області 1, граф-схему якого зображено на рис. 4.6. Аналогічно можна отримати алгоритм визначення координат точок траєкторії і для області 2. Запропонований алгоритм не містить “довгих” операцій, що забезпечує його високу швидкодію та простоту апаратної реалізації.

Приклади формування гіпербол наведено на рис. 4.7.

Висновки

Розглянуто особливості програмної реалізації розроблених методів антиліацзингу кривих другого порядку. Розроблено алгоритми і програми. Проведено тестування програмних засобів. Отримано порівняльні оцінки, які довели ефективність розроблених програмних засобів.

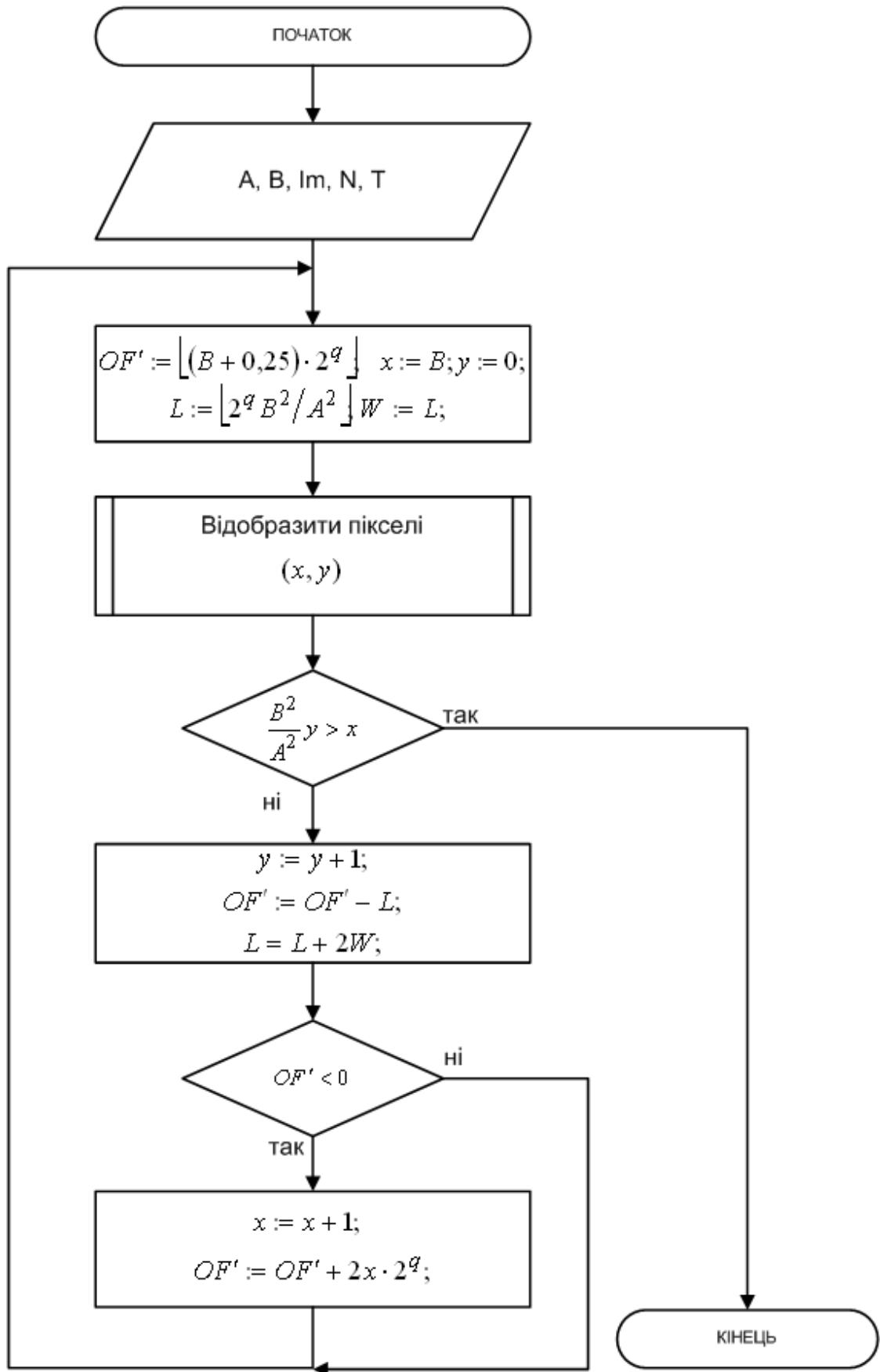


Рис. 4.6. Граф-схема алгоритму визначення координат точок траєкторії гіперболи

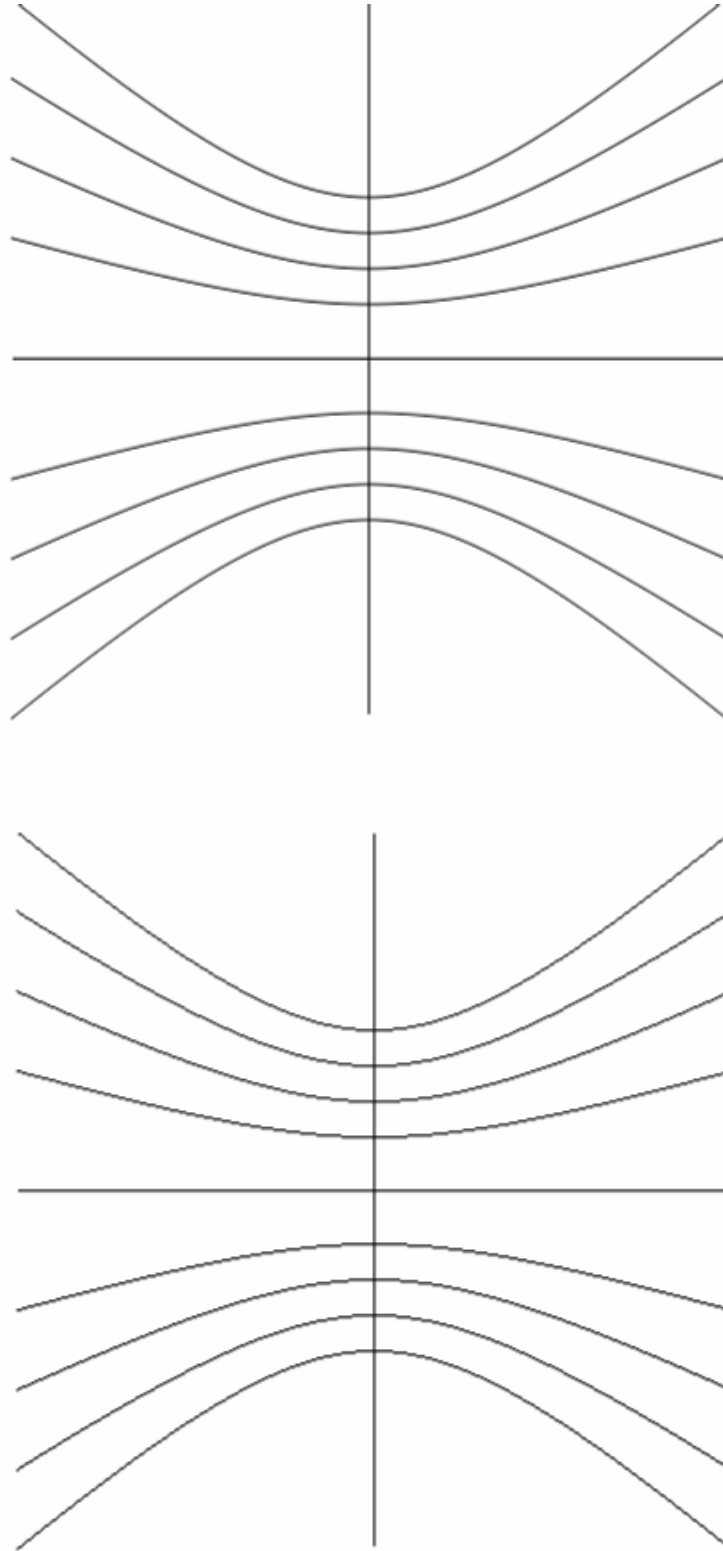


Рис. 4.7. Приклад формування траєкторій гіперболи

ВИСНОВКИ

Проведено аналіз існуючих методів антиаліайзингу. Обґрунтовано вибір методі оцінювальної функції як базового для розробки методів і засобів контурного антиаліайзингу

Вперше запропоновано метод аниаліайзингу зображень траєкторій кіл, особливість якого полягає у розбитті піксела на субпікселі з подальшим знаходженням для кожного з них знаків оцінювальних функцій, по яких розраховують площу покриття піксела траєкторією, що дозволяє підвищити продуктивність згладження;

Отримано нові аналітичні залежності для розрахунку площі покриття піксела траєкторією, що дозволило модифікувати табличний метод антиаліайзингу траєктрорій кіл, що дозволило зменшити об'єми постійної пам'яті,;

Подальшого розвитку отримали методи антиаліайзингу кривих другого порядку, в яких, на відміну від відомих, використано нові оцінювальні функції, що дозволило підвищити продуктивність згладження.

Розроблені алгоритми та програмні засоби для контурного антиаліайзингу зображень кривих другого порядку.

5. ЕКОНОМІЧНИЙ РОЗДІЛ

5.1 Технологічний аудит проведених досліджень та розроблених методів і засобів антиаліайзингу контурів графічних об'єктів

Як було зазначено у попередніх розділах виконаної магістерської кваліфікаційної роботи, бурхливий розвиток засобів обробки інформації привів до постійного зростання обсягу формування та передавання найрізноманітніших зображень як найбільш інформативного каналу комунікації між людиною та ЕОМ.

Щоб якісно розпізнавати такі зображення, при їх формуванні потрібно відображати графічні сцени з великим ступенем деталізації, тому особливу увагу потрібно приділяти не лише швидкодії формування графічних зображень, але і їх якості. При цьому у процесі синтезу зображень виникає багато проблем, одна з яких полягає у виникненні так званого ефекту *аліайзингу*, тобто появи яскраво виражених сходинок або зубців на краях графічних об'єктів, що суттєво впливає на якість та реалістичність сформованого зображення.

Як відомо, для забезпечення реалістичності синтезованих зображень у системах комп'ютерної графіки застосовують різноманітні спеціальні методи та засоби усунення ступінчастого ефекту, зокрема методи так званого антиаліайзингу, використання яких суттєво ускладнює обчислювальний процес формування зображень. А це, у свою чергу, об'єктивно обумовлює необхідність розробки нових, більш продуктивних методів, алгоритмів та структур проведення антиаліайзингу.

В попередніх розділах роботи нами було проаналізовано існуючі методи антиаліайзингу, встановлено переваги та недоліки кожного з них. В результаті було зроблено висновок, що у зв'язку з розвитком сучасних технологій виробництва великих інтегральних схем постійної пам'яті, стрімким розвитком виробництва графічних акселераторів, в яких широко використовуються методи формування реалістичних зображень, найбільш

прийнятими для розробки нових методів та алгоритмів антиаліайзингу є табличні методи реалізації функціональних залежностей. Пристрої, які базуються на використанні табличних методів перетворення, характеризуються високою швидкістю, яка обмежена тільки часом завершення перехідних процесів.

Тому перед нами буда поставлена мета – за рахунок удосконалення обчислювального процесу і застосування антиаліайзингу контурів графічних об'єктів у системах комп'ютерної графіки підвищити продуктивність процедури антиаліайзингу контурів графічних об'єктів та спростити її програмно-апаратну реалізацію.

Для реалізації поставленої мети нами було розроблено нові методи підвищення продуктивності антиаліайзингу векторів та кривих другого порядку за рахунок удосконалення обчислювального процесу, а також було розроблено алгоритми усунення аліайзингу контурів графічних об'єктів для програмно-апаратної реалізації їх у підсистемах растеризації високопродуктивних графічних систем.

Практичне значення одержаних нами результатів полягає у тому, що було розроблено алгоритми антиаліайзингу кіл та еліпсів з метою їх реалізації у графічних акселераторах; створено діючі програмні засоби для згладжування контурів графічних об'єктів у системах комп'ютерної графіки; запропоновано табличну реалізацію процедури розрахунку площі покриття піксела траєкторією відрізка прямої та кола, яка дозволила зменшити необхідні обсяги постійної пам'яті; розроблено спеціалізовані програмні засоби для тестування методів згладжування контурів графічних об'єктів, які дозволяють здійснювати порівняльні оцінки зображень за точністю і продуктивністю.

Для визначення технічного рівня та комерційного потенціалу проведених нами досліджень та розроблених методів і засобів антиаліайзингу контурів графічних об'єктів (в подальшому – нашої розробки) проведемо технологічний аудит отриманих результатів. Для проведення технологічного

аудиту були запрошені компетентні фахівці з цього питання: к.т.н., доцент Колесницький О.К., к.т.н., доцент Майданюк В.П. і к.т.н., доцент Сілагін О.В.

Запрошені до проведення технологічного аудиту експерти здійснювали оцінювання технічного рівня та комерційного потенціалу нашої розробки за методикою та рекомендаціями, які наведено в таблиці 4.1, а встановлення технічного рівня та комерційного потенціалу нашої розробки здійснювалося за рекомендаціями, які наведено в таблиці 5.2 [20]:

Таблиця 5.1 – Критерії оцінювання технічного рівня та комерційного потенціалу наукової розробки

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	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
Ринкові перспективи					
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
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	Конкуренція 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 збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї

Продовження таблиці 5.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
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
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка	Необхідно отримання	Процедура отримання	Необхідно тільки пові-	Відсутні будь-які

регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	домлення відповідним органам про виробництво та реалізацію продукту	регламентн і обмеження на виробництво та реалізацію продукту
--	--	---	---	--

Таблиця 5.2 – Потенційні технічні рівні та комерційні потенціали наукової розробки

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

Провівши технологічний аудит нашої розробки, запрошені експерти виставили оцінки, які зведено в таблицю 5.3.

Таблиця 5.3 – Результати технологічного аудиту нашої розробки

Критерії	Прізвище, ініціали експерта		
	Колесницький О.К	Майданюк В.П.	Сілагін О.В.
	Бали, виставлені експертами:		
1	4	4	4
2	4	4	4

3	4	3	3
4	3	3	3
5	4	3	3
6	4	4	4
7	4	4	4
8	4	4	3
9	3	3	3
10	4	4	3
11	3	3	3
12	3	3	3
Сума балів	СБ ₁ =44	СБ ₂ =42	СБ ₃ =40
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{3} = \frac{44 + 42 + 40}{3} = \frac{126}{3} = 42,00.$		

Результати технологічного аудиту та розрахунків, які наведено в таблиці 5.3, показують, що середньоарифметична сума балів, що їх виставили експерти, дорівнює 42-м балам.

Керуючись рекомендаціями, наведеними в таблиці 5.3, можна зробити висновок, що результати проведених нами досліджень та розроблених методів і засобів антиаліаїзингу контурів графічних об'єктів мають технічний рівень та комерційний потенціал, який вважається «високим».

Таким високий рівень нашої розробки пояснюється тим, що нами вперше розроблено метод антиаліаїзингу контурів із використанням додаткових оцінювальних функцій для обчислення площі покриття пікселя контуром графічного об'єкта, в якому антиаліаїзинг виконується безпосередньо під час формування крокової траєкторії і не потребує збільшення розмірів кадрового буфера та виконання фільтрації зображення на етапі постоброблення, що дозволяє спростити реалізацію процедури антиаліаїзингу кривих другого порядку у спеціалізованих графічних процесорах. Окрім того подальшого розвитку

отримав табличний метод антиаліайзингу меж кіл, в якому використано нові аналітичні співвідношення для розрахунку площі покриття піксела траєкторією, що дозволило збільшити крок квантування і таким чином у 2÷8 разів зменшити обсяги постійної пам'яті, які необхідні для табличної реалізації процедури антиаліайзингу у спеціалізованих графічних процесорах;

5.2 Розрахунок витрат на проведення досліджень та розробку методів і засобів антиаліайзингу контурів графічних об'єктів

При виконанні цієї роботи були використані такі статті витрат [21]:

5.2.1. Основна заробітна плата Z_o розробників (формула 5.1):

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

де M – середньомісячна заробітна плата конкретного розробника; U ВНТУ середньомісячна заробітна плата конкретного розробника коливаються в межах:

$M = (4723 \dots 20000) \text{ грн/міс.};$

T_p – число робочих днів в місяці; приймемо $T_p = 22$ дні;

t – число робочих днів роботи розробників.

Розрахунки основної заробітної плати розробників зведені в таблицю 4.4:

Таблиця 5.4 – Основна заробітна плата розробників (округлено до цілих чисел)

Найменування посади виконавця	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів (годин) роботи	Витрати на оплату праці, грн	Примітка
1. Науковий керівник магістерської кваліфікаційної роботи	20000	909,09 грн в день	25 годин	3788	При 36-годинній робочій неділі
2. Розробник- магістрант	2500	113,64	70 днів	7955	
3. Консультант	15000	681,82	2,5 години	285	При 36-годинній робочій неділі

4. Інші фахівці	10000	454,55	3	1364	
Всього				$Z_o = 13392$	грн

5.2.2. Додаткова заробітна плата Z_d розробників та робітників (формула 5.2):

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

Для нашого випадку отримаємо:

$$Z_d = 0,11 \times 13392 \approx 1473 \text{ грн.}$$

5.2.3. Нарахування на заробітну плату $Н_{зп}$ розробників розраховуються за формулою (4.3):

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

де $\beta = 22\%$ – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування.

Для нашого випадку отримаємо:

$$Н_{зп} = (13392 + 1473) \times 0,22 \approx 3270 \text{ грн.}$$

5.2.4. Амортизація A основних засобів, обладнання, комп'ютерів тощо, яка розраховується за формулою (4.4) :

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

де $Ц$ – загальна балансова вартість основних засобів, обладнання, комп'ютерів

тощо, які використовувалися під час виконання роботи, грн;

H_a – річна норма амортизаційних відрахувань. Спрощено можна прийняти, що

$$H_a = (5...25)\%;$$

T – термін, використання кожного виду основних засобів, місяці.

Розрахунки амортизаційних відрахувань наведено в таблиці 5.5.

Таблиця 5.5 – Розрахунок амортизаційних відрахувань (округлено)

Найменування обладнання, приміщень тощо	Балансова вартість, грн.	Норма амортизації і, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн.
1. Обладнання: комп'ютери, принтери тощо	38000	20	2,5 (35% використання)	554
2. Приміщення кафедри та університету	42000	2,5	2,5 (35% використання)	77
Всього				A = 631 грн

5.2.5. Витрати на матеріали M (формула 5.5):

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

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

При виконанні роботи загальна вартість всіх матеріалів становила 1000 грн.

5.2.6. Витрати на комплектуючі K (формула 5.6):

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

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

Загальна вартість комплектуючих, які були використані під час виконання цієї роботи, становила приблизно 300 грн.

5.2.7. Витрати на силову електроенергію V_e (формула 5.7):

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

де B – вартість 1 кВт-год. електроенергії, в 2020 р. $B \approx 2,5$ грн/кВт;

Π – установлена потужність обладнання, кВт; $\Pi = 1,45$ кВт;

Φ – фактична кількість годин роботи обладнання, годин; $\Phi = 196$ годин;

K_{Π} – коефіцієнт використання потужності; $K_{\Pi} < 1 = 0,87$.

$K_{д}$ – коефіцієнт корисної дії, $K_{д} = 0,76$.

Тоді витрати на силову електроенергію приблизно становитимуть:

$$B_e = \frac{B \cdot \Pi \cdot \Phi \cdot K_{\Pi}}{K_{д}} = \frac{2,5 \cdot 1,45 \cdot 196 \cdot 0,87}{0,76} \approx 813 \text{ грн.}$$

5.2.8. Інші витрати $B_{ін}$ (опалення, освітлення, ремонт, утримання приміщень тощо) розраховуються як (100...300)% від основної заробітної плати розробників, тобто:

$$B_{ін} = (1 \dots 3) \times Z_o. \quad (5.8)$$

Для нашого випадку отримаємо:

$$B_{ін} = 1,0 \times 13392 = 13392 \text{ грн.}$$

5.2.9. Сума всіх попередніх статей дає витрати на виконання роботи безпосередньо магістрантом – $B_{заг}$.

$$B_{заг} = 13392 + 1473 + 3270 + 631 + 1000 + 300 + 813 + 13392 = 34271 \text{ грн.}$$

5.2.10. Загальні витрати ZB на остаточне завершення роботи та оформлення їх результатів розраховуються за формулою (5.9):

$$ZB = \frac{B_{заг}}{\beta}, \quad (5.9)$$

де β – коефіцієнт, який характеризує етап виконання роботи на шляху до її можливого впровадження.

Загальні рекомендації: якщо розробка знаходиться: на стадії науково-дослідних робіт, то $\beta \approx 0,1$; на стадії технічного проектування, то $\beta \approx 0,2$; на стадії розробки конструкторської документації, то $\beta \approx 0,3$; на стадії розробки технологій, то $\beta \approx 0,4$; на стадії розробки дослідного зразка, то $\beta \approx 0,5$; на стадії розробки промислового зразка, $\beta \approx 0,7$; на стадії впровадження, то $\beta \approx (0,8 - 0,9)$.

Для нашого випадку доцільно прийняти, що $\beta \approx 0,7$.

Тоді: $ZB = \frac{34271}{0,7} = 48958,57$ грн або приблизно 50 тисяч грн.

Тобто загальні витрати на остаточне завершення роботи та оформлення її результатів становлять приблизно 50 тис. грн.

5.3 Розрахунок економічного ефекту від можливої комерціалізації нашої розробки

Економічний ефект від можливої комерціалізації нашої розробки можливий за рахунок її значно кращих функціональних можливостей та характеристик. Зокрема, перевагами розроблених нами програм є підвищення продуктивності та реалістичності формування графічних зображень у системах комп'ютерної графіки за рахунок згладження траєкторій кривих другого порядку. Окрім цього розроблені нами програми можна інтегрувати в існуючі пакети прикладних програм, наприклад, в Photoshop, 3Dmax, які мають масове поширення.

Тому, якщо існуючі подібні програми та засоби коштують на ринку в середньому приблизно 2500 грн, то нашу розробку, яка має значно кращі функціональні характеристики, можна буде реалізовувати на ринку дещо дорожче, наприклад, приблизно за \$100, тобто приблизно за 3000 грн, чи на 500 грн дорожче.

Аналіз місткості ринку аналогічної продукції показує, що наразі в Україні кількість охочих подібну розробку складає щороку приблизно 2000 осіб, але їх кількість буде стрімко зростати. Оскільки наша розробка має значно кращі функціональні можливості та характеристики, то вона може користуватися підвищеним попитом на ринку протягом 3-х років після впровадження.

Тобто наша розробка може бути впроваджена з 1 січня 2021 року (оскільки потребує певного доопрацювання), а її результати будуть виявлятися протягом 2021-го, 2022-го та 2023-го років.

Прогноз зростання попиту на нашу розробку складає по роках:

- 2021 р. – приблизно на $\Delta 250$ шт.;[21]:
- 2022 р. – приблизно на $\Delta 350$ шт.;
- 2023 р. – приблизно на $\Delta 500$ шт.

Розрахуємо можливе збільшення чистого прибутку $\Delta\Pi_i$, що його можна отримати потенційний інвестор від впровадження нашої розробки [21]:

$$\Delta\Pi_i = \sum_1^n (\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\nu}{100}\right),$$

(5.10)

де ΔC_o – зміна основного якісного показника від впровадження результатів розробки у цьому році. Таким показником є зміна ціни нової розробки; для нашого випадку це буде: $\Delta C_o = (3000 - 2500) = +500$ грн;

N – основний кількісний показник, який визначає обсяг діяльності у цьому році до впровадження результатів розробки; $N = 2000$ шт.;

ΔN – покращення основного кількісного показника від впровадження результатів нашої розробки. Таке покращення відповідно по роках становитиме:

$$\Delta_{21} = +250, \Delta_{22} = +350 \text{ та } \Delta_{23} = +500 \text{ шт.};$$

C_o – основний якісний показник, який визначає обсяг діяльності у році після впровадження результатів розробки; для нашого випадку $C_o = 3000$ грн;

n – кількість років, протягом яких очікується отримання позитивних результатів від можливого 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 $\rho = (0,2 \dots 0,5)$; візьмемо $\rho = 0,5$;

ν – ставка податку на прибуток. У 2020 році $\nu = 18\%$.

Величина чистого прибутку $\Delta\Pi_1$ для потенційного інвестора протягом першого року від можливого впровадження нашої розробки (2021 р.) складе:

$$\Delta\Pi_1 = [500 \cdot 2000 + 3000 \cdot 250] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) \approx 597893 \text{ грн.}$$

Величина чистого прибутку $\Delta\Pi_2$ для потенційного інвестора від можливого впровадження нашої розробки протягом другого (2022 р.) року складе:

$$\Delta\Pi_2 = [500 \cdot 2000 + 3000 \cdot 350] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) \approx 700389 \text{ грн.}$$

Величина чистого прибутку $\Delta\Pi_3$ для потенційного інвестора від можливого впровадження нашої розробки протягом третього (2023 р.) року складе:

$$\Delta\Pi_3 = [500 \cdot 2000 + 3000 \cdot 500] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) \approx 854132 \text{ грн.}$$

Приведена вартість всіх можливих чистих прибутків ПП, що їх може отримати потенційний інвестор, розраховується за формулою (в цінах на 01.01.2020 року):

$$ПП = \sum_1^t \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.11)$$

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 (2021, 2022 та 2023 роки);

τ – ставка дисконтування; приймемо за ставку дисконтування прогнозований рівень інфляції в Україні: $\tau = 0,04$ (4%);

t – період часу від моменту здійснення тих чи інших платежів (отримання прибутків та вкладення інвестицій) до моменту початку розробки. Моментом початку розробки будемо вважати 01.03.2020 року.

Тоді приведена вартість (в цінах на 01.01.2020 року) всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження нашої розробки, складе:

$$\begin{aligned} \text{ПП} &= \frac{597893}{(1+0,04)^2} + \frac{700389}{(1+0,04)^3} + \frac{854132}{(1+0,04)^4} \approx 552786 + 622643 + 730116 = \\ &= 1905545 \text{ грн або приблизно } 1906 \text{ тисяч грн.} \end{aligned}$$

Далі розрахуємо початкову теперішню вартість інвестицій PV, що можуть бути вкладені інвестором у випадку реалізації нашої розробки:

$$PV = K \times 3B = (2\dots5) \times 3B, \quad (5.12)$$

де 3B – витрати на розробку; 3B = 50 тис. грн (див. формулу 4.9);

K = (2...5) – додаткові витрати потенційного інвестора на реалізацію нашої розробки. Прийmemo, що K = 4.

Тоді для нашого випадку отримаємо:

$$PV = (2\dots5) \times 50 = 4 \times 50 = 200 \text{ тис. грн.}$$

Тоді абсолютний економічний ефект для потенційного інвестора від можливих вкладених інвестицій $E_{\text{абс}}$ може становити:

$$E_{\text{абс}} = \text{ПП} - PV, \quad (5.13)$$

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

PV – теперішня вартість інвестицій PV = 200 тис. грн.

Для нашого випадку:

$E_{\text{абс}} = 1906 - 200 = 1706$ тис. грн або приблизно по 568,67 тис. грн щорічно протягом 3-х років.

Внутрішня дохідність $E_{\text{в}}$ інвестицій, вкладених у комерціалізацію нашої розробки, розраховується за формулою (5.14):

$$E_{\text{в}} = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1, \quad (5.14)$$

де $E_{\text{абс}}$ – абсолютний ефект вкладених інвестицій; $E_{\text{абс}} = 1706$ тис. грн;

PV – теперішня вартість початкових інвестицій PV = 200 тис. грн;

$T_{ж}$ – життєвий цикл розробки, роки. $T_{ж} = 4$ (від 1.01.2020 до 31.12.2023 року).

Для нашого випадку отримаємо:

$$E_{в} = \sqrt[4]{1 + \frac{1706}{200}} - 1 = \sqrt[4]{1 + 8,53} - 1 = \sqrt[4]{9,53} - 1 = 1,757 - 1 = 0,757 = 75,7\%.$$

Далі визначимо ту мінімальну дохідність, нижче за яку потенційний інвестор не буде зацікавлений займатися комерціалізацією нашої розробки.

Мінімальна дохідність або мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою (5.15):

$$\tau = d + f, \quad (5.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2020 році в Україні $d = (0,10...0,15)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина

$f = (0,05...0,5)$, але може бути і значно більше.

Для нашого випадку отримаємо:

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

Оскільки величина $E_{в} = 75,7\% > \tau_{\text{мін}} = 65\%$, то потенційний інвестор може бути зацікавлений у комерційному впровадженні нашої розробки.

Далі розраховуємо термін окупності коштів, вкладених у нашу розробку.

Термін окупності коштів $T_{\text{ок}}$ можна розрахувати за формулою (5.16):

$$T_{\text{ок}} = \frac{1}{E_{в}}. \quad (5.16)$$

Термін окупності $T_{\text{ок}}$ коштів, вкладених у нашу розробку, може становити:

$$T_{\text{ок}} = \frac{1}{0,757} \approx 1,32 \text{ роки,}$$

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 проведення досліджень та розробку методів і засобів антиаліайзингу контурів графічних об'єктів	Не більше 60 тис. грн	50 тис. грн.	Виконано
2. Абсолютний щорічний ефект від можливого впровадження розробки, тис. грн	не менше 500 тис. грн за рік	568,67 тис. грн щорічно протягом 3-х років.	Виконано
3. Внутрішня дохідність вкладених інвестицій, %	не менше 65%	75,7%	Досягнуто
4. Термін окупності, роки	до 3-х років	1,32 роки	Виконано

Таким чином, основні техніко-економічні показники проведених досліджень та розроблених методів і засобів антиаліайзингу контурів графічних об'єктів, визначені у технічному завданні, виконані.

ЛІТЕРАТУРА

1. Crow F. *A Comparison of Antialiasing Techniques* // IEEE CG & A.– 1981. – Vol. 1. – P.40–47.
2. Leler W.J. *Human Vision, Anti-aliasing, and the cheap 4000 Line Display* // ACM. – 1980. – Vol. 14. – P.308–313.
3. Leler W.J. *Human Vision, Anti-aliasing, and the cheap 4000 Line Display* // ACM. – 1980. – Vol. 14. – P.308–313.
4. Grossman J. *Point Sample Rendering* // Master's thesis, Department of Electrical Engineering and Computer Science: MIT. - August 1998.
5. Crow F. *A Comparison of Antialiasing Techniques* // IEEE CG & A.– 1981. – Vol. 1. – P.40–47.
6. Crow F. *The aliasing problem in Computer Generated Shaded Images* // SACM. – 1977. – Vol.20. – P.799-805.
7. Фоли Дж., Ван Дем А. *Основы интерактивной машинной графики*. В 2-х книгах. М.: Мир, 1985. - 384 с.
8. Mitchell P. *Generating Antialiased Images at Low Sampling Densities* // Proceedings of SIGGRAPH 87. – 1987. Vol. 21. P.65–72.
9. Dippé M.A.Z., Wold, E.H. *Antialiasing through stochastic sampling* // Computer Graphics (Proceedings of the ACM SIGGRAPH '85 Conference). – 1985. Vol. 19. – No. 3. – P.69-78.
10. Haeberli P., Akeley K. *The accumulation buffer: hardware support for high-quality rendering* // ACM SIGGRAPH Computer Graphics. – 1990. – Vol. 24. – No.4. – P.309-318.
11. Романюк О. Н., Курінний М. С.. *Методи та засоби антиаліаїзінгу контурів об'єктів у системах комп'ютерної графіки*. Монографія / О. Н. Романюк, –Вінниця: УНІВЕСУМ-Вінниця –2006. –163 с.
12. 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.
13. Романюк О. Н. *Комп'ютерна графіка*. Навчальний посібник.

—Вінниця: УНІВЕСУМ-Вінниця—2001. —129 с.

14. Херн Д., Бейкер М. Компьютерная графика и стандарт OpenGL / Д. Херн , М. Бейкер. — М.: Издательский дом "Вильямс", 2005. — 1168 с.

15. Хилл Ф. OpenGL. Программирование компьютерной графики / Ф. Хилл — СПб: Питер, 2002. —1088 с.

16. Цисарж В. В. Математические методы компьютерной графики / В. В Цисарж., Р. И. Марусин. — К.: Факт — 2004. — 464 с.

17. Лященко М. Я., Головань М. С. *Чисельні методи*: Підручник. – К.: "Либідь", 1996. – 288 с

18. Бурян А.С., Романюк О.Н., Методи антиаліайзингу для границь кіл та еліпсів. Матеріали XX Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів “Стан, досягнення і перспективи інформаційних систем і технологій”, Одеса, 2020. – с. 36-39.

19. Романюк О.Н., Бурян А.С., Курінний М.С. Метод зглядження зображення траєктої кола. Матеріали IX Міжнародної науково-практичної конференції молодих вчених та студентів «Молодь у світі сучасних технологій», МССТ-2020, Херсон, 2020.

20. 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.).

21. Козловський В. О. Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт. – Вінниця: ВНТУ, 2012.

ДОДАТКИ

Додаток А

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

ЗАТВЕРДЖУЮ

д.т.н., проф. Романюк О.Н.

“30” жовтня 2019 року

Технічне завдання
на магістерську кваліфікаційну роботу за спеціальністю
121 – Інженерія програмного забезпечення

Керівник магістерської кваліфікаційної роботи:

_____ д.т.н., проф. О.Н.Романюк

" ____ " _____ 2019 р.

Виконав:

_____ студент ПІ –18м А.С. Бур'ян

" ____ " _____ 2019 р.

Вінниця – 2019 року

1. Найменування та галузь застосування

Магістерська кваліфікаційна робота: «Розробка методів і засобів антиаліайзингу зображень кривих другого порядку».

Галузь застосування - системи комп'ютерної графіки.

2. Підстава для розробки.

Підставою для розробки магістерської кваліфікаційної роботи є рішення засідання кафедри програмного забезпечення (протокол №__ від «__»_____ 2020 року).

3. Мета та призначення розробки.

Метою роботи є підвищення продуктивності антиаліайзингу кривих другого порядку за рахунок розробки та використання нових методів і засобів. Призначення роботи – розробка методів і засобів моделювання світлових ефектів та графічного 3D редактора.

4. Джерела розробки:

- завдання на магістерську кваліфікаційну роботу.
- Романюк О. Н., Курінний М. С.. Методи та засоби антиаліайзингу контурів об'єктів у системах комп'ютерної графіки. Монографія / О. Н. Романюк, –Вінниця: УНІВЕСУМ-Вінниця –2006. –163 с.
- Crow F. A Comparison of Antialiasing Techniques // IEEE CG & A.– 1998. – Vol. 1. – P.40–47.

5. Технічні вимоги

тип антиаліайзингу – контурний;

графічний режим – TrueColor;

метод формування крокової траєкторії – метод оцінювальної функції;

типи кривих – коло, еліпс, гіпербола.

6. Конструктивні вимоги.

Засоби антиаліайзингу реалізувати програмно з використанням графічної відеокарти

Графічна та текстова документація повинна відповідати діючим стандартам України.

7. 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	Примітка
1	Аналіз методів і засобів антиаліайзингу графічних об'єктів	10.01.2020	Вик.
2	Розробка високопродуктивних методів антиаліайзингу для траєкторій кіл	5.02.2020	Вик.
3	Розробка методів антиаліайзингу меж еліпсів та гіпербол	5.03.2020	Вик.
4	Програмна реалізація антиаліайзингу кривих другого порядку	15.04.2020	Вик.
5	Економічна частина	27.05.2020	Вик.

10. Порядок контролю та прийняття.

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

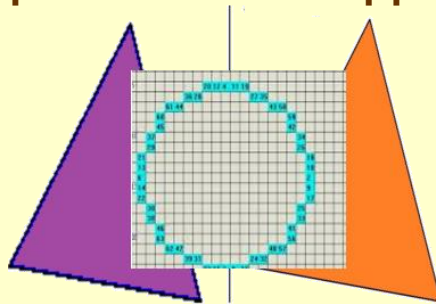
Прийняття магістерської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком захисту.

**ІЛЮСТРАТИВНИЙ МАТЕРІАЛ
ДО ЗАХИСТУ БАКАЛАВРСЬКОЇ ДИПЛОМНОЇ РОБОТИ**

					08-22.БДР 004.16.000.СК			
					Ілюстративний матеріал	<i>Літ.</i>	<i>Маса</i>	<i>Масштаб</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				1 : 1
<i>Розроб.</i>		<i>Бур'ян А.С</i>						
<i>Перевір.</i>		<i>Романюк О.Н.</i>						
<i>Т. Контр.</i>						<i>Арк.</i>	6	<i>Аркушів</i> 6
<i>Реценз.</i>		<i>Колесницький О.К.</i>			<u>Розробка методів і засобів антиаліаїзингу зображень кривих другого порядку</u>		ІІІ –18м	
<i>Н. Контр.</i>		<i>Романюк О.Н</i>						
<i>Затверд.</i>		<i>Романюк О.Н.</i>						

Міністерство освіти та науки України
Вінницький національний технічний університет

РОЗРОБКА МЕТОДІВ І ЗАСОБІВ АНТИАЛІАЙЗИНГУ ЗОБРАЖЕНЬ КРИВИХ ДРУГОГО ПОРЯДКУ



Виконав: студент II курсу, групи ПІ –18м

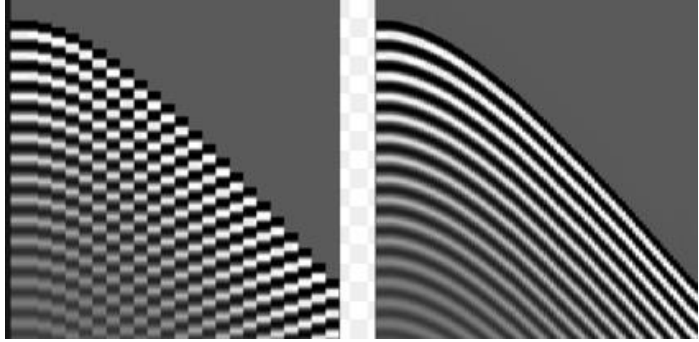
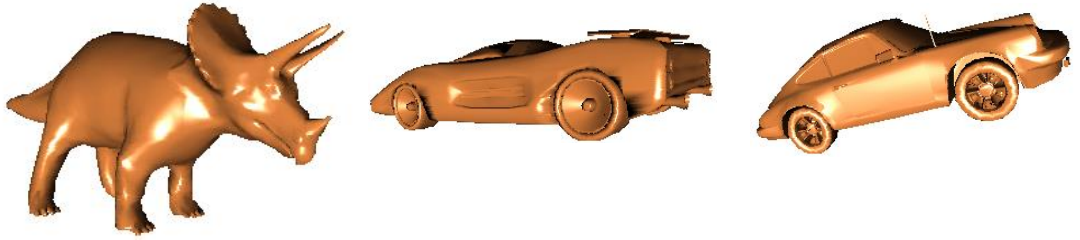
Бур'ян А.С.

Керівник д.т.н., проф.. Романюк О.Н.

1

Вінниця 2020

Ефект аліайзінгу



Мета і завдання дослідження. Метою роботи є підвищення продуктивності антиалайзингу кривих другого порядку за рахунок розробки та використання нових методів і засобів

Основними задачами дослідження є:

- аналіз методів і засобів антиалайзингу контурів;
- спрощення аналітичних залежностей для задач антиалайзингу;
- розробка методів підвищення продуктивності згладжування траєкторій кіл, еліпсів і гіпербол;
- розробка алгоритмів і програм усунення алаізингу контурів кривих другого порядку;
- комп'ютерне моделювання для перевірки отриманих результатів.

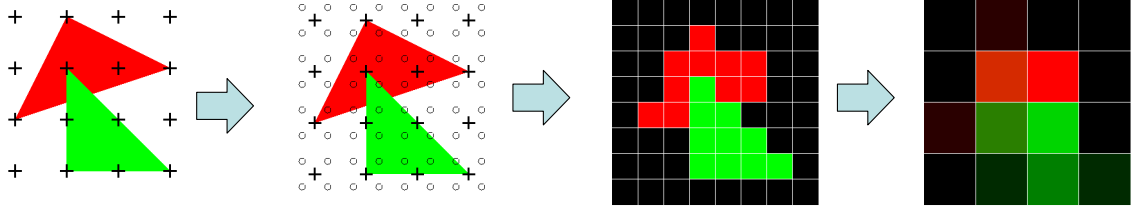
Об'єкт дослідження – процес антиалайзингу зображень кривих другого порядку

Предмет дослідження – методи і засоби антиалайзингу зображень кривих другого порядку.

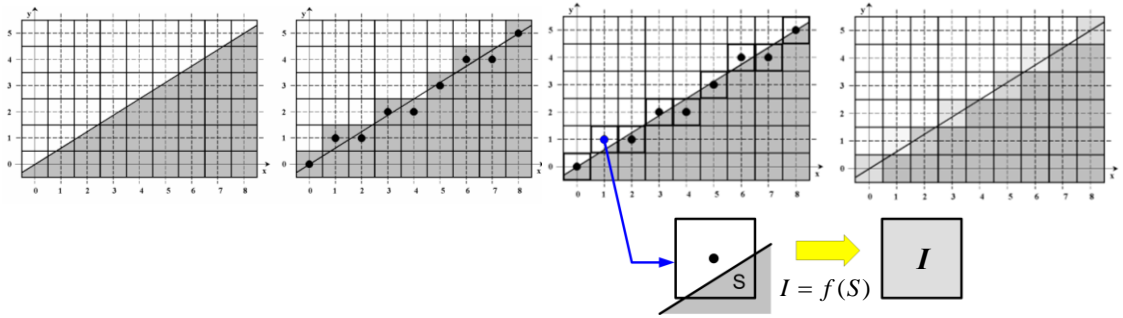
Методи дослідження У процесі дослідження застосовувалася: теорія чисел та чисельних методів, математичний апарат апроксимації функцій; методи аналітичної геометрії для розробки методів згладження зображень кривих другого порядку; теорія алгоритмів і програмного забезпечення для розробки програм; комп'ютерне моделювання для перевірки отриманих результатів.

Методи антиаліазингу

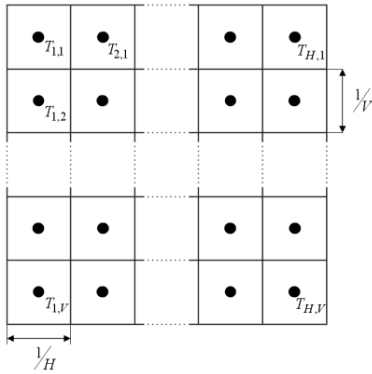
Метод додаткової вибірки



Метод крайового антиаліазингу



Метод додаткових оцінювальних функцій



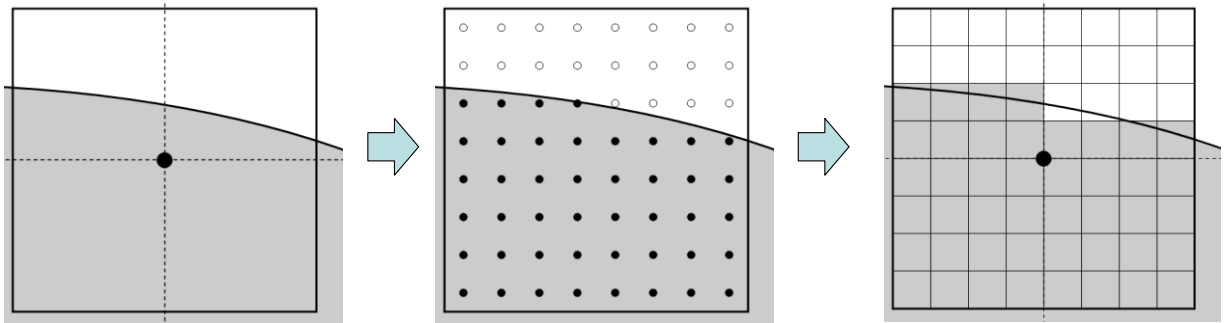
$$S_{покр.} \approx S_a = \frac{\sum_{i=1}^H \sum_{j=1}^V P_{i,j}}{H \cdot V}$$

Для точок, які покриваються графічним примітивом, $P_{i,j} = 1$.
 В усіх інших випадках $P_{i,j} = 0$.

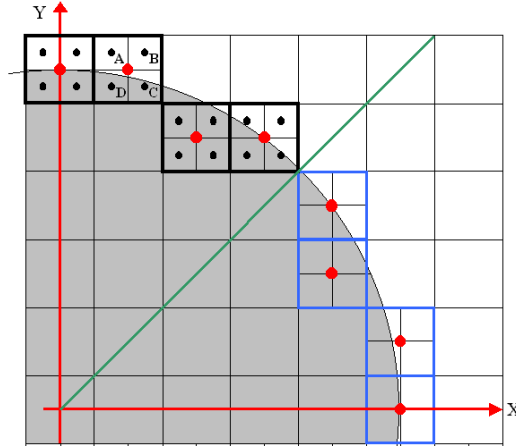
$$\max(\Delta S) = \frac{(V + H)^2}{8V^2H}$$

Для випадку, коли $H = V = C$: $\max(\Delta S) = \frac{1}{2C}$.

Розташування допоміжних точок всередині пікселя

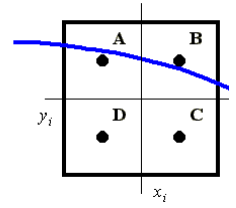


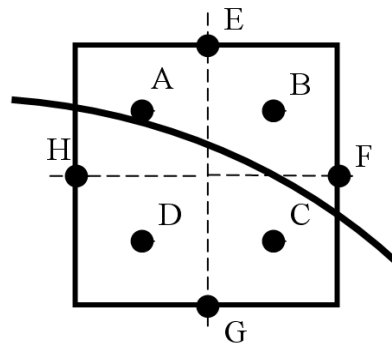
Формування оцінювальної функції



Визначення оцінювальної функції в допоміжних точках

$$O\Phi_i^A = O\Phi_i - \frac{x_i}{2} + \frac{y_i}{2} + \frac{1}{8} \approx O\Phi_i - \left(\frac{x_i - y_i}{2}\right); \quad O\Phi_i^C = O\Phi_i + \frac{x_i}{2} - \frac{y_i}{2} + \frac{1}{8} \approx O\Phi_i + \left(\frac{x_i - y_i}{2}\right);$$

$$O\Phi_i^B = O\Phi_i + \frac{x_i}{2} + \frac{y_i}{2} + \frac{1}{8} \approx O\Phi_i + \left(\frac{x_i + y_i}{2}\right); \quad O\Phi_i^D = O\Phi_i - \frac{x_i}{2} - \frac{y_i}{2} + \frac{1}{8} \approx O\Phi_i - \left(\frac{x_i + y_i}{2}\right).$$




Визначення оцінювальної функції в допоміжних точках

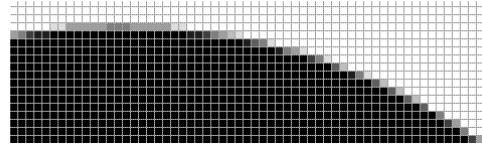
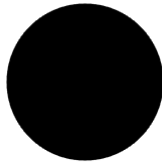
$$O\Phi_i^E = x_i^2 + (y_i + 1/2)^2 - R^2 = (x_i^2 + y_i^2) - R^2 + y_i + \frac{1}{4} = O\Phi_i + y_i + \frac{1}{4};$$

$$O\Phi_i^F = (x_i + 1/2)^2 + y_i^2 - R^2 = (x_i^2 + y_i^2) - R^2 + x_i + \frac{1}{4} = O\Phi_i + x_i + \frac{1}{4};$$

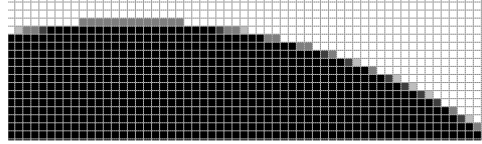
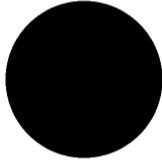
$$O\Phi_i^G = x_i^2 + (y_i - 1/2)^2 - R^2 = (x_i^2 + y_i^2) - R^2 - y_i + \frac{1}{4} = O\Phi_i - y_i + \frac{1}{4};$$

$$O\Phi_i^H = (x_i - 1/2)^2 + y_i^2 - R^2 = (x_i^2 + y_i^2) - R^2 - x_i + \frac{1}{4} = O\Phi_i - x_i + \frac{1}{4}$$

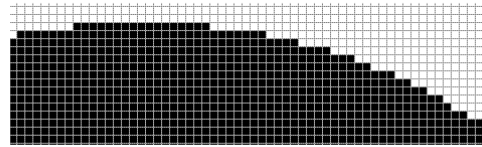
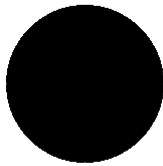
Приклади антиаліазингу меж кіл для методу додаткових оцінювальних функцій



Використано вісім додаткових оцінювальних функцій

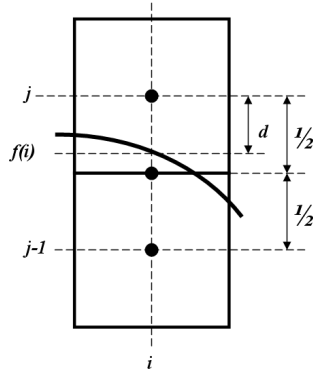


Використано чотири додаткові оцінювальні функції



Без антиаліазингу

Метод модифікованої оцінювальної функції для антиаліазингу меж еліпсів



$$I_p(i, j) = I_M S + I_\Phi(1 - S)$$

$$f(i) = \sqrt{A^2 - i^2 \frac{A^2}{B^2}}$$

$$OF'(i, j) = f^2(i) - (j - 0,5)^2$$

$$S \approx \frac{OF' + 0,25 - 0,5A^2/B^2}{2j}$$

Розрахунок інтенсивності кольору

$$x = OF' + 0,25 - 0,5A^2/B^2; \quad M = 2j; \quad I_p(i, j) = I_M \frac{x}{M}$$

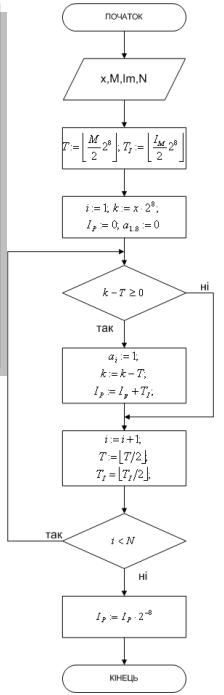
$$\frac{x}{M} = \frac{1}{2}a_1 + \frac{1}{4}a_2 + \frac{1}{8}a_3 + \dots + \frac{1}{256}a_8$$

$$I_p = \frac{I_M}{2}a_1 + \frac{I_\Phi}{2}a_1 + \frac{I_M}{4}a_2 + \frac{I_\Phi}{4}a_2 + \frac{I_M}{8}a_3 + \frac{I_\Phi}{8}a_3 + \dots + \frac{I_M}{256}a_8 + \frac{I_\Phi}{256}a_8$$

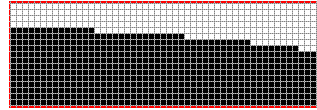
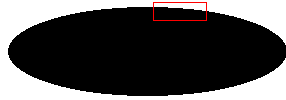
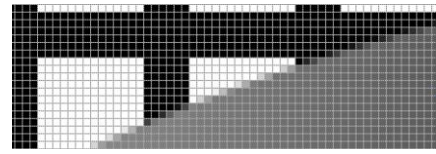
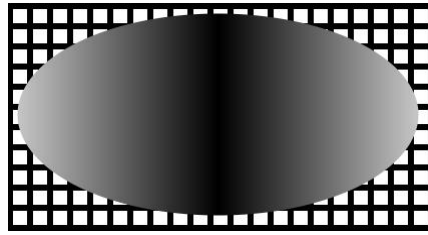
Знаходження коефіцієнтів $a_1 \div a_8$:

$a_i = 0$, якщо $k_{i-1} < \frac{M}{2^i}$; інакше $a_i = 1$.

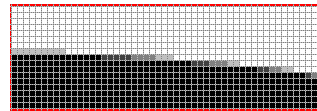
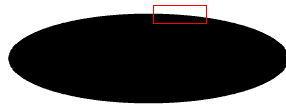
$k_0 = x$; $k_i = k_{i-1} - \frac{M}{2^i} a_i$.



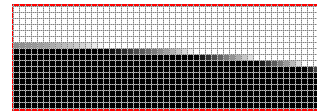
Приклад антиаліазингу границь еліпсів та гіпербол



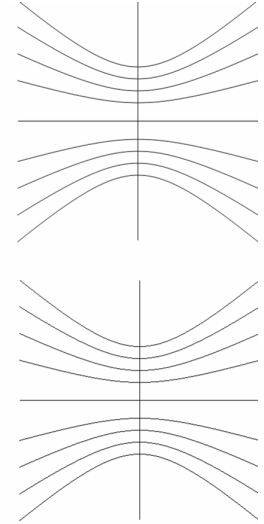
$N=0$



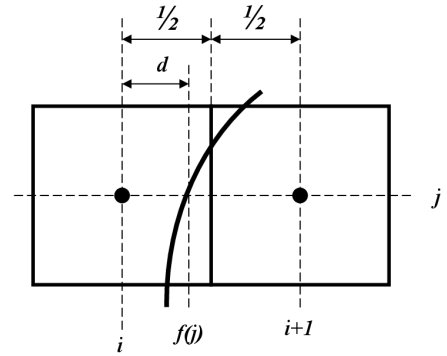
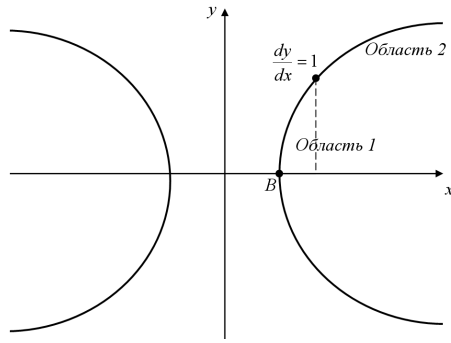
$N=2$



$N=8$

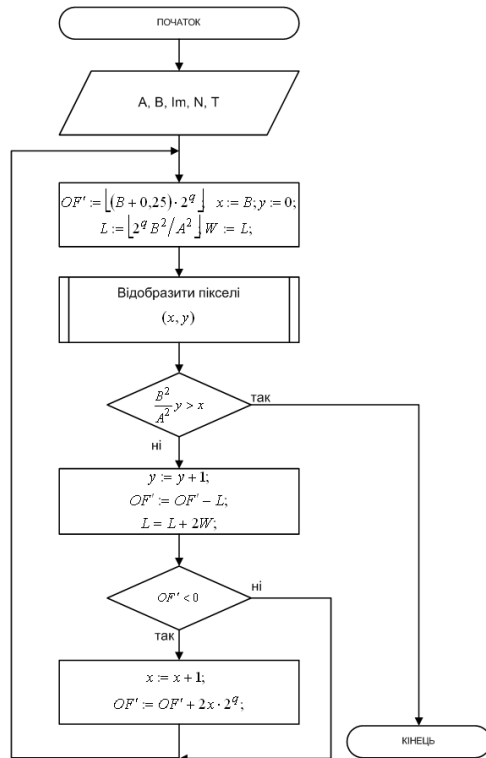


Гіпербола

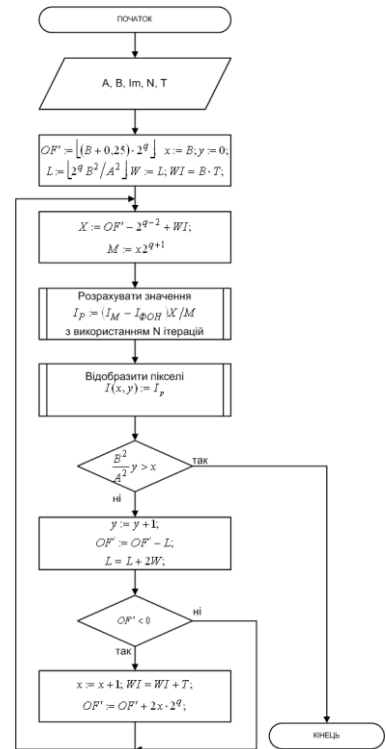


$$\begin{aligned}
 I_p(x, y) &= S_1(I_M - I_{\text{фон}}) + I_{\text{фон}} = \left(\frac{OF' - 0,25 + T \cdot x}{2x} \right) (I_M - I_{\text{фон}}) + I_{\text{фон}} = \\
 &= \frac{OF' - 0,25 + T \cdot x}{2x} (I_M - I_{\text{фон}}) + I_{\text{фон}}.
 \end{aligned}$$

Граф-схеми алгоритмів

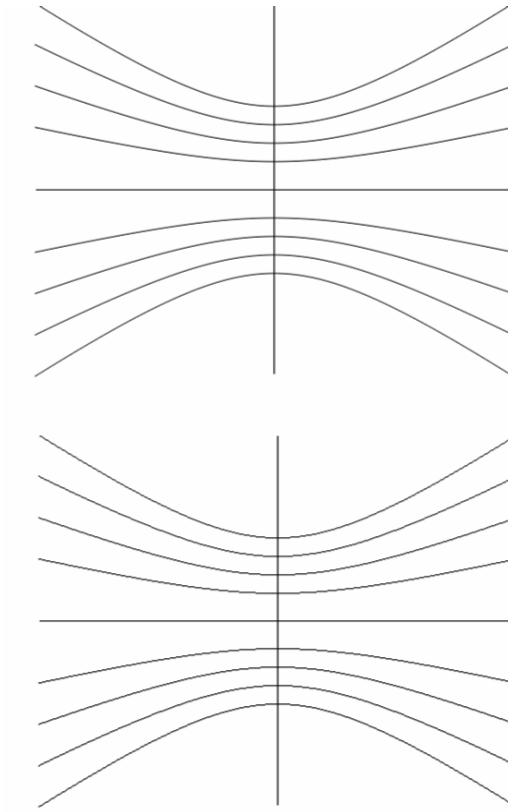


Граф-схема алгоритму визначення координат точок траєкторії гіперболи

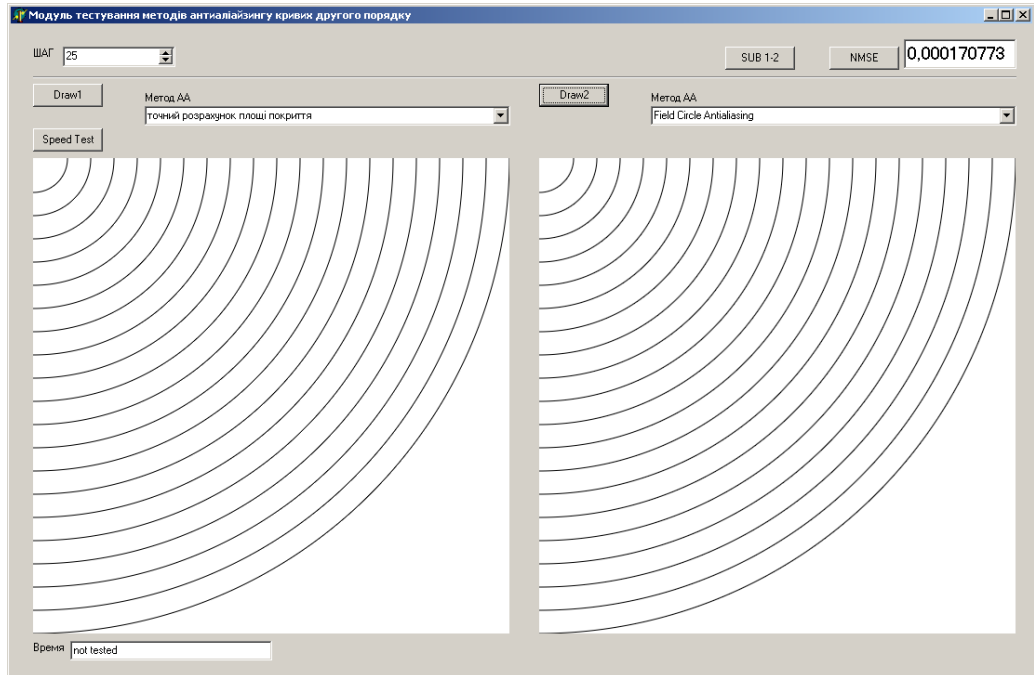


Граф-схема алгоритму антиаліазингу границі гіперболи

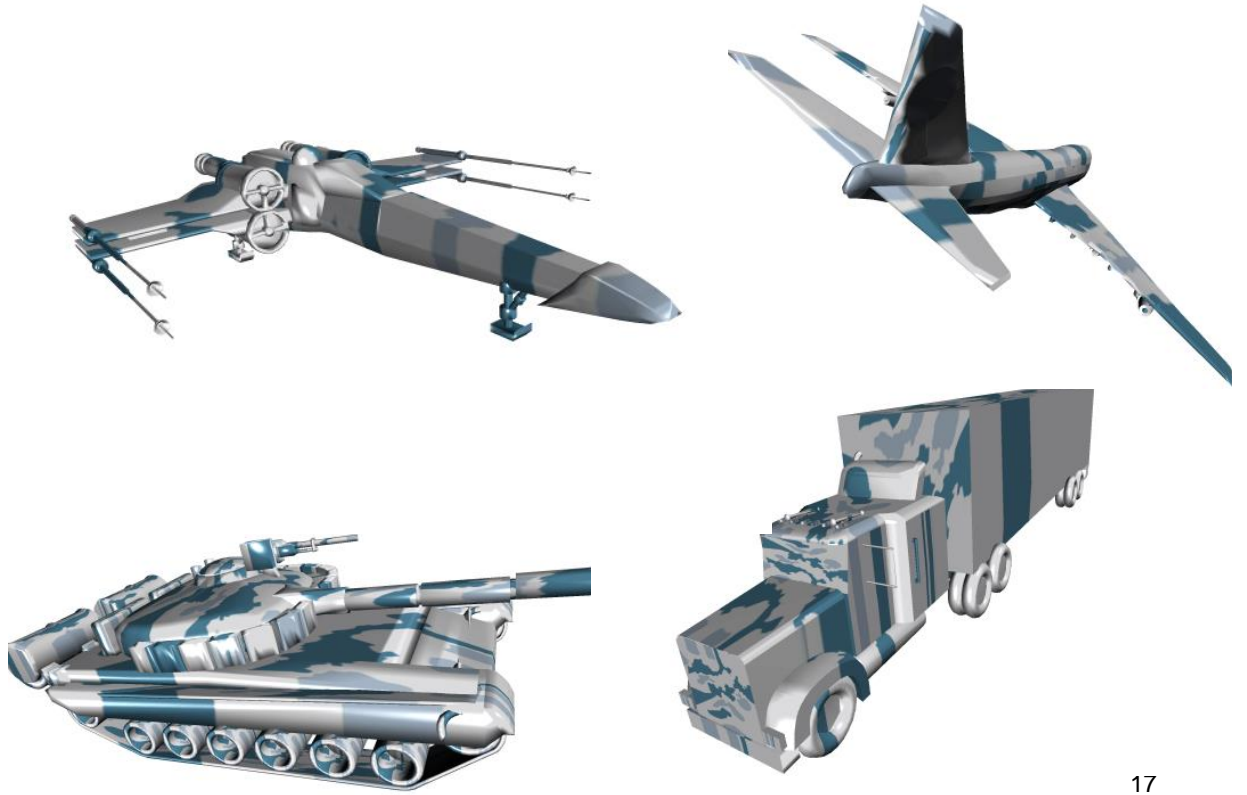
Результат роботи алгоритму антиаліайзингу границь гіпербол



Зовнішній вигляд інтерфейсу



Приклади зображень, сформованих із використанням розроблених алгоритмів



Наукова новизна. Практичне значення одержаних результатів

1

Наукова новизна одержаних результатів:

- вперше запропоновано метод антиаліаїзingu зображень траєкторій кіл, особливість якого полягає у розбитті пікселя на субпікселі з подальшим знаходженням для кожного з них знаків оцінювальних функцій, по яких розраховують площу покриття пікселя траєкторією, що дозволяє підвищити продуктивність згладження;
 - отримано нові аналітичні залежності для розрахунку площі покриття пікселя траєкторією, що дозволило модифікувати табличний метод антиаліаїзingu траєкторій кіл, що дозволило зменшити об'єми постійної пам'яті;
 - подальшого розвитку отримали методи антиаліаїзingu кривих другого порядку, в яких, на відміну від відомих, використано нові оцінювальні функції, що дозволило підвищити продуктивність згладження.
- Достовірність отриманих результатів підтверджується збігом результатів теоретичних досліджень із результатами комп'ютерного моделювання.
- Практичне значення одержаних результатів полягає у тому, що на основі проведених теоретичних досліджень і отриманих наукових результатів розроблено алгоритми та програми для антиаліаїзingu кривих другого порядку.

**Лістинг основних процедур програмного модуля для антиаліаїзінгу
кривих другого порядку**

```
unit Test2D;
interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, GR32_Image, ExtCtrls, Spin, Buttons, Math;

type
  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Edit1: TEdit;
    Image1: TImage;
    Image2: TImage;
    Button4: TButton;
    ComboBox1: TComboBox;
    ComboBox2: TComboBox;
    SpinEdit1: TSpinEdit;
    Label1: TLabel;
    Label2: TLabel;
    Edit3: TEdit;
    Button5: TButton;
    Bevel1: TBevel;
    Label3: TLabel;
    Label4: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure SpinEdit1Change(Sender: TObject);
    procedure Button5Click(Sender: TObject);
```

```

private
  { Private declarations }
public
  { Public declarations }
end;
function CalcSField(yp,ypold,Z:integer):integer;
function CalcVLine(d:integer):integer;
function CalcV(d:integer):integer;
function CalcVSquare(d:integer):integer;
function CalcVSq(d:integer):integer;
function CalcVReal(d:real):integer;

procedure DrawWuCircle(R:integer;ACanvas:TCanvas;Black:boolean);
procedure DrawGuptaSproulLine(dx,dy:integer;ACanvas:TCanvas);
procedure DrawGuptaSproulLine2(dx,dy:integer;ACanvas:TCanvas);
procedure DrawGuptaSproulLineReal(dx,dy:integer;ACanvas:TCanvas);
procedure DrawGuptaSproulLineVstr(dx,dy:integer;ACanvas:TCanvas);
procedure DrawGuptaSproulLine2Vstr(dx,dy:integer;ACanvas:TCanvas);
procedure DrawGuptaSproulLineRealVstr(dx,dy:integer;ACanvas:TCanvas);
procedure DrawFieldCircle(R:integer; ACanvas:TCanvas;Black:boolean);
procedure DrawModOFCircle(R:integer;ACanvas:TCanvas;Black:boolean);
procedure DrawDopOFCircle(R:integer;ACanvas:TCanvas;Black:boolean);

procedure CalcTableWu;
procedure LoadTable;
procedure DrawPixelsC(size:integer; ACanvas1:TCanvas);
procedure ClearPixelsC(size:integer);
function CheckOF4New(OFunc,xp,yp:integer):integer;
function Integral(xa,xb,R:real):real;
function SReal(R,xp,yp:integer):integer;
procedure DrawRealCircle(R:integer;ACanvas:TCanvas;Black:boolean);
var
  Form1: TForm1;
  Table: array [0..1005] of real;
  TableWu: array [0..1024,0..1024] of byte;
  str1:string;
  maximer:real;
  step:integer;
  xs,ys:integer;
  hold,hnew:integer;
  Sn:integer;

```

```

Su,yold,ynew:integer;
PixelsC: array [0..1000,0..1000] of integer;
implementation
{$R *.DFM}
//////////CALC V using Linear Approx!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!1
function CalcVLine(d:integer):integer;
var
S,dp,dp1,dp2,V1,V2,t,d1,d2:integer;
one:integer;
begin
one:=1 shl 10;
t:=one shr 1;
dp:=abs(d);
d1:=dp+t;
dp1:=abs(d1);
if(dp1 > one) then dp1:=one;
V1:=CalcV(dp1);
if d1<0 then V1:=one-V1;
d2:=dp-t;
dp2:=abs(d2);
if(dp2 > one) then dp2:=one;
V2:=CalcV(dp2);
if d2<0 then V2:=one-V2;
if V1<0 then V1:=0;
if V1>one then V1:=one;
if V2<0 then V2:=0;
if V2>one then V2:=one;

S:=V1-V2;
if S<0 then S:=0;
if S>one then S:=one;
result:=S shr 2;
end;

function CalcV(d:integer):integer;
var
one:integer;
three:integer;
d1:integer;
begin
one:=1 shl 10;

```

```

three:=3 shl 10;
if (d < 256) then d1:=512+((29*d) shr 5)
else
if (d < 512) then d1:=576+((d*85) shr 7)
else
if (d < 768) then d1:=734+((d*91) shr 8)
else
d1:=944+((d*41) shr 9);
result:=d1;
end;
//////////CALC V using 2-nd degree polynom
function CalcVSquare(d:integer):integer;
var
S,dp,dp1,dp2,V1,V2,t,d1,d2:integer;
one:integer;
begin
one:=1 shl 10;
t:=one shr 1;
dp:=abs(d);
d1:=dp+t;
dp1:=abs(d1);
if(dp1 > one) then dp1:=one;
V1:=CalcVSq(dp1);
if d1<0 then V1:=one-V1;
d2:=dp-t;
dp2:=abs(d2);
if(dp2 > one) then dp2:=one;
V2:=CalcVSq(dp2);
if d2<0 then V2:=one-V2;
S:=V1-V2;
if S<0 then S:=0;
if S>one then S:=one;
result:=(S shr 2);
end;
function CalcVSq(d:integer):integer;
var
one:integer;
d1:integer;
begin
one:=1 shl 10;
if d>= one then begin result:=one; exit; end;

```

```

d1:=(d*d) shr 10;
d1:=(one shr 1)+d-(d1 shr 1);
if d1> one then d1:=one;
if d1 < 0 then d1:=0;
result:=d1;
end;
procedure DrawGuptaSproulLineVstr(dx,dy:integer;ACanvas:TCanvas);
var
delta,BP,MP,OF1:integer;
xlim,Z,x,y,d,T,d_up,d_down:integer;
lp:integer;
er,maxe:real;
d_real:real;
begin
Z:=1 shl 20;
BP:=round((dx*Z)/sqrt(dx*dx+dy*dy));
MP:=round((dy*Z)/sqrt(dx*dx+dy*dy));
  maxe:=0;
OF1:=BP shr 1;
delta:=BP-MP;
x:=0;y:=0;
xlim:=(dx+1) shr 1;
while(x<xlim) do
begin
if (OF1>=0) then OF1:=OF1-MP
else OF1:=OF1+delta;
if (OF1>=0) then
begin
  x:=x+1;
  d:=(BP shr 1)-OF1;
  d_up:=d-BP;
  d_down:=d+BP;
end
else
begin
  x:=x+1;y:=y+1;
  d:=-OF1-(BP shr 1);
  d_up:=d-BP;
  d_down:=d+BP;
end;
end;
/// main pixel

```



```

lp:=CalcVLine((d div (1 shl 10)));
if lp>255 then lp:=255;
maximer:=lp;
PixelsC[x,y]:=RGB(255-lp,255-lp,255-lp);
PixelsC[y,x]:=RGB(255-lp,255-lp,255-lp);
PixelsC[dx-x,dy-y]:=RGB(255-lp,255-lp,255-lp);
PixelsC[dy-y,dx-x]:=RGB(255-lp,255-lp,255-lp);
/// upper pixel
lp:=CalcVLine((d_up div (1 shl 10)));
if lp>255 then lp:=255;
maximer:=lp;
PixelsC[x,y+1]:=RGB(255-lp,255-lp,255-lp);
PixelsC[y+1,x]:=RGB(255-lp,255-lp,255-lp);
if (dy-y-1)>0 then
begin
PixelsC[dx-x,dy-(y+1)]:=RGB(255-lp,255-lp,255-lp);
PixelsC[dy-(y+1),dx-x]:=RGB(255-lp,255-lp,255-lp);
end;
/// low pixel
lp:=CalcVLine((d_down div (1 shl 10)));
if lp>255 then lp:=255;
maximer:=lp;
if(y>0) then begin
PixelsC[x,y-1]:=RGB(255-lp,255-lp,255-lp);
PixelsC[y-1,x]:=RGB(255-lp,255-lp,255-lp);
end;
if (dy-y+1)>0 then
begin
PixelsC[dx-x,dy-(y-1)]:=RGB(255-lp,255-lp,255-lp);
PixelsC[dy-(y-1),dx-x]:=RGB(255-lp,255-lp,255-lp);
end;
end;
maximer:=maxe;
end;

procedure DrawGuptaSproulLineRealVstr(dx,dy:integer;ACanvas:TCanvas);
var
T,delta,BP,MP,OF1,d,d_up,d_down:real;
x,y:integer;
lp:integer;

```

```

er,maxe:real;
d_real:real;
xlim:integer;
begin
T:=sqrt(dx*dx+dy*dy);
BP:=dx/T;
MP:=dy/T;
  maxe:=0;
OF1:=BP/2;
delta:=BP-MP;
x:=0;y:=0;
xlim:=(dx+1) shr 1;
while(x<xlim) do
begin
if (OF1>=0) then OF1:=OF1-MP
else OF1:=OF1+delta;

if (OF1>=0) then
begin
  x:=x+1;
  d:=(BP/2)-OF1;
  d_up:=d-BP;
  d_down:=d+BP;
end
else
begin
  x:=x+1;y:=y+1;
  d:=-OF1-(BP/2);
  d_up:=d-BP;
  d_down:=d+BP;
end;
lp:=CalcVReal(d);
if lp>255 then lp:=255;
PixelsC[x,y]:=RGB(255-lp,255-lp,255-lp);
PixelsC[y,x]:=RGB(255-lp,255-lp,255-lp);
PixelsC[dx-x,dy-y]:=RGB(255-lp,255-lp,255-lp);
PixelsC[dy-y,dx-x]:=RGB(255-lp,255-lp,255-lp);
lp:=CalcVReal(d_up);
if lp>255 then lp:=255;
PixelsC[x,y+1]:=RGB(255-lp,255-lp,255-lp);
PixelsC[y+1,x]:=RGB(255-lp,255-lp,255-lp);

```

```

if (dy-y-1)>0 then
begin
  PixelsC[dx-x,dy-(y+1)]:=RGB(255-lp,255-lp,255-lp);
  PixelsC[dy-(y+1),dx-x]:=RGB(255-lp,255-lp,255-lp);
end;
lp:=CalcVReal(d_down);
  if lp>255 then lp:=255;
if(y>0)
then begin
  PixelsC[x,y-1]:=RGB(255-lp,255-lp,255-lp);
  PixelsC[y-1,x]:=RGB(255-lp,255-lp,255-lp);
end;
if (dy-y+1)>0 then
begin
  PixelsC[dx-x,dy-(y-1)]:=RGB(255-lp,255-lp,255-lp);
  PixelsC[dy-(y-1),dx-x]:=RGB(255-lp,255-lp,255-lp);
end;
end;
end;
end;
procedure CalcTableWu;
var
r,j:integer;
sq:real;
begin
for r:=1 to 1024 do
begin
for j:=1 to r do
begin
sq:=sqrt(r*r*1.0-j*j*1.0);
TableWu[r,j]:=floor(255*( Ceil(sq)*1.0 - sq)+0.5);
end;
end;
end;
end;
procedure DrawWuCircle(R:integer;ACanvas:TCanvas;Black:boolean);
var
c1,j,x,y:integer;
l,T,D:byte;
begin
x:=R;
y:=0;
T:=0;

```

```

if (Black) then
begin
PixelsC[x,y]:=clBlack;
PixelsC[y,x]:=clBlack;
c1:=clBlack;
end
else
begin
PixelsC[x,y]:=clWhite;
PixelsC[y,x]:=clWhite;
c1:=clWhite;
end;
if (Black) then ACanvas.Pen.Color:=clBlack else ACanvas.Pen.Color:=clWhite;
while(y<(x-1)) do
begin
y:=y+1;
D:=TableWu[R,y];
if (D<T) then x:=x-1;
if (Black) then l:=not D else l:=D;
PixelsC[x,y]:=RGB(255-l,255-l,255-l);
T:=D;
ACanvas.MoveTo(y,0); ACanvas.LineTo(y,x);
PixelsC[y,x]:=RGB(255-l,255-l,255-l);
end;
end;
procedure DrawFieldCircle(R:integer; ACanvas:TCanvas;Black:boolean);
var
x,y:integer;
f2:integer;
c1,i,Z,R1:integer;
lp1,lp:integer;
ypo,hi,hi1,k2,k,d,d1,d2,er,erm,yp1,yp2,yp:integer;
begin
Z:=256;
y:=R*Z;
x:=0;
f2:=R*R*Z*Z-Z*(Z div 4);
yp2:=round(Z*sqrt(R*R-0.5*0.5));
yp1:=round(Z*sqrt(R*R-1.5*1.5));
yp:=yp1;
ypo:=yp1;

```

```

hi:=64;
// PutPixels 0,1,2
if (not Black) then ACanvas.Pen.Color:=clWhite else ACanvas.Pen.Color:=clBlack;
    if (Black) then begin lp:=255-Su; lp1:=255-Sn;c1:=clBlack; end
    else begin lp:=Su; lp1:=Sn;c1:=clWhite; end;
lp:=CalcSField(yp2,yp2,Z);
if (Sn = -1) then
    begin
        if (Black) then lp:=255-Su else lp:=Su;
        PixelsC[x,yold]:=RGB(lp,lp,lp);
        PixelsC[yold,x]:=RGB(lp,lp,lp);
    end
    else
    begin
        if (Black) then begin lp:=255-Su; lp1:=255-Sn; end
        else begin lp:=Su; lp1:=Sn; end;
        PixelsC[x,yold]:=RGB(lp,lp,lp);
        PixelsC[x,ynew]:=RGB(lp1,lp1,lp1);
        PixelsC[yold,x]:=RGB(255-Su,255-Su,255-Su);
        PixelsC[ynew,x]:=RGB(255-Sn,255-Sn,255-Sn);
    end;
x:=x+1;
lp:=CalcSField(yp1,yp2,Z);
if (Sn = -1) then
    begin
        if (Black) then begin lp:=255-Su; lp1:=255-Sn; end
        else begin lp:=Su; lp1:=Sn; end;
        PixelsC[x,yold]:=RGB(lp,lp,lp);
        PixelsC[yold,x]:=RGB(lp,lp,lp);
    end
    else
    begin
        if (Black) then begin lp:=255-Su; lp1:=255-Sn; c1:=clBlack; end
        else begin lp:=Su; lp1:=Sn; c1:=clWhite;end;
        PixelsC[x,yold]:=RGB(lp,lp,lp);
        PixelsC[x,ynew]:=RGB(lp1,lp1,lp1);
        PixelsC[yold,x]:=RGB(lp,lp,lp);
        PixelsC[ynew,x]:=RGB(lp1,lp1,lp1);
    end;
while(x<y) do
begin

```

```

f2:=f2-2*x*Z*Z-Z*Z-2*Z*Z;
x:=x+1;
yp:=2*yp1-yp2;
//////////minimize error
er:=yp*yp-f2;
if (er > 0) then
  begin
    d:=-1;
    erm:=er;
    k:=0;
    k2:=0;
    while(erm>0) do
      begin
        erm:=erm-2*yp+2*k+1;
        k:=k+1;
      end;
      yp:=yp-k+1;
    end
  else
    begin
      d:=1;
      erm:=er;
      k:=0;
      k2:=0;
      while(erm<0) do
        begin
          erm:=erm+2*yp+2*k+1;
          k:=k+1;
        end;
        yp:=yp+k-1;
      end;
    end;
  end;
lp:=CalcSField(yp,ypo,Z);
if (Sn = -1) then
  begin
    if (Black) then begin lp:=255-Su; lp1:=255-Sn;c1:=clBlack; end
    else begin lp:=Su; lp1:=Sn;c1:=clWhite; end;
    PixelsC[x,yold]:=RGB(lp,lp,lp);
    ACanvas.MoveTo(x,0); ACanvas.LineTo(x,yold);
    PixelsC[yold,x]:=RGB(lp,lp,lp);
    y:=yold;
  end
end

```

```
else
begin
  if (Black) then begin lp:=255-Su; lp1:=255-Sn; c1:=clBlack;end
  else begin lp:=Su; lp1:=Sn; c1:=clWhite; end;
  PixelsC[x,yold]:=RGB(lp,lp,lp);
  PixelsC[x,ynew]:=RGB(lp1,lp1,lp1);
  PixelsC[yold,x]:=RGB(lp,lp,lp);
  PixelsC[ynew,x]:=RGB(lp1,lp1,lp1);
  if (Black) then begin c1:=clBlack; end
  else begin c1:=clWhite; end;
y:=ynew;
end;
yp1:=yp;
yp2:=yp1;
yp0:=yp;
end;
end;
```