УДК 004.925.4

О. В. Романюк, асп.

ВИСОКОПРОДУКТИВНИЙ МЕТОД ПЕРСПЕКТИВНО-КОРЕКТНОГО ТЕКСТУРУВАННЯ

Запропоновано метод перспективно-коректного текстурування, який не містить операцій ділення та передбачає нецілочислове задання текстурних координат. Метод дозволяє знаходити точні значення текстурних координат.

Вступ

Основна задача комп'ютерної графіки пов'язана з синтезом реалістичних зображень, необхідність формування яких виникає під час вирішення низки практичних задач у машинобудівному та архітектурному проектуванні, дизайні, мультиплікації, іграх, відеотренажерах, рекламі, медицині і т. д. З появою систем віртуальної реальності, в яких спостерігач занурюється в світ моделі, виникає потреба візуалізації віртуального середовища, максимально наближеного до того, що людина спостерігає в природних умовах.

Одним із способів забезпечення реалістичності графічної сцени є відтворення рельєфу та мілких деталей на поверхні об'єктів графічної сцени, що потребує великих обсягів обчислень. В той же час, для багатьох додатків необхідна дуже висока продуктивність формування графічних зображень, у зв'язку з цим задача підвищення продуктивності формування рельєфних зображень є досить актуальною.

Аналіз останніх досліджень та публікацій. Постановка задачі

Один із підходів до побудови високореалістичних зображень полягає у використанні текстур [1—5], які накладаються на графічні об'єкти. Використання текстур у багатьох випадках дозволяє успішно розв'язувати задачі, які надзвичайно трудомістко розв'язуються прямими методами. Текстурування дозволяє суттєво зменшити обчислювальні витрати та зробити можливим інтерактивний режим візуалізації.

У задачах текстурування встановлюється співвідношення між екранними координатами і текстурними координатами [2]. При афінному текстуруванні [3] текстурні координати лінійно інтерполюються вздовж рядка растеризації, що призводить до появи артефактів та недостовірного відтворення перспективи об'єкта.

При перспективно-коректному текстуруванні [1, 2, 4] використовують нелінійні функції, розрахунок яких передбачає попіксельне виконання трудомістких операцій. Перспективно-коректне текстурування у переважній більшості випадків реалізують за методом Хекберта [1]

$$u = \frac{ax + by + c}{gx + hy + i}, \quad v = \frac{dx + ey + f}{gx + hy + i}, \tag{1}$$

де u і v — текстурні координати, x і y — екранні координати об'єкта, $a \ldots i$ — коефіцієнти полігона, який текстурується.

Як видно з формули (1), знаходження текстурних координат є трудомісткою процедурою, оскільки для кожного піксела зображення потрібно виконати дві операції ділення, що суттєво позначається на швидкодії формування графічних сцен. З метою спрощення обчислювального процесу доцільно вилучити трудомісткі операції ділення з процесу перспективно-коректного текстурування.

Хоча методи лінійної [2] або квадратичної [3, 4] апроксимації гіперболічної кривої не використовують операції ділення, однак мають місце відносно великі похибки, які, безумовно, впливають на якість текстурування.

У роботі [5] для апроксимації гіперболічної кривої запропоновано використовувати метод середньої точки, який дозволяє знайти не наближені, а точні значення текстурних координат. У цьому випадку значення кожної текстурної координати перевіряється на відповідність певній умові для конкретних значень екранних координат об'єкта. Якщо текстурна координата відповідає умові, то вважається, що для заданих екранних координат об'єкта текстурна координата знайдена, якщо ні — то перевіряється наступне значення текстурної координати на відповідність. При цьому, замість операцій ділення використовуються операції додавання та порівняння, що значно зменшує обсяг обчислень при перспективно-коректному текстуруванні. Недоліком методу є припущення, що текстурні координати задані цілими числами, що складає тільки окремий випадок текстурування.

Згідно зі специфікацією OpenGL [6] текстурні координати задаються в діапазоні $[0;\,1]$, тобто є дробовими числами, що зумовлює необхідність розробки нового методу перспективно-коректного текстурування, який би враховував нецілочислове задання текстурних координат.

Метою роботи є розробка високопродуктивного методу перспективно-коректного текстурування, який не використовує операції ділення та передбачає нецілочислове задання текстурних координат.

Метод перспективно-коректного текстурування

Нехай потрібно знайти текстурні координати u і v. Оскільки координати u і v розраховуються аналогічно, то в подальшому будемо розглядати тільки координату u з узагальненням результатів і на координату v.

При перспективно-коректному текстуруванні за методом середньої точки розраховується послідовність текстурних координат u, які відповідають умові [5]

$$\frac{ax+by+c}{gx+hy+i}-0,5 \le u < \frac{ax+by+c}{gx+hy+i}+0,5\,. \tag{2}$$

Умова (2) прийнятна для випадку цілочислового задання текстурних координат.

У загальному випадку, коли значення текстурних координат лежать у діапазоні $[0;\,1]$, умова (2) набуде вигляду

$$\frac{ax+by+c}{gx+hy+i}-\frac{\Delta u}{2} \le u < \frac{ax+by+c}{gx+hy+i}+\frac{\Delta u}{2}, \tag{3}$$

де Δu — різниця між сусідніми значеннями текстурної координати u, яка знаходиться за формулою $\Delta u = 1/p$, p — кількість точок текстури по осі U.

Введемо позначення $t = 2(gx+hy+i)$. Помножимо нерівність (3) на t

$$2(ax+by+c)-\Delta u(gx+hy+i) \le ut < 2(ax+by+c)+\Delta u(gx+hy+i).$$

Запишемо отриману нерівність у вигляді системи нерівностей

$$\begin{cases} ut-2(ax+by+c)+\Delta u(gx+hy+i) \ge 0; \\ ut-2(ax+by+c)-\Delta u(gx+hy+i) < 0. \end{cases}$$

Після спрощення отримаємо

$$\begin{cases} ut-x(2a-\Delta ug)-y(2b-\Delta uh)-(2c-\Delta ui) \ge 0; \\ ut-x(2a+\Delta ug)-y(2b+\Delta uh)-(2c+\Delta ui) < 0. \end{cases} \tag{4}$$

Введемо такі позначення:

$$N(x,y,u) = ut-x(2a-\Delta ug)-y(2b-\Delta uh)-(2c-\Delta ui);$$

$$M(x,\,y,\,u) = ut-x(2a+\Delta ug)-y(2b+\Delta uh)-(2c+\Delta ui).$$

Тоді умови (4) матимуть вигляд:

$$\begin{cases} N(x,\,y,\,u) \ge 0; \\ M(x,\,y,\,u) < 0. \end{cases} \tag{5}$$

Система нерівностей (5) визначає умови, яким повинна відповідати текстурна координата u

для заданих значень екранних координат x і y. Якщо для поточного значення координати u умови (5) не справджуються, то до нього додається (віднімається) приріст Δu, і для нового значення перевіряються умови (5). Додавання (віднімання) приросту Δu відбувається до тих пір, поки умови (5) не будуть справджуватись. Коли знайдено значення текстурної координати u, яке задовольняє умовам (5), відбувається перехід до наступної точки рядка растеризації.

Розглянемо, як змінюватимуться $t(x, y)$, $N(x, y, u)$ і $M(x,y,u)$ зі зміною координат x і y на 1.

Зі збільшенням x на 1

$$t(x+1, y) = t(x, y) + 2g;$$

$$N(x+1, y, u) = N(x, y, u) + 2gu - (2a - \Delta ug);$$

$$M(x+1, y, u) = M(x, y, u) + 2gu - (2a + \Delta ug).$$

Якщо зменшити x на 1, то

$$t(x-1, y) = t(x, y) - 2g;$$

$$N(x-1, y, u) = N(x, y, u) - 2gu + (2a - \Delta ug);$$

$$M(x-1, y, u) = M(x, y, u) - 2gu + (2a + \Delta ug).$$

Зі збільшенням y на 1

$$t(x, y+1) = t(x, y) + 2h;$$

$$N(x, y+1, u) = N(x, y, u) + 2hu - (2b - \Delta uh);$$

$$M(x, y+1, u) = M(x, y, u) + 2hu - (2b + \Delta uh).$$

Вирази $(2a - \Delta ug)$, $(2a + \Delta ug)$, $(2b - \Delta uh)$ і $(2b + \Delta uh)$ достатньо розрахувати один раз для всього полігона. Тоді знаходження умов (5) для наступної точки рядка растеризації потребуватиме виконання лише двох операцій множення та чотирьох операцій додавання.

Якщо поточне значення текстурної координати не задовольняє умовам (5), то до нього потрібно додати чи відняти приріст Δu. Визначимо як змінюються $N(x,y,u)$ і $M(x,y,u)$ зі зміною текстурної координати u на Δu.

Зі збільшенням u на Δu

$$\begin{cases} N(x,y,u+\Delta u) = N(x,y,u) + \Delta u \cdot t(x,y); \\ M(x,y,u+\Delta u) = M(x,y,u) + \Delta u \cdot t(x,y); \end{cases} \tag{6}$$

Зі зменшенням u на Δu

$$\begin{cases} N(x,y,u-\Delta u) = N(x,y,u) - \Delta u \cdot t(x,y); \\ M(x,y,u-\Delta u) = M(x,y,u) - \Delta u \cdot t(x,y); \end{cases} \tag{7}$$

Вищенаведені формули дозволяють знайти значення умов $N(x,y,u)$ і $M(x,y,u)$ для нового значення координати u, виконавши лише дві операції множення та дві операції додавання.

З формул (6) і (7) видно, що вирази $N(x,y,u)$ і $M(x,y,u)$ збільшуються зі збільшенням u, і зменшуються зі зменшенням u, оскільки $\Delta u > 0$ і $t > 0$. Ця властивість дозволяє вибрати тип зміни координати u — збільшувати чи зменшувати її. Якщо $N(x, y, u) < 0$, то до поточного значення u потрібно додавати приріст Δu. Якщо $M(x, y, u) > 0$, то виконується віднімання приросту Δu від поточного значення координати u.

Аналогічним чином розраховується текстурна координата *v*.

Принцип дії запропонованого методу зображено на рисунку.

З рисунку видно, що для конкретного випадку знаходження координати *u* в точці x_i потрібно виконати 3 ітерації додавання приросту Δu, а в точці x_{i+1} — 2 ітерації, і т. д. Коли гіперболічна крива змінюється пологіше, то спостерігається зменшення кількості ітерацій додавання приростів Δu.

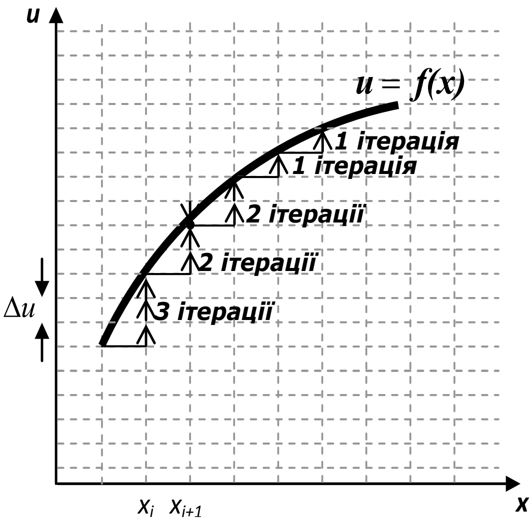

Рис. Адитивний підхід до апроксимації гіперболічної кривої

Висновки

У запропонованому методі перспективно-коректного текстурування операції ділення замінюються на операції додавання, що дозволяє суттєво зменшити обсяги обчислень і прискорити процес текстурування.

Крім того, метод дозволяє знаходити значення текстурних координат із заданою точністю, яка визначається інтервалом між сусідніми значеннями текстурних координат. На відміну від запропонованого методу, у класичному методі перспективно-коректного текстурування знайдене значення текстурної координати потрібно округляти або відкидати дробову частину, що призводить до похибок текстурування.

Порівняно з методом [5] запропонований метод є універсальним, оскільки може бути застосованим як для цілочислового, так і для дробового задання текстурних координат.

СПИСОК ЛІТЕРАТУРИ

1. Heckbert P. S. Interpolation for Polygon Texture Mapping and Shading / P. S. Heckbert, H. P. Moreton / State of the Art in Computer Graphics: Visualization and Modelling. — Springer-Verlag, 1991. — P. 101—111.

2. Hecker C. Perspective Texture Mapping Part IV : Approximations // Game Developer. — 1995, Vol. 2, No. 6. — P. 19—25.

3. Wolfgang Engel. Programming Vertex, Geometry, and Pixel Shaders Charles / Engel Wolfgang. — Charles River Media, 2008. — 494 p.

4. Ламот А. Программирование трехмерных игр для Windows. Советы профессионала по трехмерной графике и растеризации. — М. : ИД «Вильямс», 2004. — 1424 с.

5. Barenbrug B. Algorithms for division free perspective correct rendering / B. Barenbrug , F. J. Peters , C.W.A.M. van Overveld / Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware. — Interlaken, Switzerland, 2000. — P. 7—13.

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

Рекомендована кафедрою програмного забезпечення

Надійшла до редакції 31.02.10
Рекомендована до друку 23.03.10

Романюк Оксана Володимирівна — аспірантка кафедри програмного забезпечення.
Вінницький національний технічний університет