УДК 681.3.082.5

О. Н. Романюк, к. т. н., доц.;
М. С. Курінний, асп.

ВИКОРИСТАННЯ КОНУСНОЇ МОДЕЛІ ПІКСЕЛЯ ДЛЯ АНТИАЛІАЙЗИНГУ ВІДРІЗКІВ ПРЯМИХ

1. Вступ

Під час дискретизації неперервних зображень виникають спотворення, обумовлені недостатньою роздільною здатністю дискретної решітки. На неперервних краях об'єктів з'являються чітко виражені сходинки або зубці. Цей артефакт отримав назву ступінчастого ефекту чи ефекту *аліайзингу* [1]. Ефект аліайзингу суттєво погіршує якість сформованого зображення. Встановлено, що у разі використання 17-дюймового монітора і розміщенні спостерігача на відстані 65 см від екрану для повного усунення ефекту аліайзингу потрібен монітор з роздільною здатністю як мінімум 4000×4000 пікселів, а для людей з рівнем зору вище середнього — взагалі 8000×8000 пікселів [2]. Сучасний рівень технологій поки що не в змозі забезпечити таку роздільну здатність. Таким чином, для забезпечення належної якості зображень необхідно використовувати спеціальні методи антиаліайзингу.

Метою даної статі є розробка ефективного методу антиаліайзингу для відрізків прямих.

2. Аналіз наявних методів та постановка задачі

Методи усунення ступінчастого ефекту розділяють на дві групи [1]. До першої групи відносяться методи, які базуються на збільшенні дискретизації [3]. В таких методах сцена спочатку розраховується з вищою роздільною здатністю, що дає можливість врахувати дрібні деталі, а під час відображення зменшується шляхом усереднення. Основний недолік цих методів полягає у великій обчислювальній складності, оскільки зі збільшенням дискретизації в n разів, кількість пікселів (а отже і кількість обчислень на один піксель) збільшується в n^2 разів [3].

В методах другої групи піксель розглядається не як умовна точка, а як скінченна область, оскільки в реальних пристроях відображення піксель не є ідеальною точкою, а має певну форму [1]. При цьому інтенсивність пікселів на краях графічних об'єктів встановлюється пропорційно до площі тих частин пікселів, які покриваються цим об'єктом. Методи даної групи характеризуються меншою обчислювальною складністю і є більш прийнятними для випадків, коли необхідно генерувати зображення в реальному масштабі часу.

Складність обчислення площі покриття та якість результативного зображення залежить від математичної моделі форми пікселя. В більшості наявних алгоритмів піксель розглядається як квадрат зі стороною, що дорівнює одиниці, оскільки при цьому значно спрощуються обчислення. Математична модель пікселя, в якій останній розглядається як круг з діаметром, що дорівнює одиниці, більш адекватна реальності. Вищу якість забезпечують моделі, які враховують, що інтенсивність світла, яке випромінює піксель, є максимальною в центрі пікселя та зменшується з віддаленням від нього [3]. Найпростішою серед них є модель, в якій інтенсивність кольору пікселя максимальна в центрі і лінійно зменшується до нуля на відстані r від центра. Така модель отримала назву конічної [4].

Використовуючи конічну модель, інтенсивність кольору пікселів визначається таким чином. В центрі пікселя встановлюється конус, висота і радіус якого підібрані так, щоб об’єм конуса дорівнював 1. Інтенсивність кольору точки встановлюється пропорційно до об’єму, який границя графічного примітиву відтинає від конуса, встановленого в центрі пікселя. Приклад застосування конічної моделі для відрізка прямої показаний на рис. 1.

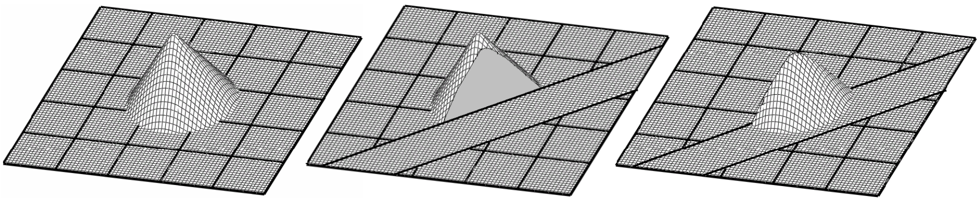

Рис. 1 Приклад застосування конічної моделі для відрізка прямої

Одним з найвідоміших алгоритмів антиаліайзингу відрізків прямих, який використовує конічну модель, є алгоритм Гупти-Спроула [3]. Для даного алгоритму радіус основи конуса r дорівнює одній дискреті. Вважається, що пряма має форму смуги, а відрізок прямої розглядається як прямокутник ширина якого дорівнює t. Об’єм, який відтинається прямою від конуса, залежить лише від відстані d між центром пікселя та прямою і товщини прямої t [4]. Відповідно інтенсивність кольору пікселя визначається за формулою

$$I = F(d,t),$$

де d — відстань між прямою та центром пікселя; t — товщина прямої.

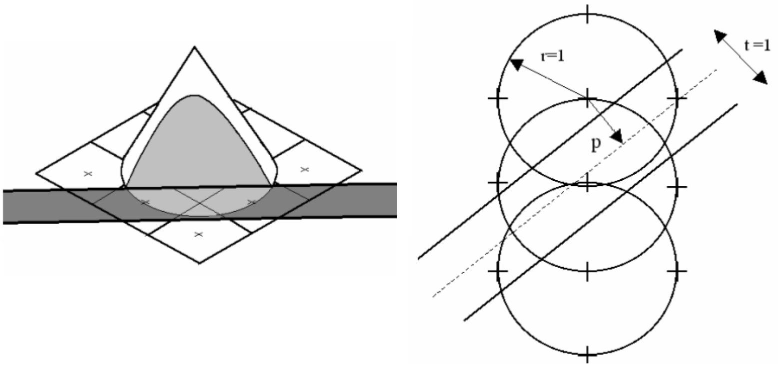

Рис. 2 Визначення інтенсивності пікселів згідно алгоритму Гупти-Спроула

Алгоритм на кожному кроці підсвічує три пікселя: центральний та два його сусіда зверху та знизу (рис. 2).

Для обчислення виразу $F(d,t)$ застосовується чисельне інтегрування [4]. В алгоритмі Гупти-Спроула використовується таблиця заздалегідь розрахованих значень функції $F(d,t)$ для $t = 1$. Основний недолік такого підходу полягає у тому, що для ліній різної товщини потрібно будувати окремі таблиці, що ускладнює апаратну реалізацію.

Отже, актуальними є питання пошуку шляхів обчислення об’єму перетину безпосередньо в циклі інтерполювання. Одним з можливих підходів є апроксимація функції $F(d,t)$ поліномом 2-го ступеня.

3. Результати досліджень

Об’єм частини конуса знаходиться за формулою:

$$F(d,t) = V(d+t) - V(d-t),$$

де $V(d)$ — об’єм, який відтинається від конуса площиною, перпендикулярною до основи конуса.

З рис. 3 видно, що об’єм $V(d)$ можна знайти за формулою:

$$V(d) = \int_0^d S(t)\,dt, \tag{1}$$

де $S(t)$ — площа поперечного перерізу конуса площиною, яка перпендикулярна до основи конуса і розміщена на відстані t від центра конуса (рис. 3); d — відстань між центром пікселя і ідеальною прямою.

Для знаходження виразу $S(t)$ будемо використовувати рівняння конусу в системі координат, центр якої збігається з центром конуса (рис. 3):

$$z(x,y) = \begin{cases} H - \dfrac{H}{R}\sqrt{x^2+y^2}, \; \text{для } \sqrt{x^2+y^2} \le R \\ 0, \; \text{для } \sqrt{x^2+y^2} > R \end{cases},$$

де H — висота конуса; R — радіус конуса.

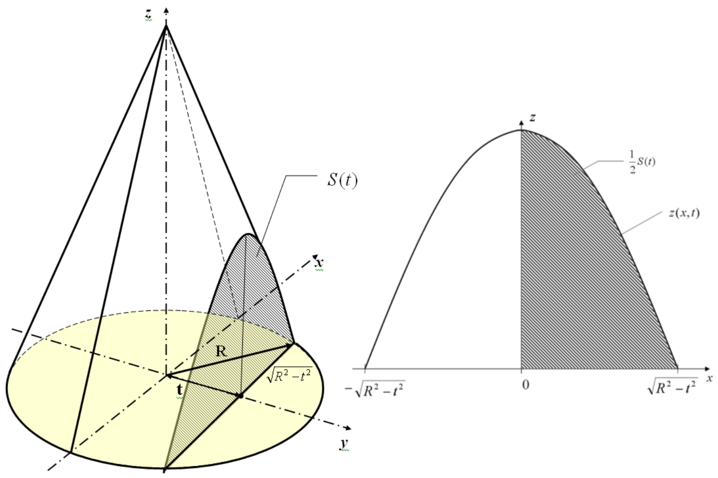

Рис. 3. Знаходження об’єму, який відтинається від конуса ідеальною площиною

З рис. 3 видно, що площу поперечного перерізу $S(t)$ можна знайти, обчисливши такий інтеграл

$$S(t) = 2\int_0^{\sqrt{R^2-t^2}} z(x,t)\,dx = 2\int_0^{\sqrt{R^2-t^2}} \left(H - \frac{H}{R}\sqrt{x^2+t^2} \right) dx.$$

Обчисливши інтеграл, отримуємо

$$S(t) = H\sqrt{1-t^2} - Ht^2 \ln\left(\sqrt{1-t^2}+1\right) + Ht^2 \ln(t).$$

Підставивши останній вираз в (1), знаходимо:

$$V(d) = \int_0^d \left(H\sqrt{1-t^2} - Ht^2 \ln\left(\sqrt{1-t^2}+1\right) + Ht^2 \ln(t) \right) dt. \tag{2}$$

Обчислимо даний інтеграл за методом адаптивної квадратури [5] для $d = 0{,}05; 0{,}1; 0{,}15 \ldots 1$. Результати обчислень зведені в таблицю.

 ISSN 1997-9266. Вісник Вінницького політехнічного інституту. 2004. № 4

Результати обчислення інтегралу (2)

d	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5	0,55	0,6	0,65	0,7	0,75	0,8	0,0,8	0,9	0,95	1
V	0,024	0,047	0,070	0,092	0,112	0,132	0,150	0,166	0,181	0,195	0,207	0,217	0,226	0,234	0,239	0,244	0,247	0,249	0,250	0,250

Після апроксимації даних, наведених у таблиці, за допомогою методу найменших квадратів [5] було отримано такий поліном

$$V_p(d) = -0,571 \cdot d^2 + 1,076 \cdot d - 6,774 \cdot 10^{-3}.$$

Приведений поліном відносно складний, тому доцільно використати більш простий з прийнятною точністю апроксимації. Дослідимо поліном:

$$V_{p1}(d) = -0,5 \cdot d^2 + 1 \cdot d.$$

На рис. 4 показані графіки залежності абсолютної похибки функцій $V_p(d)$ та $V_{p1}(d)$ від відстані d.

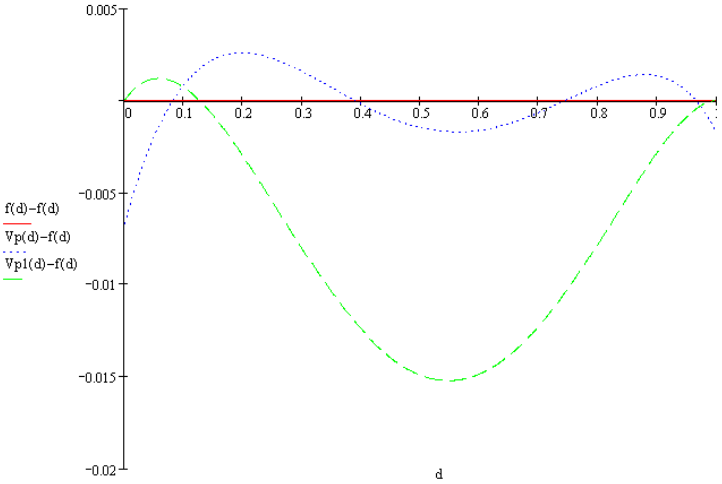

Рис. 4. Графіки залежності абсолютної похибки функцій $V_p(d)$ та $V_{p1}(d)$ від відстані d

З графіка видно, що абсолютна похибка в разі використання полінома $V_{p1}(d)$ не перевищує 1,5 % від максимального рівня інтенсивності. З використанням 256 рівнів інтенсивності різниця в найгіршому випадку складе 4 рівня. В роботі [6] показано, що таку різницю людське око за нормальних умов розрізнити не в змозі.

Для ефективного застосування конічної моделі необхідно зменшити обчислювальні витрати на знаходження відстані від центра пікселя до прямої. Найпоширенішими методами побудови відрізків прямих є методи, основані на використанні оцінювальних функцій [6]. Знайдемо зв'язок між оцінювальною функцією та відстанню від центра пікселя до прямої.

Як базовий використаємо алгоритм лінійної інтерполяції за методом оцінювальної функції з початковим значенням, яке дорівнює $\lfloor БП/2 \rfloor$ [7].

Оцінювальна функція розраховується за формулами

$$F_{i+1} = \begin{cases} F_i - МП, & \text{якщо } F_i \geq 0; \\ F_i + \Delta, & \text{якщо } F_i < 0, \end{cases}$$

де $F_0 = \lfloor БП/2 \rfloor$, $\Delta = БП - МП$, БП – більший координатний приріст відрізка, МП — менший координатний приріст відрізка.

Якщо $F_i \geq 0$, то виконується крок по провідній координаті. Коли $F_i < 0$, то виконується комбінований крок (по обох координатах). Даний алгоритм забезпечує максимальну точність.

В загальному вигляді для даного алгоритму залежність оцінювальної функції від координат точки має вигляд:

$$F(x, y) = \text{БП} \cdot y - \text{МП} \cdot x + \text{БП}/2\,. \tag{3}$$

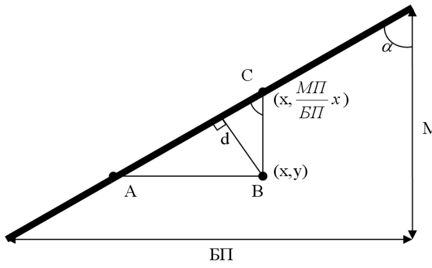

Рис. 5. Залежність між оцінювальною функцією та відстанню від точки до прямої

З рис. 5 видно, що відстань від центра пікселя до прямої можна знайти за формулою

$$d = \text{BC} \cdot \sin\alpha\,, \tag{4}$$

де α — кут нахилу прямої.

Відстань BC дорівнює

$$\text{BC} = \frac{\text{МП}}{\text{БП}}x - y\,.$$

Синус кута нахилу можна знайти за допомогою такого виразу

$$\sin\alpha = \frac{\text{БП}}{\sqrt{\text{БП}^2 + \text{МП}^2}}\,.$$

Підставивши два останні вирази у вираз (4) отримуємо наступну формулу для знаходження відстані від центра пікселя до прямої

$$d = \frac{\text{МП} \cdot x - \text{БП} \cdot y}{\sqrt{\text{БП}^2 + \text{МП}^2}} = \frac{\text{БП}/2 - \left(\text{БП} \cdot y - \text{МП} \cdot x + \text{БП}/2\right)}{\sqrt{\text{БП}^2 + \text{МП}^2}} = \frac{\text{БП}/2 - \text{ОФ}}{\sqrt{\text{БП}^2 + \text{МП}^2}}\,.$$

Розглянемо таку модифікацію оцінювальної функції:

$$OF = \text{БП}' \cdot y - \text{МП}' \cdot x + \text{БП}'/2\,,$$

де $\text{БП}' = \dfrac{\text{БП}}{\sqrt{\text{БП}^2 + \text{МП}^2}}$; $\text{МП}' = \dfrac{\text{МП}}{\sqrt{\text{БП}^2 + \text{МП}^2}}$.

Знак даної функції співпадає зі знаком функції (3), отже її можна використати замість оцінювальної функції (3) під час визначення координат точок траєкторії. Таким чином інтерполювання відрізка прямої з параметрами БП та МП зводиться до інтерполювання за БП тактів відрізка прямої з параметрами $\text{БП}'$ та $\text{МП}'$. При чому знак оцінювальної функції використовується для визначення координат точок траєкторії, а значення оцінювальної функції визначає відстань від центра пікселя до прямої.

Врахувавши все вищесказане, можна запропонувати такий алгоритм антиаліайзингу відрізка прямої для першого октанту.

1. Встановлюються початкові значення змінних:

$$T = \sqrt{\text{БП}^2 + \text{МП}^2}\,;\ x := 0\,,\ y := 0\,;\quad \text{БП}' = \text{БП}/T\,;$$

$$\text{МП}' = \text{МП}/T\,;\ OF = \text{БП}'/2\,;\ \Delta = \text{БП}' - \text{МП}'\,;$$

2. Поточні значення оцінювальної функції визначаються згідно з виразами

$$OF_{i+1} = \begin{cases} OF_i - \text{МП}', & \text{якщо } OF_i \geq 0; \\ OF_i + \Delta, & \text{якщо } OF_i < 0; \end{cases}$$

3. Якщо $OF \geq 0$, то виконується горизонтальний крок $x := x + 1$. Знаходиться відстань від центра пікселя до прямої: $d = \text{БП}'/2 - OF$;

ISSN 1997-9266. Вісник Вінницького політехнічного інституту. 2004. № 4

4. Якщо $OF' < 0$, виконується діагональний крок $x := x+1$; $y := y+1$. Знаходиться відстань від центра пікселя до прямої: $d = \left|OF'\right| - БП'/2$;

5. Для визначення інтенсивності кольору точок знаходять:

$$d_1 = |d| + t;\ d_2 = |d| - t;$$

$$\text{ЯКЩО } d_1 > 1 \text{ ТО } d_1 = 1;$$

$$\text{ЯКЩО } d_2 < -1, \text{ ТО } d_2 = -1;$$

$$\text{ЯКЩО } d_1 \geq 0 \text{ ТО } V_1 := 1/2 + \left(d_1 - 0{,}5 \cdot d_1^2\right) \text{ ІНАКШЕ } V_1 := 1/2 - \left(\left|d_1\right| - 0{,}5 \cdot d_1^2\right).$$

$$\text{ЯКЩО } d_2 \geq 0 \text{ ТО } V_2 := 1/2 + \left(d_2 - 0{,}5 \cdot d_2^2\right) \text{ ІНАКШЕ } V_2 := 1/2 - \left(\left|d_2\right| - 0{,}5 \cdot d_2^2\right).$$

$$I = \left(V_1 - V_2\right) I_M.$$

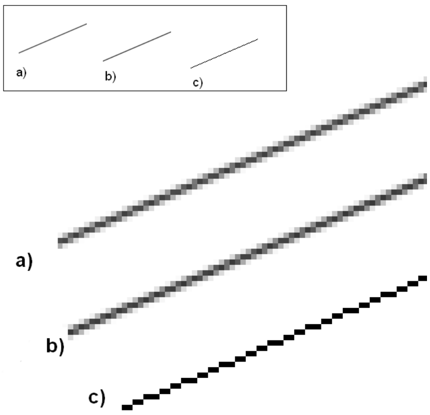

Рис. 6 Приклад роботи алгоритму, відрізки прямих побудовані з використанням: а) запропонованого алгоритму; b) алгоритму Гупти-Спроула; c) алгоритму Брезенхема

6. Як і в алгоритмі Гупти-Спроула [4], крім основного пікселя також заповнюється «нижній» та «верхній» пікселі. Для них інтенсивність кольору визначається аналогічно, з урахуванням того, що відстані від центра пікселя до прямої знаходяться за формулами: $d_{верх} = d + БП'$, $d_{низ} = \left|БП' - d\right|$.

7. Кроки 2—6 виконуються до закінчення формування відрізка.

Приклад результату роботи алгоритму показаний на рис. 6.

4. Висновки

Запропонований алгоритм не потребує використання таблиці заздалегідь розрахованих значень об'єму перетину, що робить більш простою його апаратну реалізацію. Крім того він містить таку ж кількість операцій типу «множення», як і в алгоритмі Гупти-Спроула наведенному в [4].

Запропонований алгоритм може бути використаний у системах генерації графічних образів з метою покращення якості зображень.

СПИСОК ЛІТЕРАТУРИ

1. Роджерс Д. Алгоритмические основы машинной графики. Пер. с англ. — М.: Мир, 1989. — 512 с.
2. William J. Leler. Human Vision, Anti-aliasing, and the cheap 4000 Line Display // ACM. — 1980. — Vol. 14. — P. 308—313.
3. Crow, Franklin C. A Comparison of Antialiasing Techniques. // IEEE CG & A. — 1981. — Vol. 1. — P. 40—47.
4. S. Gupta and R. F. Sproull. Filtering edges for gray-scale displays. // Computer Graphics. — 1981. —Vol. 15. — No. 3. — P. 1—5.
5. Лященко М. Я., Головань М. С. Чисельні методи: Підручник. — К.: Либідь, 1996. — 288 с
6. Booth, K. S., Bryden, M. B., Cowan, W. B., Morgan, M. F., Plante, B. L. On The Parameters Of Human Visual Performance: An Investigation Of The Benefits Of Antialiasing // CGA (7). — 1987. — No. 9. — P. 34—41.
7. Ободник Д. Т. Анализ и разработка цифровых интерполяторов для систем отображения измерительной информации. Дис. канд. техн. наук: 05.11.16. — Винница ВПИ, 1983. — 160 с.

Рекомендована кафедрою програмного забезпечення

Надійшла до редакції 10.02.04
Рекомендована до опублікування 19.03.04

Романюк Олександр Никифорович — доцент, ***Курінний Михайло Сергійович*** — аспірант.

Кафедра програмного забезпечення, Вінницький національний технічний університет