

УДК 004.925.4

## РОЗРОБКА МЕТОДІВ ТЕКСТУРУВАННЯ ДЛЯ ЗАДАЧ ФОТОРЕАЛІСТИЧНОГО РЕНДЕРИНГУ

Романюк О. Н., Дудник О. О.

Вінницький національний технічний університет

*Запропоновано методи для підвищення продуктивності та реалістичності текстурування, зокрема, методи перспективно-коректного текстурування, фільтрації текстур, рельєфного текстурування.*

При формуванні графічних зображень вирішується двоєдина задача – підвищення продуктивності та підвищення реалістичності. Сьогодні продуктивності графічних засобів достатньо для формування зображень, які за своїми візуальними властивостями близькі до фотографій, тобто досягається фотографічна якість.

Базовим підходом до формування фотореалістичних зображень є текстурування, яке полягає у накладанні на поверхню полігонів ділянок існуючих зображень (як правило з фотографій). Для визначення кольору пікселя на екрані встановлюється відповідність між екранним пікселем та точкою на текстурі.

При визначенні текстурних координат важливо врахувати деформацію полігонів при перспективних перетвореннях. Тому при перспективно-коректному текстуруванні, у переважній більшості випадків, текстурні координати визначають за методом Хекберта [1-4]:

$$\begin{aligned} u_i &= \frac{A_1 \cdot x_i + B_1 \cdot y_i + C_1}{D \cdot x_i + E \cdot y_i + F}, \\ v_i &= \frac{A_2 \cdot x_i + B_2 \cdot y_i + C_2}{D \cdot x_i + E \cdot y_i + F}. \end{aligned} \quad (1)$$

де  $u$  і  $v$  – текстурні координати,  $x$  і  $y$  – екранні координати об'єкта,  $A_1, A_2, B_1, B_2, C_1, C_2, D, E, F$  – коефіцієнти полігона, який текстурується. Як видно з формули (1), знаходження текстурних координат є трудомісткою процедурою, тому було знайдено ітеративні формули для зменшення кількості операцій.

Формула (1) досить складна для обчислення в реальному часі на кожен піксель. Розроблено алгоритм паралельного підрахунку координат текселів: для кожного рядка растеризації спочатку обчислюються параметри  $u_t$  і  $u_b$ , після чого паралельно обчислюються значення чисельника за формулою [2]:

$$w_n = u_t + A_1 \cdot n,$$

і знаменника за формулою

$$v_n = u_b + D \cdot n,$$

після чого виконується операція ділення для визначення координати:

$$u_n = \frac{w_n}{v_n}.$$

Оскільки для кожного наступного  $x$ ,  $n$  збільшується на 1, має місце така формула [1]:

$$w_n = w_{n-1} + A_1, \quad v_n = v_{n-1} + D. \quad (2)$$

Таким чином, відповідно до формул (2) можна запропонувати послідовну процедуру обчислення текстурних координат:  $w_n$  і  $v_n$  для кожного  $x$  обчислюються шляхом додавання до  $w_{n-1}$  і  $v_{n-1}$ , визначених для  $x-1$ ,  $A_1$  і  $D$  відповідно [2].

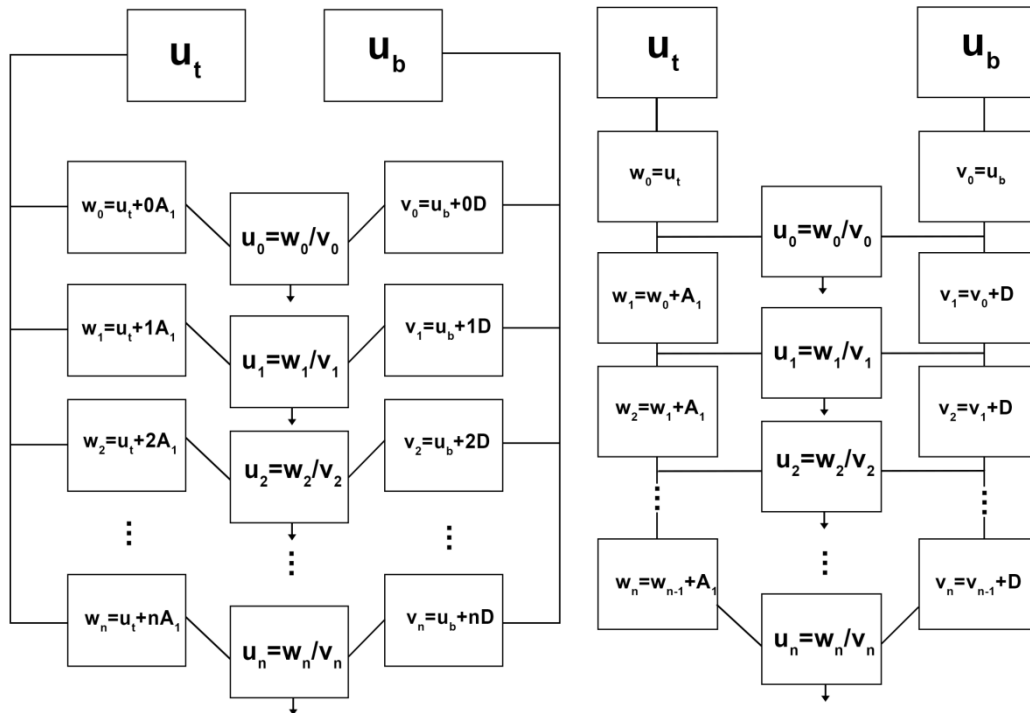


Рисунок 1 – Паралельний (зліва) та послідовний (зправа) підрахунок координат текстелів

Різниця в рівнях деталізації ділянки зображення на екрані та на текстурі призводить до появи аліазингу, тому при накладанні текстур виконують фільтрацію. Серед поширених методів фільтрації є білінійна та трілінійна.

При білінійній фільтрації значення кольору пікселя обраховується шляхом білінійної інтерполяції значень кольорів чотирьох текстелів. При трілінійній фільтрації виконують лінійну інтерполяцію результатів білінійної фільтрації на двох сусідніх МІР-рівнях. На жаль, ці методи не забезпечують високу реалістичність та

призводять до втрати деталей зображення, оскільки вони не враховують положення полігонів відносно спостерігача.

Найвищого рівня реалістичності дозволяє досягти анізотропна фільтрація. При анізотропній фільтрації проекція пікселя на поверхню текстури розглядається, як витягнутий еліпс, що дозволяє точніше визначати кольори пікселів (рис 2).

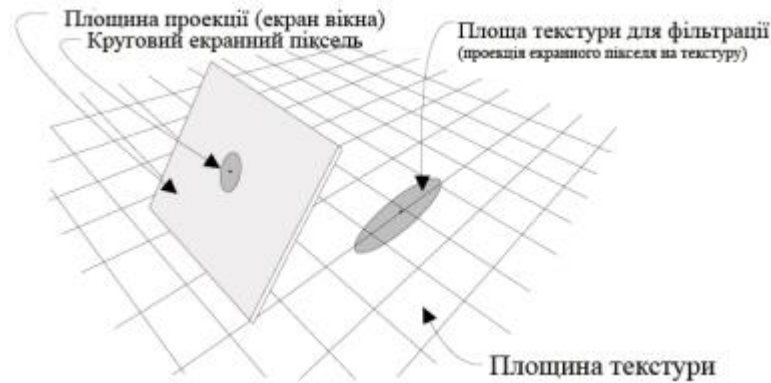


Рисунок 2 – Проекція пікселя на текстуру

Для визначення необхідних для фільтрації текселів складають рівняння еліпса за алгоритмом [3-4]:

1. Визначення векторів осей:

$$(U_x, V_x) = \left( \frac{du}{dx}, \frac{dv}{dx} \right),$$

$$(U_y, V_y) = \left( \frac{du}{dy}, \frac{dv}{dy} \right),$$

2. Визначення коефіцієнтів рівняння:

$$A = V_x^2 + V_y^2, \quad B = -2(U_x V_x + U_y V_y),$$

$$C = U_x^2 + U_y^2, \quad F = (U_x V_y + U_y V_x)^2,$$

3. Кінцеве рівняння:

$$AU^2 + BUV + CV^2 = F,$$

де  $U = u - u_0, V = v - v_0$ .

Похідні знаходять за формулами:

$$\frac{\partial u}{\partial x} = u(x+1, y) - u(x, y) = \frac{A(x+1) + By + C}{G(x+1) + Hy + I} - \frac{Ax + By + C}{Gx + Hy + I},$$

$$\frac{\partial v}{\partial x} = v(x+1, y) - v(x, y) = \frac{D(x+1) + Ey + F}{G(x+1) + Hy + I} - \frac{Dx + Ey + F}{Gx + Hy + I},$$

$$\frac{\partial u}{\partial y} = u(x, y+1) - u(x, y) = \frac{Ax + B(y+1) + C}{Gx + H(y+1) + I} - \frac{Ax + By + C}{Gx + Hy + I},$$

$$\frac{\partial v}{\partial y} = v(x, y+1) - v(x, y) = \frac{Dx + E(y+1) + F}{Gx + H(y+1) + I} - \frac{Dx + Ey + F}{Gx + Hy + I}.$$

Як видно з наведених формул обчислення часткових похідних для визначення коефіцієнтів рівняння еліпса є трудомісткою процедурою. На кожен піксель в рядку растеризації необхідно виконати 10 операцій множення, 10 операцій додавання, 4 операції ділення та 1 операцію віднімання. Тому запропоновано ітераційні формули для спрощення процедури визначення похідних [2]:

$$\frac{\partial u}{\partial x} = \frac{num_{i-1} + M}{(K + Gx)(L + Gx)}, \quad \frac{\partial v}{\partial x} = \frac{num_{i-1} + N}{(K + Gx)(L + Gx)},$$

$$\frac{\partial u}{\partial y} = \frac{num_{i-1} + O}{(K + Gx)(L + Gx)}, \quad \frac{\partial v}{\partial y} = \frac{num_{i-1} + P}{(K + Gx)(L + Gx)}.$$

де  $J = G + I$ ,  $K = I + Hy$ ,  $L = G + I + Hy = J + Hy$ , а параметри  $N, M, O, P$  визначаються за формулами:

$$M = AH - BG, \quad N = DH - EG, \quad O = BG - AH, \quad P = EG - DH.$$

Таким чином, кількість операцій додавання зменшено на 4, операцій множення зменшено на 8, а операцій віднімання зменшено на 1 [2].

Встановлено, що частина проміжних результатів обчислень при визначенні параметрів рівняння еліпса може бути закешована в оперативній пам'яті для повторного використання без перерахунку, що дозволяє підвищити продуктивність обчислень до 70% [3].

Для визначення кольору екранного пікселя використовують усереднення кольору всіх текселів, які попадають в еліпс. При усередненні кольорів текселів можуть використовуватись вагові коефіцієнти, що дозволяють врахувати фізичні особливості пікселів пристроїв виводу. В переважній більшості існуючих засобів комп'ютерної графіки використовують вагові коефіцієнти, які не залежать від розмірів та форми проекції пікселя на текстуру [5-8]. Тому запропоновано вагову функцію на основі Гаусівської моделі пікселя (рис 4), яка дає можливість точніше визначати кольори пікселів [5]:

$$W(u, v) = \left( 1 - \frac{-1}{2} \left[ \frac{[U \cdot \cos(\theta) + V \cdot \sin(\theta)]^2}{\sigma_1^2} + \frac{[V \cdot \cos(\theta) - U \cdot \sin(\theta)]^2}{\sigma_2^2} \right] \right)^8$$

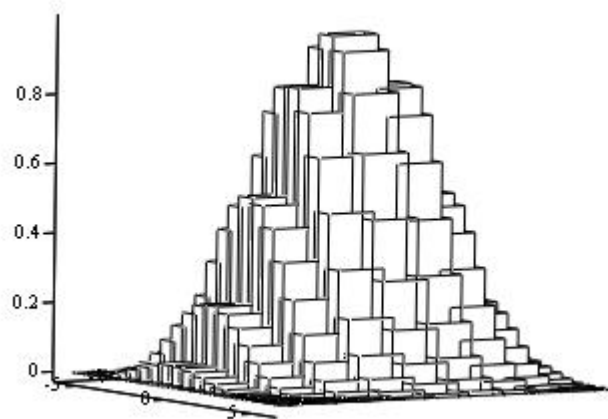


Рисунок 3 – Графік вагової функції

Висока обчислювальна складність ускладнює використання вагових функцій на основі гаусівської моделі пікселя у складних графічних сценах із високими вимогами до продуктивності обчислень. Тому запропоновано метод анізотропної фільтрації з використанням спеціальних текстурних карт вагових коефіцієнтів. Застосування карт вагових коефіцієнтів дає можливість істотно підвищити продуктивність анізотропної фільтрації із використанням складних вагових функцій за рахунок використання заздалегідь розрахованих вагових коефіцієнтів.

Для реалістичного відтворення нерівностей використовують різні методи рельєфного текстурування. Найбільш ефективним вважають *parallax mapping*. *Parallax mapping* використовується для процедурного створення тривимірного опису текстурованої поверхні з використанням карт висот замість генерації нової геометрії. *Parallax mapping* виконують шляхом зміщенням текстурних координат таким чином, щоб поверхня здавалася об'мною. Основний принцип методу полягає в тому, щоб визначати колір пікселя за текстурними координатами тієї точки, де видовий вектор перетинає поверхню. Зміщення текстурних координат визначають шляхом прорахунку трасування променів для карти висот.

Виконання анізотропної фільтрації при текстуруванні з використанням техніки *parallax mapping* може призводити до некоректного визначення кольорів пікселів, надмірного розмиття окремих ділянок зображення. Задача полягає в тому, що відомі методи анізотропної фільтрації не враховують нерівності поверхні, а тому всі текселі, що формують проекцію пікселя на текстуру в формі еліпса, вважаються видимими та використовуються для визначення кольору пікселя. Насправді ділянки з більшою висотою можуть перекривати ділянки з меншою висотою.

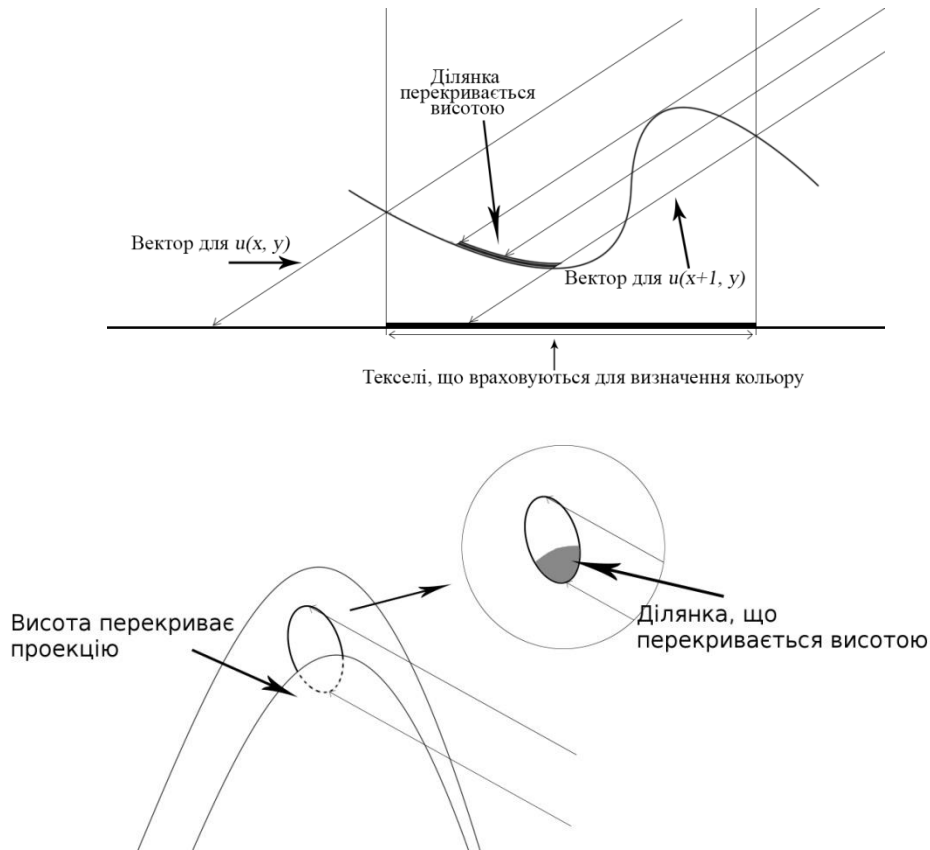


Рисунок 4 – Перекриття частини проекції пікселя більшою висотою

Для усунення артефактів запропоновано метод врахування рельєфних особливостей поверхні при анізотропній фільтрації текстур шляхом зворотного трасування променя для визначення видимості точки.

Трасування видового вектора – достатньо ресурсомістка процедура, а тому потребує використання спрощених методів обчислень для забезпечення високої продуктивності при відтворенні графічних зображень у реальному часі. Високу точність і достатню продуктивність забезпечує метод Доннеллі, що базується на використанні додаткової тривимірної текстури – карти відстаней до поверхні. Ця текстура будується на основі карти висот, де в якості третього виміру використовується висота. При цьому значення кожного текселя є найближча відстань до поверхні, заданої картою висот [6].

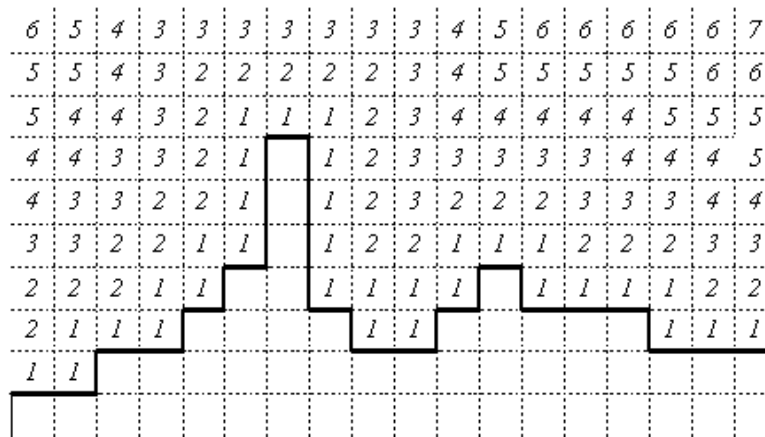


Рисунок 5 – Двовимірний фрагмент карти відстаней до поверхні

Було доведено, що метод Донеллі не є оптимальним для поверхонь із складним рельєфом. Тому запропоновано модифікацію метода Донеллі, яка полягає в уточненні карти відстаней до поверхні для виконання parallax mapping. Вектора, який проходить через деяку точку  $P$ , що лежить над поверхнею, і перетинає поверхню таким чином, що частина вектора на проміжку від точки  $P$  до поверхні рівна мінімальній відстані від цієї точки до поверхні, і, при цьому, не перетинає поверхню на проміжку від початку вектора до цієї точки  $P$  може не існувати. В такому випадку зберігати в карті відстаней мінімальну відстань від точки  $P$  до поверхні не доцільно. Оптимальніше зберігати найменшу з можливих відстаней від точки до поверхні за умови, що вектор, який проходить через точку  $P$ , не перетинається з поверхнею на проміжку від початку вектора до точки  $P$ . Уточнення дає можливість знизити обчислювальні затрати при текстуруванні поверхонь із складним рельєфом без втрати точності обчислень.

## ВИСНОВКИ

Запропоновано методи для підвищення продуктивності та реалістичності текстурування, зокрема, методи перспективно-коректного текстурування, фільтрації текстур, рельєфного текстурування.

## ЛІТЕРАТУРА

1. Романюк О. Н. Неортогональна растеризація при перспективнокоректному текстуруванні [Текст] / О. Н. Романюк, О. О. Дудник, О. В. Мельник // VI Международная конференция «Моделирование и компьютерная графика», м. Красноармійськ, 25-29 травня 2015 р. — 2015. — С. 174-178.
2. Романюк О. Н. Метод підвищення продуктивності перспективно-коректного текстурування [Текст] / О. Н. Романюк, О. О. Дудник // Наукові праці

ДонНТУ. Серія "Інформатика, кібернетика та обчислювальна техніка". - 2016. - № 1 (22). - С. 43-46.

3. Романюк О. Н. Підвищення продуктивності перспективно-коректного текстування з використанням анізотропної фільтрації [Текст] / О. Н. Романюк, О. О. Дудник // Вимірювальна та обчислювальна техніка в технологічних процесах. - 2016. - № 3. - С. 192-195.

4. Романюк О. Н. Анізотропна фільтрація текстур з використанням методів кешування [Текст] / О. Н. Романюк, О. О. Дудник // Вісник Вінницького політехнічного інституту. - 2016. - № 6. - С. 59-64.

5. Романюк О. Н. Анізотропна фільтрація з використанням вагової функції на основі Гаусівської моделі пікселя [Текст] / О. Н. Романюк, І. В. Абрамчук, О. О. Дудник // Вимірювальна та обчислювальна техніка в технологічних процесах. - 2016. - № 2. - С. 117-121.

6. Романюк С. О. Анізотропна фільтрація з використанням вагових функцій / С. О. Романюк, О. О. Дудник, Л. А. Савицька, О. В. Романюк // Вісник Херсонського національного технічного університету. – Херсон – 2015. - № 3. - С. 459-46

7. Романюк О. Н. Математичні моделі пікселя / О. Н. Романюк, О. О. Дудник // Електронні інформаційні ресурси: створення, використання доступ. Збірник міжнародної науково-практичної конференції. - Вінниця. - 2014. - С. 289-293.

8. Романюк О. Н. Модифікація гаусівської моделі пікселя для задач антиаліазингу [Текст] / О. Н. Романюк, І. В. Абрамчук, О. О. Дудник, О. В. Мельник // Наукові праці ДонНТУ. Серія "Інформатика, кібернетика та обчислювальна техніка". — 2015. - № 1(20). — С. 84-88.

9. Романюк О. Н. Особливості анізотропної фільтрації текстур при використанні технології parallax mapping [Текст] / О. Н. Романюк, О. О. Дудник // Вісник хмельницького національного університету серія: Технічні науки. - 2017. - № 1. - С. 236-241.

10. Романюк О. Н. Модифікований метод parallax mapping з використанням карти відстаней до поверхні [Текст] / О. Н. Романюк, О. О. Дудник, О. В. Романюк // Інформаційні технології та комп'ютерна інженерія. - 2017. - № 1. - С. 78-82.

Отримано 26.06.2017