

АНАЛІЗ МЕТОДІВ ЗАФАРБОВУВАННЯ ПОВЕРХОНЬ ТРИВИМІРНИХ ОБ'ЄКТІВ

Романюк О. Н., д.т.н., E-mail: rom8591@gmail.com

Отришко В.О., магістрант, E-mail: wolf1702@ukr.net

Нечипорук М.Л., студент, E-mail: Kolia6@list.ru

Генерація 3D-зображень є складною обчислювальною задачею, в зв'язку з чим виконують її декомпозицію [1, 2]. У більшості графічних систем складові поверхні об'єктів апроксимують мережею трикутників, які після трансформацій зі світової системи растеризують в екранній системі координат.

Найпоширеніше розбиття зображень на трикутники пояснюється такими причинами [1]: трикутник є найпростішим полігоном, вершини якого однозначно задають грань; будь-яку поверхню можна гарантовано розбити на трикутники; обчислювальна складність алгоритмів розбиття поверхні на трикутники істотно менша порівняно з використанням інших полігонів; реалізація процедур рендерингу найпростіша для області, обмеженої трикутником; для трикутника легко визначити трьох його найближчих сусідів, що мають із ним спільні грані; видові перетворення трикутника суттєво простіші порівняно з криволінійними об'єктами; процедура текстурування трикутників найбільш проста. Полігональна модель підтримується стандартами Direct3D і OpenGL.

Історично першим методом зафарбовування був метод [1,2] однотонного зафарбовування, згідно з яким для кожного плоского трикутника визначався вектор нормалі, а на його основі – колір. Складові трикутники об'єкта заповнювалися одним кольором без його градації, що призводило до різкої зміни інтенсивності кольору на межах трикутників. Однотонне зафарбовування вимагає найменших обчислювальних затрат. Воно хоча й має низьку якість, але внесено до функцій Direct3D і підтримується тривимірними акселераторами.

До найпоширеніших методів зафарбовування відносять метод Гуро [1, 3, 4, 5], який забезпечує прийнятний компроміс між швидкодією формування тривимірних зображень та їх якістю. Процес зафарбовування має такі стадії : а) розраховують вектори нормалей до кожної грані; б) шляхом усереднення нормалей усіх граней, яким належить вершина, розраховують нормалі у вершинах трикутника (багатокутника); с) визначають інтенсивності кольору у вершинах багатокутника, використовуючи значення нормалей; д) зафарбовують ділянку, обмежену багатокутником, шляхом лінійної інтерполяції інтенсивностей кольору вздовж ребер, а потім і між ребрами вздовж кожного рядка растеризації. Останнім часом затінення за Гуро часто використовують як проміжну стадію з інтерактивного формування 3D-зображення, покладаючи побудову повноцінної сцени на етап фінального рендерингу.

У більшості випадків для реалізації зафарбовування за методом Гуро знаходять прирости інтенсивності кольору вздовж ребер трикутника і потім між їх точками, які мають однакову ординату (абсцису). Суттєвий недолік такого

підходу полягає в необхідності визначення приросту інтенсивностей кольору для кожного рядка растеризації.

Апаратне визначення інтенсивностей кольору вздовж ребер і рядків растеризації трикутника передбачає використання кодових лінійних інтерполяторів за умови, що прирости інтенсивностей кольору визначено на програмному рівні.

Широке впровадження методу Гуро в графічних акселераторах пов'язано з розробкою фірмою ATI технології TrueForm [6], яка активно підтримувалася графічними відеокартами. Суть технології полягає в додатковій триангуляції трикутника, для чого плоский трикутник трансформують у поверхню третього порядку, а потім – зворотно в набір трикутників (патч). У найпоширенішому випадку трикутник розбивають на дев'ять складових трикутників. Для цього через вершини трикутника проводять три криві Без'є [4] кожна з яких заміняється на ламану. Це дозволяє визначити по дві додаткові вершини для кожного ребра трикутника. За значеннями координат шести додаткових точок визначають ще одну точку в середині трикутника.

Технологія TrueForm дозволяє підвищити реалістичність 3D-зображень і не вимагає модифікації підсистеми рендерингу відеокарти. Додаткове триангуляція підвищує деталізацією поверхні і її згладженість, але призводить до суттєвого збільшення тривалості циклу підготування до зафарбовування.

До недоліків методу Гуро можна віднести: а) метод використовує для визначення інтенсивностей кольору лінійну інтерполяцію, в той час як дифузна та спекулярна складові кольору мають нелінійний характер зміни; б) не враховується локальна кривизна поверхні, оскільки вектори нормалей визначаються тільки для вершин трикутника; в) відблиски відтворюються тільки в разі, якщо вершини трикутників знаходяться в їх зоні (відблиск, який не має спільних точок із вершинами трикутників, або розташований усередині трикутника, не буде сформовано); г) на межах двох трикутників проявляються смуги Маха, які пов'язані з літеральним гальмуванням на сітківці ока; д) має місце зміна інтенсивності кольору зображення від кадру до кадру, що виражається в миготінні, особливо відблисків, оскільки при формуванні динамічних зображень змінюється структура та положення вузлів триангуляційної мережі; е) наявність артефакту типу “зірка”, який полягає у тому, що відблиск, який повинен мати форму еліпса, має форму зірки. Це пояснюється тим, що ділянки відблиску формуються в різних трикутниках і проявляється ефект смуг Маха; ж) метод не враховує перспективу об'єкта..

На сучасному етапі розвитку комп'ютерної графіки гостро стоїть питання про підвищення реалістичності відтворення 3D-зображень [1, 2]. У зв'язку з цим має місце тенденція використання більш складних моделей освітлення та методів зафарбовування. Хоча метод Гуро й продовжує широко використовуватися для формування тривимірних зображень у реальному часі, однак більш перспективним вважається метод Фонга [1, 2, 4, 7, 8], у якому замість значень інтенсивності кольору інтерполюються вектори нормалей, які потім використовуються у функції тонування для обчислення інтенсивності кольору кожного елемента зображення. Метод характеризується, порівняно з методом Гуро, значно більшими обчислювальними витратами, однак при цьому

досягається краща локальна апроксимація кривизни поверхні і, як наслідок, отримують більш реалістичні зображення.

У більшості випадків при зафарбовуванні за Фонгом використовуються моделі освітлення Бліна та Фонга [1, 7, 9]. Це пояснюється тим, що вказані ДФВЗ є похідними від косинуса кута між векторами, які легко знайти через їх скалярний добуток. Коли джерело світла та спостерігач розташовані на достатньо великій відстані від об'єкта (найпоширеніший випадок у комп'ютерній графіці), вектор \vec{H} для моделі Бліна розраховується один раз для кожного трикутника. У зв'язку з цим у більшості випадків для зафарбовування поверхонь за методом Фонга використовується модель освітлення Бліна.

У реалізації методу Фонга за методом Д. Бішопа і Е. Вейнера [10] інтенсивність кольору розраховується з використанням розкладу в ряд Тейлора. Вектор нормалі до поверхні в точці (x, y) знаходять за формулою

$$\vec{N}(x, y) = \vec{A}x + \vec{B}y + \vec{C}$$

Вектори $\vec{A}, \vec{B}, \vec{C}$ можна визначити з трьох рівнянь для вершин. Для дифузного відбиття $I_d = (\vec{L} \cdot \vec{N}) / (|\vec{L}| \cdot |\vec{N}|) = (\vec{L}(\vec{A}x + \vec{B}y + \vec{C})) / (|\vec{L}| \cdot |\vec{A}x + \vec{B}y + \vec{C}|)$.

Останній вираз можна записати в такій формі

$$I_d(x, y) = (ax + by + c) / \sqrt{dx^2 + exy + fy^2 + gx + hy + i}$$

Параметри $a, b, c, d, e, f, g, h, i$ є скалярними величинами, наприклад, $a = (\vec{L} \cdot \vec{A}) / |\vec{L}|$. Якщо знаменник у виразі $I_d(x, y)$ розкласти в ряд Тейлора і залишити члени до другої степені, то можна отримати таке рівняння $I(x, y)_d = T_3x^2 + T_4xy + T_3y^2 + T_2x + T_1y + T_0$, де всі T_k є функціями від $a, b, c, d, e, f, g, h, i$. Якщо використати праві різниці, то для розрахунку поточного значення $I(x, y)_d$ необхідно виконати тільки операції додавання. Аналогічно розраховується й спекулярна складова кольору.

Метод дозволяє зменшити кількість операцій для обчислення інтенсивності кольору пікселя до п'яти додавань і одного доступу до пам'яті, однак додатково необхідно обчислювати ряд Тейлора, що вимагає близько 500 операцій із плаваючою комою на вершину полігона, а це робить метод занадто повільним.

А. Хаст [11] запропонував метод зафарбовування, використавши для знаходження інтенсивностей кольору метод Ейлера і метод випереджувальних різниць. Для спрощення розрахунку інтенсивності кольору було запропоновано лінійно інтерполювати значення похідних між початковою та кінцевою точками рядка растеризації. Оскільки при обчисленні за формулою Ейлера мають місце похибки і при лінійній інтерполяції похідних знаходяться наближені її значення, то значення інтенсивностей у кінцевій точці рядка растеризації може суттєво відрізнятись від дійсного значення. Один із підходів до усунення такого артефакту полягає у визначенні точного значення інтенсивності кольору в кінцевій точці рядка растеризації трикутника, визначенні абсолютної похибки і її розподілу на точки рядка растеризації.

У роботі [1] розроблено метод для обчислення спекулярної складової кольору, у якому використовують степеневу та логарифмічну функції.

Запропоновано таблично-алгоритмічну реалізацію зафарбовування, яка, на жаль, при апаратній реалізації не забезпечила прийнятних за якістю результатів, оскільки мають місце візуальні відмінності порівняно з методом Фонга.

У роботах [13, 14] розглянуто питання використання для зафарбовування методу «цифра за цифрою». Найчастіше метод використовують для визначення векторів із використанням сферично-кутової інтерполяції. Оскільки метод «цифра за цифрою» є послідовним, то це обмежує його швидкодію.

При зафарбовуванні 3D-об'єкта визначають нормовані вектори до поверхні об'єкта, джерела світла й спостерігача, а також допоміжні вектори залежно від вибору моделі освітлення. Нормалізація вектора потребує виконання трьох операцій ділення, трьох операцій множення, двох операцій додавання та операцію знаходження квадратного кореня, виконання яких достатньо трудомістке. У зв'язку з цим актуальним питанням є спрощення процедури нормалізації з метою її апаратної реалізації.

Науковцями фірми VIDIA було отримано формулу [14] для наближеної нормалізації векторів із використанням одного кроку ітерації Ньютон-Рафсона: $\vec{N}_n \approx \vec{N} \cdot (3 - \vec{N} \cdot \vec{N}) / 3$. Хоча наведена формула достатньо проста, однак великі похибки визначення ортогональних складових вектора обмежують її використання для задач зафарбовування.

Р. Ліон [15], використавши розклад у ряд Тейлора виразу $1/\sqrt{\vec{N} \cdot \vec{N}}$, отримав формулу для наближеної нормалізації вектора нормалі

$$\vec{N}_n \approx \vec{N} \left(1 - \frac{1}{2} ((\vec{N} \cdot \vec{N}) - 1) + \frac{3}{8} ((\vec{N} \cdot \vec{N}) - 1)^2 \right),$$

у якій операції ділення замінено на зсув, що спрощує апаратну реалізацію. На жаль, формула має велику похибку апроксимації, що обмежує її застосування.

Інтерполяцію одиничних векторів між початковим \vec{N}_a та кінцевим \vec{N}_b векторами можна виконати за формулою [1]

$$\vec{N}(w) = \vec{N}_a \frac{\sin((1-w)\psi)}{\sin\psi} + \vec{N}_b \frac{\sin(w\psi)}{\sin\psi}, \quad (1.2)$$

де $w \in [0, 1]$, а ψ – кут між векторами \vec{N}_a та \vec{N}_b . Розрахунок векторів передбачає визначення синуса, ресурсоємної функції арккосинуса для знаходження невідомих w та ψ , а також виконання операції ділення.

Достатньо трудомісткою є процедура обчислення ДФВЗ. У системах реального часу в більшості випадків використовують функцію $\cos^n \gamma$, де n змінюється в діапазоні від 1 до 1000 [1, 2]. Одним із підходів до розрахунку ДФВЗ є таблична реалізація, яка передбачає зберігання значень функцій в ПЗП. Для зберігання значень функції $\cos^n \gamma$ ($n = 1, 255$) необхідно використовувати двовимірні таблиці загальним обсягом більшим 4 Мбайтів [16]. Для зменшення обсягів таблиць у роботі [16] пропонується використовувати нерівномірне квантування $\cos^n \gamma$, а також спеціалізований процесор.

Серед існуючих методів тільки в методі Гуро та однотонного зафарбовування в циклі визначення інтенсивностей складових кольору використовуються прості, з обчислювальної точки зору, операції. Це дозволяє їх реалізувати апаратно. Інші методи передбачають виконання трудомістких операцій та процедур і орієнтовані, в основному, на програмну реалізацію.

Перелік посилань

1. 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.
2. Романюк О. Н. Комп'ютерна графіка. Навчальний посібник / О. Н. Романюк — Вінниця: ВДГУ, 2001. — 129 с.
3. Романюк О. Н. Новий підхід до підвищення реалістичності зафарбовування тривимірних об'єктів за методом Гуро / О. Н. Романюк // Інформаційні технології та комп'ютерна інженерія. — 2005. — № 2. — С. 106—109.
4. Роджерс Д., Адамс Дж. Математические основы машинной графики / Д. Роджерс, Дж. Адамс : Пер. с англ. — М. : Мир, 2001. — 604 с.
5. Романюк О. Н. Один із шляхів спрощення обчислювального процесу при зафарбовуванні тривимірних графічних об'єктів за методом Гуро / О. Н. Романюк // Вимірювальна та обчислювальна техніка в технологічних процесах. — 2004. — № 2. — С. 72—75.
6. Vlachos A. Curved PN triangles / A. Vlachos, J. Peters, C. Boyd, J. Mitchell // In ACM Symposium on Interactive 3D Graphics 2001. — 2001. — P. 159—166.
7. Романюк О. Н. Альтернативна реалізація дистрибутивної двопроменевої функції для моделей освітлення Бліна та Фонга / О. Н. Романюк // Наукові праці Донецького національного технічного університету. Серія «Обчислювальна техніка та автоматизація». — 2006. — Випуск 106. — С. 219—228.
8. Романюк О. Н. Новий підхід до реалізації процедури зафарбовування за методом Фонга / О. Н. Романюк, А. В. Чорний // Вісник Херсонського державного технічного університету. — 2003. — Вип. 22. — С. 154—160.
9. Романюк О. Н. Класифікація дистрибутивних функцій відбивної здатності поверхні / О. Н. Романюк // Наукові праці Донецького національного технічного університету. — Серія «Інформатика, кібернетика і обчислювальна техніка». — 2008. — Випуск 9 (132). — С. 145—151.
10. Bishop G. Fast Phong shading / G. Bishop, D. M. Weimar // Computer Graphics. 4 April. — Vol. 20. — 1986. — P. 103—106.
11. Hast A. Approximated Phong Shading by using the Euler Method. / A. Hast, T. Barrera, E. Bengtsson // Eurographics 01(2001). — P. 43—48.
12. Ming-Hsiu Lai An efficient modified Phong shading algorithm & its low-complexity realization / Lai Ming-Hsiu, Yu Ming-Feng, Chen Sau-Gee // Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on Volume 4, Issue, 2002. Page(s) : IV-201 - IV-204, vol. 4.
13. Артамонов Е. И. Синтез структур специализированных средств машинной графики / Е. И. Артамонов, В. М. Хачумов. — М. : Ин-т проблем управления АН СССР, 1991. — 144 с.
14. Euh, J. Power-Aware Computer Graphics Rendering. / J. Euh, J. Chittamuru, W. Burleson // [The Journal of VLSI Signal Processing](#), 2005, [Volume 39, Numbers 1/2](#). — P. 15—33.
15. Lyon R.F. Phong Shading Reformulation for Hardware Renderer Simplification / R. F. Lyon. — Apple Technical Report № 43. — 1993. — 33 p.
16. Shin H. A Hardware Cost Minimized Fast Phong Shader / H. Shin, J. Lee, L. Kim // IEEE Transactions on VLSI System. — April 2003. vol. 9, no.2. — P. 297—304