

УДК 004.925

О.Н. Романюк, М.Д. ОбідникВінницький національний технічний університет, м. Вінниця
кафедра програмного забезпеченняE-mail: ran12345@mail.ru, nikolya_vin@mail.ru**ОДИН ІЗ ПІДХОДІВ ДО ПІДВИЩЕННЯ ШВИДКОДІЇ ЗАФАРБОВУВАННЯ****Анотація**

Романюк О.Н., Обідник М.Д. Один із підходів до підвищення швидкодії зафарбовування. Запропоновано метод підвищення швидкодії зафарбовування, в якому за рахунок використання нових апроксимаційних формул суттєво зменшена обчислювальна складність процедур визначення векторів, а також інтенсивностей кольору.

Ключові слова: зафарбовування, вектори нормалей, растеризація, інтенсивність кольору.

Загальна постановка задачі.

Графічна форма подання інформації є найбільш інформативною, тому реалістичні зображення використовуються майже в усіх інженерних і наукових задачах для візуального відтворення реальних об'єктів. Формування реалістичних зображень характеризується значними обчислювальними витратами, що обумовлює підвищені вимоги як до засобів, так і до методів формування графічних зображень.

При синтезі графічних сцен необхідно вирішувати двоєдину задачу – забезпечення високої реалістичності графічних об'єктів і досягнення прийнятної для конкретної задачі часу формування графічних сцен.

Підвищення реалістичності відтворення графічних сцен передбачає використання більш складних моделей об'єктів реального світу та методів рендерингу [1-6], збільшення рівня деталізації поверхонь для коректної апроксимації об'єктів реального світу. Причому темпи зростання геометричної складності тривимірних зображень перевищують темпи зростання продуктивності графічних засобів.

Деталізація об'єктів, яка визначається щільністю триангуляційної мережі, повинна бути достатньою для адекватного відтворення конструктивних особливостей об'єкту. Вважається, що для екрана з роздільною здатністю 1280×1024 при середній кількості до 100000 полігонів у сцені забезпечується задовільна за сучасними мірками якість зображення, 500000 полігонів забезпечують якість, близьку до побутового відео, понад 1000000 полігонів у сцені дозволяє наблизитися до фотореалізму [1, 3]. В сучасних комп'ютерних іграх в сцені використовується в середньому 300000–500000 полігонів. У найближчі роки очікуються ігри [2], що використовуватимуть до 1000000–3000000 полігонів у сцені. Підвищення якості зображень припускає подальше зростання кількості вершин полігонів у моделях, причому темпи цього зростання обганяють темпи росту швидкості графічних прискорювачів [5].

Найбільшу обчислювальну складність має процес тонування (зафарбовування) полігонів, що відбувається на етапі рендерингу в графічному конвеєрі [5]. Проте теоретичних основ зафарбовування високополігональних поверхонь не розроблено. У статті запропоновано один з підходів до підвищення швидкодії зафарбовування високополігональних сцен.

Аналіз методів і постановка задачі.

Найбільшого поширення отримали такі методи тонування:

- плоске тонування [4];
- тонування за Гуро [4];
- тонування за Фонгом [5].

Пласке тонування передбачає зафарбовування полігона одним кольором [4]. Тому, навіть при суттєвому збільшенні кількості полігонів, з яких складається графічний об'єкт, він не може забезпечити необхідну реалістичність зображення.

У методі Гуро розраховуються значення інтенсивності кольору для полігональних вершин, які потім у процесі растеризації лінійно інтерполюються уздовж ребер і рядків сканування [4]. Алгоритмічна та схемотехнічна простота методу Гуро, можливість виконання паралельних незалежних обчислень спонукає багатьох науковців продовжувати роботи з удосконалювання цього підходу.

Згідно з методом Гуро кожен полігон зафарбовується шляхом лінійної інтерполяції значень інтенсивності кольору у вершинах, спочатку уздовж кожного ребра, а потім і між ребрами уздовж кожного скануючого рядка (рисунк 1).

Для всіх вершин запам'ятовується початкова інтенсивність, а також зміна інтенсивності при кожному одиничному кроці по координаті x . Заповнення видимого інтервалу на скануючому рядку проводиться шляхом інтерполяції між значеннями інтенсивності на двох ребрах, що обмежують інтервал (рисунк 1).

Для кольорових об'єктів окремо інтерполюється кожен із компонентів кольору.

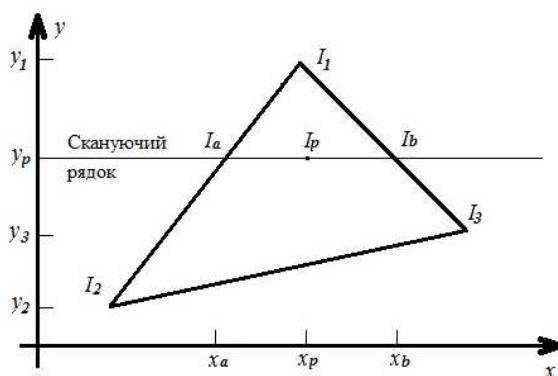


Рисунок 1 — Інтерполяція інтенсивності кольору

Більш перспективним вважається метод Фонга, в якому замість значень інтенсивності кольору інтерполюються вектори нормалей, які потім використовуються у функції тонування для обчислення інтенсивності кольору кожного елемента зображення [5]. Цей метод по відношенню до методу Гуро характеризується значно більшими обчислювальними витратами, однак при цьому досягається краща локальна апроксимація кривизни поверхні і, як наслідок, отримують більш реалістичні зображення.

Методи Гуро та Фонга є достатньо трудомісткими, оскільки для кожної точки поверхні об'єктів визначаються інтенсивності складових кольору точок зображення та їх екранні координати. При цьому використовуються трудомісткі операції.

Мета статті – зменшення обчислювальної складності процедури зафарбовування високополігональних поверхонь за методом Гуро та Фонга, що дасть змогу значно скоротити тривалість візуалізації складних проектів.

Розв'язання задачі та результати досліджень.

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

$$\Delta I = \frac{I_b - I_a}{\Delta x}, \quad (1)$$

де I_a , I_b – значення складових інтенсивності кольору відповідно в початковій та кінцевій точках PPT, який включає всі піксели між лівою та правою точками ребер трикутника в

одному із ортогональних напрямків; Δx – довжина рядка растеризації.

Аналогічно, за методом Фонга, прирости векторів уздовж РРТ трикутника визначаються за формулою

$$\Delta \vec{N} = \frac{\vec{N}_b - \vec{N}_a}{\Delta x},$$

де $\vec{N}_a, \vec{N}_b$ – нормалі полігону відповідно в початковій та кінцевій точках РРТ; Δx – довжина рядка растеризації

Наведені формули аналогічні, тому запишемо їх у такому вигляді

$$A = \frac{a_2 - a_1}{x_2 - x_1}.$$

Остання формула включає трудомістку операцію ділення, яка, безумовно, суттєво впливає на час зафарбовування. Це пояснюється тим, що операція ділення виконується для всіх ребер трикутника і рядків растеризації.

Для середньостатистичного трикутника, який включає 100 точок, кількість пікселів у рядку растеризації трикутника, як правило, не перевищує 16 [5].

Однією з властивостей двійкової системи числення є те, що операція ділення на 2^n – це зсув операнда на n розрядів у сторону молодших. Використовуючи цю властивість, можна замінити операцію ділення на певну кількість зсувів і нагромаджувальних додавань. Зрозуміло, що точність такої процедури залежить від кількості проведених операцій зсуву.

Для реалізації операції ділення можна запропонувати таку апроксимаційну формулу

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

де k – довжина рядка растеризації, n – степінь двійки (кількість операцій зсуву), i_j – коефіцієнт, що приймає значення «0» або «1» залежно від наявності степеневого доданку в апроксимаційній формулі.

Розглянемо використання формули (2) для конкретного випадку. Нехай $k=7$ та $n=6$:

$$\frac{1}{7} \approx \frac{1}{8} + \frac{1}{64}. \quad (3)$$

При інтерполяції інтенсивності кольору, згідно формули (1), максимальне значення приросту інтенсивності кольору для режиму TrueColor буде мати при $I_a=0$, $I_b=255$. Тобто

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

При використанні апроксимаційної формули (3)

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

Абсолютна похибка обчислень, для випадку, що розглядається, становить

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

Відносна похибка обчислень дорівнює

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

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

Апроксимаційні формули для виконання операції ділення з довжиною рядків растеризації до 16 точок наведено у таблиці 1.

Таблиця 1 — Апроксимаційні формули для $k=16$ і $n=8$

k	Апроксимаційна формула	$\Delta_{\max}$	$\delta_{\%}$
2	$1/2$	0	0
3	$\frac{1}{3} \approx \frac{1}{4} + \frac{1}{16} + \frac{1}{64} + \frac{1}{256}$	-0,3320	-0,39
4	$1/4$	0	0
5	$\frac{1}{5} \approx \frac{1}{8} + \frac{1}{16} + \frac{1}{128} + \frac{1}{256}$	-0,1992	-0,39
6	$\frac{1}{6} \approx \frac{1}{8} + \frac{1}{32} + \frac{1}{128} + \frac{1}{256}$	0,3320	0,78
7	$\frac{1}{7} \approx \frac{1}{8} + \frac{1}{64} + \frac{1}{256}$	0,4269	1,17
8	$1/8$	0	0
9	$\frac{1}{9} \approx \frac{1}{16} + \frac{1}{32} + \frac{1}{64}$	-0,4427	-1,56
10	$\frac{1}{10} \approx \frac{1}{16} + \frac{1}{32} + \frac{1}{128}$	0,3984	1,56
11	$\frac{1}{11} \approx \frac{1}{16} + \frac{1}{64} + \frac{1}{128} + \frac{1}{256}$	-0,2716	-1,17
12	$\frac{1}{12} \approx \frac{1}{16} + \frac{1}{64} + \frac{1}{256}$	-0,3320	-1,56
13	$\frac{1}{13} \approx \frac{1}{16} + \frac{1}{64}$	0,3065	1,56
14	$\frac{1}{14} \approx \frac{1}{16} + \frac{1}{128}$	-0,2846	-1,56
15	$\frac{1}{15} \approx \frac{1}{16} + \frac{1}{256}$	-0,0664	-0,39
16	$1/16$	0	0

При застосуванні апроксимаційних формул, що враховують 8 операцій зсуву, максимальна абсолютна похибка обчислень становить $\Delta_{\max} = 0,4427$, а середня відносна похибка – $\delta_{\%} = 0,81\%$.

На рисунку 2 наведено графік відносних похибок для $n=6$, $n=7$ та $n=8$.

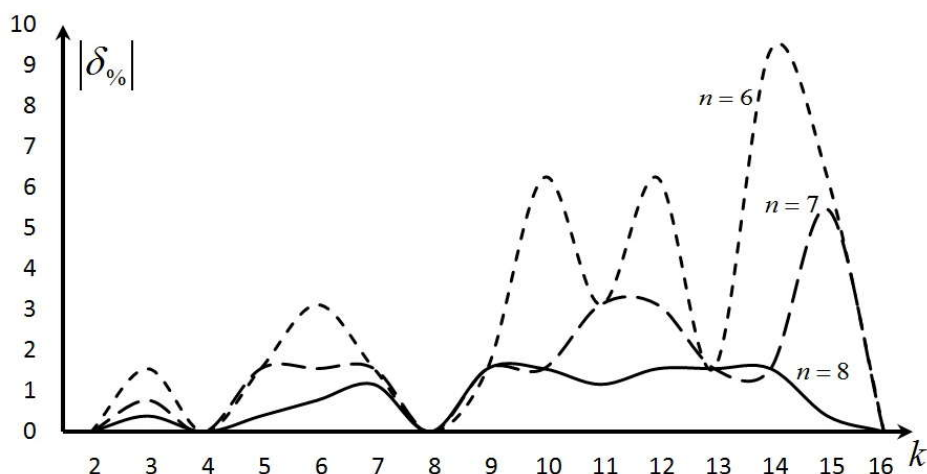


Рисунок 2 — Графік відносних похибок формул апроксимації

Проаналізуємо трудомісткість процедури зафарбовування за методом Гуро для середньостатистичного трикутника, який включає 100 точок [5].

Використаємо для розрахунків трикутник ABC з $\Delta x_{AB}=7$, $\Delta y_{AB}=8$, $\Delta x_{AC}=9$, $\Delta y_{AC}=12$, $\Delta x_{BC}=16$, $\Delta y_{BC}=4$.

Для оцінки трудомісткості виконання процедури зафарбовування трикутника за методом Гуро використано кількість тактів процесора Pentium M виконання складових інструкцій [2]. Інструкція задає одну окрему операцію процесора, визначену системою його команд [2]. У таблиці 2 наведено кількість необхідних для розрахунку операцій відповідно до кожного етапу зафарбовування.

Таблиця 2 — Склад і кількість інструкцій для зафарбовування середньостатистичного трикутника за методом Гуро

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

Структурну схему пристрою для виконання операції ділення за формулою (2), зображено на рисунку 3.

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

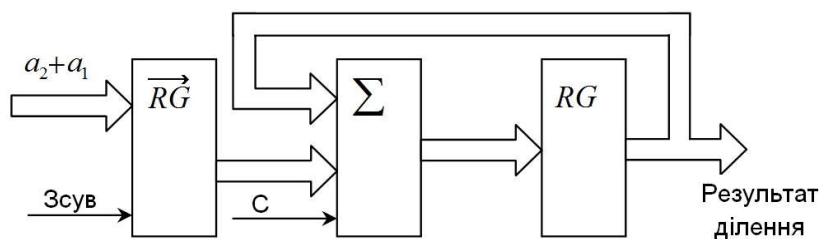


Рисунок 3 — Пристрій, що виконує операцію ділення за апроксимаційними формулами

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

Висновки.

У статті запропоновано підхід до підвищення швидкодії зафарбовування шляхом заміни складної операції ділення апроксимаційною формулою, яка передбачає певну кількість зсувів і нагромаджувальних додавань. При реалізації запропонованого підходу у методі Гуро досягається підвищення швидкодії в 1,43 рази.

Література

1. Адамс Дж. Direct: продвинутая анимация / Дж. Адамс.- М.: Кудиц-образ, 2004. - 480 с.
2. Оптимизация для процессоров семейства Pentium [Електронний ресурс] / А. Фог — Режим доступу: <http://megalib.com/books/914/CHMFirstPage.htm>.
3. Пэрент Р. Компьютерная анимация. Теория и алгоритмы: пер. с англ. / Рик Пэрент — М. : Кудиц-образ, 2004. — 560 с.
4. Рейнбоу В. Компьютерная графика. Энциклопедия / В. Рейнбоу. — С-Пб: Питер, 2003. — 196 с.
5. Farin G. Mathematical Principles for Scientific Computing and Visualization / G. Farin, D. Hansford . — Peters, Ltd. 2008. — 275 p.
6. Fosner R. Real—Time Shader Programming / Ron Fosner. —The Morgan Kaufmann Series in Computer Graphics, 2002. — 406 p.

Надійшла до редакції:
11.01.2011

Рекомендовано до друку:
д-р техн. наук, проф. Воронцов О.Г.

Abstract

Romanyk O.N., Obidnyk M.D. One of approaches to increase the speed of coloring. Method is suggested for improvement of shading rate, in which as a result of application of new approximation formulas computational complexity, of the procedures of vectors determination as well as color intensity is considerably reduced.

Keywords: shading, normal vectors, rasterization, color intensity.

Аннотация

Романюк А.Н., Ободник Н.Д. Один из подходов к повышению быстродействия закраски. Предложен метод повышения быстродействия закраски, в котором за счет использования новых аппроксимационных формул существенно уменьшена вычислительная сложность процедур определения векторов, а также интенсивности цвета.

Ключевые слова: закраска, векторы нормалей, растеризация, интенсивность цвета.

© Романюк О.Н., Обідник М.Д., 2011