

УДК004.925.6

О.Н. Романюк, канд. техн. наук, доц., Вінниц.
нац. техн. ун-т

ПАРАЛЕЛЬНА СТРУКТУРА ДЛЯ РЕАЛІЗАЦІЇ ЗАФАРБОВУВАННЯ В СИСТЕМАХ КОМП'ЮТЕРНОЇ ГРАФІКИ

А.Н. Романюк. Паралельная структура для реализации закраски в системах компьютерной графики. Предлагается структура для параллельного вычисления интенсивностей цвета сразу нескольких точек в строке растеризации при реализации закраски по методу Гуро.

A.N. Romanyuk. Parallel structure development for shading realization in computer graphic system. The structure of the several points parallel computations in rasterization line for the Gouraud shading method is proposed.

Підвищення ефективності обробки інформації є актуальним завданням комп'ютерної графіки. Вимоги до реалістичності формування тривимірних зображень постійно зростають, що обумовлює необхідність підвищення продуктивності засобів комп'ютерної графіки. Для відтворення реалістичних сцен необхідно зафарбовувати біля 500 тис. трикутників із частотою, як мінімум, 20...25 кадрів у секунду. Очевидно, що сучасні системи комп'ютерної графіки не завжди в змозі забезпечити необхідну продуктивність. У зв'язку з цим постає питання розробки структурованих підходів, які б забезпечували формування тривимірних реалістичних сцен у реальному часі при достатньо високій реалістичності відтворення всіх складових графічного об'єкта.

Однією з основних та найбільш трудомістких процедур рендерингу є процедура зафарбовування, згідно з якою для кожної точки поверхні визначається інтенсивність кольору та екранні координати [1, 2]. При визначенні інтенсивності кольору точок зображення враховують розташування джерела світла та спостерігача, оптичні властивості матеріалу, спектральні характеристики джерела світла, кривизну поверхні [2, 3]. У зв'язку з цим реалізація зафарбовування характеризується значними обчислювальними витратами, що обумовлює необхідність розробки високопродуктивних методів, алгоритмів і структур.

У методі Гуро, який отримав найбільше поширення в графічних акселераторах, розраховуються значення інтенсивностей для полігональних вершин, які потім у процесі растрування лінійно інтерполюються вздовж ребер і рядків сканування [3, 4]. Алгоритмічна та схемотехнічна простота методу Гуро спонукає науковців продовжувати роботи з удосконалення цього підходу. Суттєвого підвищення продуктивності методу Гуро можна досягти за рахунок розпаралелення обчислювального процесу [1]. При цьому важливо досягти збалансованого завантаження складових зафарбовщиків.

Розроблено два методи розпаралелення рендерингу Гуро, у яких використано властивість сталості приросту інтенсивності кольору вздовж рядків растрування, які визначають напрямки адресації внутрішніх точок трикутника при його зафарбовуванні [6]. Перший метод зафарбовування полягає у незалежному визначенні інтенсивностей кольору в парних і непарних точках рядка растрування між лівим і правим ребром трикутника, що забезпечує підвищення продуктивності зафарбовування до двох разів (рис. 1, а) [1]. Другий метод базується на розподілі обчислювальних дій процедури зафарбовування на нижньому рівні, коли графічному процесору задано параметри зіставних трикутників, а також інтенсивності кольору в їх вершинах [6]. Зафарбовування області, обмеженої трикутником, проводиться з використанням триангуляції Серпінського першого порядку. Згідно з методом трикутник розбивається на чотири зіставних шляхом з'єднання середин його сторін. У результаті отримують чотири трикутники, рівні за площею (рис. 1, б). Зіставні трикутники зафарбовують паралельно. При цьому використовується тільки один зафарбовщик, який тонує один із трикутників, а результати його роботи переносяться на всі інші трикутники з певним зміщенням за координатами і відповідними трансфор-

маціями інтенсивностей кольору точок. За рахунок розпаралелення обчислювального процесу вихідний трикутник буде зафарбовуватись в чотири рази швидше.



Рис. 1. Принципи розпаралелення

Сучасні графічні акселератори потребують ще більшого підвищення продуктивності зафарбовування, що обумовлює актуальність пошуку нових підходів і апаратних рішень. Тому розробка структур для розпаралелення обчислювального процесу при зафарбовуванні згідно з методом Гуро, які забезпечують формування цілого сегмента пікселів у рядку растрівання набуває особливого значення.

Пропонується підхід, який дозволяє підвищити швидкодію процедури зафарбовування за рахунок одночасного визначення інтенсивностей кольору цілого сегмента точок у рядку растрівання трикутників, на які попередньо триангулювався об'єкт. При реалізації сегментного зафарбування використана властивість, яка полягає в сталості приросту інтенсивності кольору трикутника вздовж рядка растрівання між точками його лівого і правого ребер [4]. Це дозволяє визначити інтенсивність i -тої точки відносно початку сегмента за формулою:

$$I_j = I_0 + i\Delta I_\Gamma,$$

де I_0 — інтенсивність першої точки в сегменті,

ΔI_Γ — приріст інтенсивності вздовж рядка растрівання.

Визначається оптимальний розмір сегмента, за умови, що він повинен бути кратним степені двійки.

Значення $i \cdot \Delta I_\Gamma$ найпростіше отримати, використовуючи блоки множення, однак такий підхід характеризується великими апаратними витратами. Більш доцільно знайти значення $i \cdot \Delta I_\Gamma$ з використанням операцій додавання та віднімання операндів типу $2^w \Delta I_\Gamma$, які легко отримати зсувом ΔI_Γ у бік старших розрядів.

Нехай p — розрядність сегмента. $\log_2 p$ зсувів приросту ΔI_Γ інтенсивності кольору виконується в бік старших розрядів. Ураховуючи, що операнд ΔI_Γ є вихідним, кількість операндів типу $2^w \Delta I_\Gamma$ буде дорівнювати $k = \log_2 p + 1$, де $w < k$.

Визначається, скільки різних приростів інтенсивностей кольору можна отримати з використанням мікрооперацій додавання та віднімання при умові, що операнди $2^w \Delta I_\Gamma$ відомі.

Кількість комбінацій двох чисел знаходиться за формулою

$$C_k^2 = \frac{k!}{2!(k-2)!}.$$

При визначенні різних значень приростів інтенсивностей кольору шляхом додавання операндів $2^w \Delta I_\Gamma$ операнд $\Delta I_\Gamma 2^{\log_2 p}$ не використовується, тому кількість різних значень приростів інтенсивностей, які можна отримати мікрооперацією додавання, дорівнює

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

Ураховуючи, що $2^{j-1} + 2^{j-2} = 2^j - 2^{j-2}$ і $2^{j-1} = 2^j - 2^{j-1}$, з використанням мікрооперації віднімання можна отримати ще

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

значень приростів інтенсивностей кольору.

Таким чином, з використанням операцій додавання та віднімання операндів типу $2^w \cdot \Delta I_{\Gamma}$ можна отримати

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

значень приростів інтенсивностей кольору.

Є певні наведені співвідношення між розміром сегмента та кількістю приростів інтенсивності кольору, що можуть бути отримані при застосуванні операції зсуву, додавання і віднімання (див. таблицю).

Співвідношення між розміром сегмента та кількістю приростів інтенсивностей кольору

Розрядність сегмента	Кількість приростів інтенсивностей кольору
2	2
4	4
8	8
16	14
32	22
64	32

Для прикладу можна розглянути зафарбовування трикутника при умові, що $p=8$. Цей випадок доцільно розглянути більш детально, оскільки розрядність сучасних мікросхем відеопам'яті мають саме таке значення.

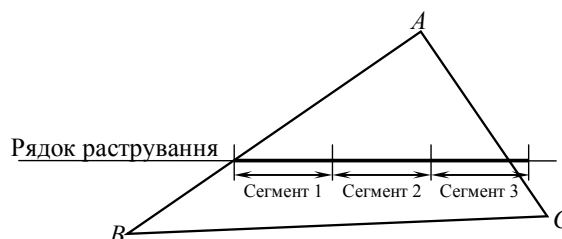


Рис. 2. Приклад розбиття рядка растрівання на сегменти

Початок першого сегмента збігається з точкою лівого ребра трикутника на його рядку растрівання (рис. 1). Кількість сегментів дорівнює $\lceil m/8 \rceil$, де m — кількість точок у рядку растрівання між лівим та правим ребром трикутника.

При визначенні інтенсивностей кольору точок поточного сегмента доцільно знайти і інтенсивність початкової точки наступного сегмента, оскільки стратегія, за якої кінцева точка одного сегмента є початковою точкою наступного, призведе до того, що розмір сегмента не буде кратним степені двійки.

Для знаходження інтенсивностей точок сегмента необхідно визначити значення приростів інтенсивностей $2\Delta I_{\Gamma} \dots 8\Delta I_{\Gamma}$.

Функціональна схема блока формування приростів інтенсивностей (БФПІ) кольору, який включає регістр для зберігання приросту інтенсивності ΔI_{Γ} і чотири суматори (рис. 3). На виході регістра монтажним зсувом отримують три значення приростів $2\Delta I_{\Gamma}$, $4\Delta I_{\Gamma}$, $8\Delta I_{\Gamma}$. На ви-

ходах суматорів і регістра БФП отримують всі необхідні значення приростів інтенсивності. Слід відмітити, що суматор Σ_4 працює в режимі віднімання чисел в доповняльному коді.

Значення приростів інтенсивностей подається на блок формування інтенсивностей кольору точок сегмента, де формуються інтенсивності кольору всіх точок поточного сегменту та інтенсивність кольору першої точки наступного сегмента (рис. 4).

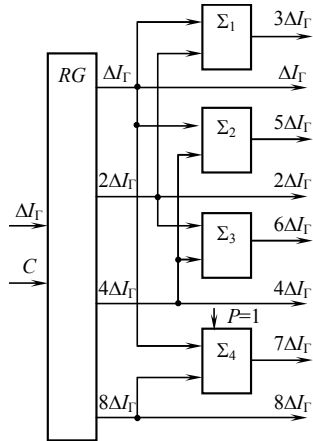


Рис. 3. Функціональна схема блока формування приростів інтенсивностей кольору

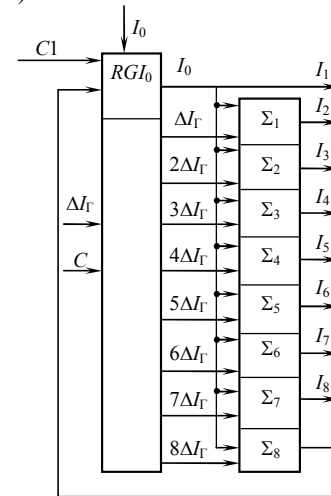


Рис. 4. Функціональна схема блока формування інтенсивностей кольорів точок сегмента

Після формування поточного сегмента значення інтенсивності кольору з виходу Σ_8 заноситься в регістр і використовується для визначення інтенсивностей кольору наступного сегмента.

Довжина рядка растрування трикутника між точками лівого і правого його ребра може бути не кратною довжині сегмента, тобто кінцева точка останнього сегмента може не збігатися з кінцевою точкою рядка растрування трикутника (приклад формування третього сегмента — див. рисунок 2). Для блокування запису точок останнього сегмента, що не належать рядку області трикутника, додатково вводиться регістр маски (значення 1 в ньому буде свідчити про те, що запис інтенсивності кольору точки сегмента дозволяється у відеопам'яті). Три молодших розряди двійкового представлення m вказує на номер розряду в регістрі маски, з якого необхідно записати нулі для блокування запису значень інтенсивностей кольору тих точок сегмента, які не належать трикутнику.

Для реалізації запропонованого підходу доцільно використати відеопам'ять з блоковою організацією запису/зчитування, наприклад, коли комірки відеопам'яті жорстко прив'язані до екранної площини.

Оскільки блоки відеопам'яті прив'язані до екранної площини, а сегменти безпосередньо до рядка растрування трикутника, то вони в загальному випадку не збігаються.

При запису сегмента в відеопам'ять необхідно визначити номер блока в відеопам'яті і здійснити зсув сегмента відносно початку блока. Нехай x — абсциса початку рядка растрування, тоді ціла частина від ділення x на вісім буде вказувати на номер блока у відеопам'яті, а три молодші розряди x — на кількість зсувів сегмента. Відповідно одна частина сегмента запишеться в вибраний блок, а інша частина — в наступний.

Вказана процедура вимагає додаткових апаратних витрат для “склеювання” частин сегментів. У зв'язку з цим, для сегментної реалізації зафарбовування більш доцільно використати відеопам'ять, у якій передбачено прискорення блоків до вибраної точки екранної площини. В цьому випадку відпадає необхідність розділення сегмента на складові.

Запропонована структура дозволяє суттєво підвищити продуктивність зафарбовування і може бути рекомендована для впровадження у високопродуктивних графічних системах формування реалістичних зображень.

Література

1. Романюк О.Н. Розподілення обчислювального процесу при реалізації процедур рендерингу / Романюк О.Н., Чорний А.В., Рудомін А.Г. // Вісн. Кременчуц. держ. політехн. ун-ту. — 2003. — № 4. — С. 14 — 18.
2. Роджерс Д. Алгоритмические основы машинной графики: Пер. с англ. — М.: Мир, 1989. — 512 с.
3. Фоли Дж. Основы интерактивной машинной графики. / Фоли Дж., Ван Дем А. В 2 кн. — М.: Мир, 1985. — 384 с.
4. Gouraud H. Continuous shading of curved surfaces // IEEE Trans. on Comp. — 1971. — № 6, June. — P. 623 — 628.
5. Романюк О.Н. Один із шляхів спрощення обчислювального процесу при зафарбовуванні тривимірних об'єктів за методом Гуро // Вимірюв. та обчисл. техніка в технол. процесах. — 2004. — № 2. — С. 72 — 75.
6. Романюк О.Н. Організація обчислювального процесу при розпаралеленні рендерингу Гуро / Романюк О.Н., Чорний А.В. // Вимірюв. та обчисл. техніка в технол. процесах: Зб. наук. пр. — 2000. — С. 105 — 109.

Надійшла до редакції 12 березня 2007 р.
