

О. Н. Романюк¹
О. О. Дудник¹

Анізотропна фільтрація текстур з використанням методів кешування

¹Вінницький національний технічний університет

Для підвищення продуктивності текстурування запропоновано методи кешування проміжних результатів обчислень при визначенні коефіцієнтів рівняння еліпса для анізотропної фільтрації текстур.

Ключові слова: текстурування, фільтрація текстур, анізотропна фільтрація.

Вступ

Підвищення інформативності комп'ютерної графіки досягають за рахунок формування зображень, які точніше відтворюють конструктивні та візуальні особливості об'єкта. Під час формування таких зображень у графічних системах необхідно відображати сцени з великою деталізацією, тому на цьому етапі розвитку комп'ютерної графіки особливу увагу приділяють реалістичності. Для побудови високореалістичних зображень використовують текстури, які накладають на графічні об'єкти [1—5]. Процес текстурування вимагає виконання значної кількості складних обчислень в реальному часі, що істотно підвищує вимоги до апаратного забезпечення. Сучасні засоби обчислювальної техніки оперують достатньо великими обсягами пам'яті, тому існує можливість підвищення продуктивності складних обчислень за рахунок використання заздалегідь розрахованих значень.

Аналіз літератури та постановка задачі

Текстурування — це спосіб надання фактурних особливостей 3D деталі — полігону: кольору, блиску, матовості та інших фізичних властивостей (для імітації природного матеріалу, наприклад: паперу, дерева, каменю, металу тощо) [1—5].

Текстурування є важливою процедурою 3D-рендерингу, оскільки дозволяє відтворити також малі об'єкти поверхні, створення яких полігонами виявилось б надмірно ресурсомістким. Наприклад, складки на одязі, дрібні камені, предмети на поверхні стін і ґрунту та багато іншого [1, 3, 5]. Текстура використовується в якості шару для додання певного ефекту або зміни геометрії всьому зображенню чи його частини. Генерація текстури полягає в проектуванні зображення на тривимірну поверхню.

Використання текстур для імітації нерівностей на поверхні передбачає встановлення співвідношення між екранними координатами об'єкта та координатами текстури. Найпоширенішими є афінне [1, 3] та перспективно-коректне накладання текстур [1—3].

За афінного текстурування [1, 2, 5] текстурні координати лінійно інтерполюються вздовж рядка rasterизації, що призводить до появи артефактів та недостовірного відтворення перспективи об'єкта.

За перспективно-коректного текстурування для визначення координат текселя, що відповідає екранному пікселю, використовують нелінійні функції, розрахунок яких передбачає виконання трудомістких операцій. Перспективно-коректне текстурування у переважній більшості випадків реалізують за методом Хекберта [1—3, 5]

$$u = \frac{Ax + By + C}{Gx + Hy + I}; \quad v = \frac{Dx + Ey + F}{Gx + Hy + I}, \quad (1)$$

де u і v — текстурні координати; x і y — екранні координати об'єкта $A, B, C, D, E, F, G, H, I$ — коефіцієнти полігона, який текстурується.

Визначення кольору пікселя на екрані шляхом врахування кольору лише одного текселя, приводить до появи різноманітних помилок візуалізації — артефактів. Тому для визначення кольору пікселя використовують різні методи фільтрації текстур [1, 4, 5].

На цьому етапі для текстурування широко використовується анізотропна фільтрація, яка дозволяє точніше визначати кольори пікселів, які відповідають елементам текстури, розташованим не паралельно екрану. Це один із найкращих і поширених методів текстурування. На відміну від ізотропних видів фільтрації (білінійна [3, 5] та трилінійна [3, 5]) використовується проекція пікселя на текстурну поверхню [1, 4, 5].

За анізотропної фільтрації проекція пікселя на поверхню текстури розглядається як витягнутий еліпс (рис. 1) та передбачає усереднення кольору текселів усередині проекції, що дозволяє точніше визначати кольори пікселів [1, 4].

Для визначення необхідних для фільтрації текселів складають рівняння еліпса за формулами [4, 5]

$$\begin{aligned} (U_x, V_x) &= \left(\frac{du}{dx}, \frac{dv}{dx} \right); \\ (U_y, V_y) &= \left(\frac{du}{dy}, \frac{dv}{dy} \right); \\ A &= V_x^2 + V_y^2; \\ B &= -2(U_x V_x + U_y V_y); \\ C &= U_x^2 + U_y^2; \\ F &= (U_x V_y + U_y V_x)^2; \\ AU^2 + BUV + CV^2 &= F, \end{aligned} \quad (2)$$

де $U = u - u_0$, $V = v - v_0$.

Як видно з формул (1) і (2), обчислення похідних для визначення коефіцієнтів рівняння еліпса є трудомісткою процедурою. На кожен піксель у рядку растеризації необхідно виконати 10 операцій множення, 10 операцій додавання, 4 операції ділення та 1 операцію віднімання [5]. Для обчислення коефіцієнтів рівняння еліпса (A , B , C , F) необхідно виконати ще 9 операцій множення та 4 операції додавання. Тому існує потреба у розробці методів, які б мали меншу обчислювальну складність.

Метою роботи є підвищення продуктивності анізотропної фільтрації текстур, шляхом застосування методів кешування даних.

Результати дослідження

З формул (1) і (2) видно, що коефіцієнти рівняння еліпса залежать лише від параметрів проекції (коефіцієнти $A...I$ формули (1)) і координат пікселя на екрані. В переважній більшості графічних сцен для усіх полігонів використовуються однакові параметри проекції, тому значення коефіцієнтів $A...I$ формули (1) для усіх полігонів рівні. Виходячи з цього, можна зробити висновок, що для будь-якої пари x і y значення коефіцієнтів рівняння еліпса можуть бути обчислені до початку рендерингу сцени, незалежно від належності цих координат конкретному полігону. Діапазони можливих значень x і y залежать від розмірів сторін екрану, які також відомі до початку рендерингу графічної сцени. Тому існує можливість визначення та кешування в оперативній пам'яті коефіцієнтів рівняння еліпса для усіх пар x і y до початку рендерингу сцени з їх подальшим використанням у процесі растеризації полігонів.

Розрахуємо необхідний обсяг додаткової пам'яті для застосування такої техніки кешування. Кожен із коефіцієнтів рівняння еліпса (A , B , C , F) може бути заданий числом з рухомою комою. Числа з рухомою комою в обчислювальній техніці задають відповідно до стандарту IEEE 754 [6]. Розрядність числа з рухомою комою одинарної точності складає 4 байти, а для чисел з подвійною точністю — 8 байт. Тому для збереження в пам'яті чотирьох коефіцієнтів рівняння для одного

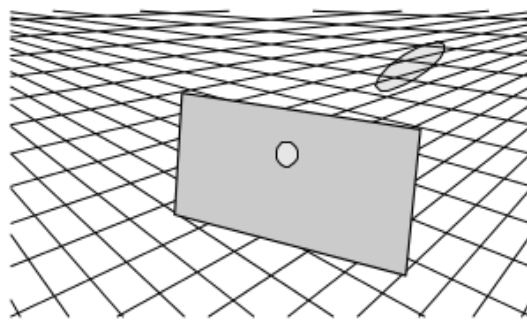


Рис. 1. Проекція пікселя на поверхню текстури

екранного пікселя необхідно 16 байт у разі використання чисел одинарної точності або 32 байти для чисел подвійної точності. Обчислити необхідний для фільтрації усієї сцени обсяг додаткової пам'яті для чисел одинарної точності можна за формулою

$$V = 16(w \cdot h),$$

а для чисел подвійної точності за формулою

$$V = 32(w \cdot h),$$

де V — необхідний обсяг додаткової пам'яті; w — ширина екрану, h — висота екрану.

Ефект від використання такого методу кешування для чисел з подвійною точністю на екранах з різною роздільною здатністю відображено у табл. 1. Моделювання проводилось на комп'ютері на базі процесора Intel(R) Core(TM) i5-4590 з тактовою частотою 3,30 ГГц, оперативна пам'ять 8ГБ DDR3 з частотою 1600 МГц.

Таблиця 1

Показники ефективності кешування параметрів рівняння еліпса

Роздільна здатність	Час обчислення без кешування (с)	Час обчислення з кешуванням (с)	Об'єм додаткової пам'яті (Кб)	Приріст продуктивності (%)
VGA 640×480	0,0073	0,0018	9600	75,6
XGA 1024×768	0,0191	0,0035	24576	81,73
HD 720p 1280×720	0,0225	0,0056	28800	75,0
Full HD 1080p 1920×1080	0,0496	0,0114	64800	77,11

Як видно з табл. 1, кешування параметрів рівняння еліпса дозволяє істотно зменшити час, необхідний на їх визначення (понад 75 %). При цьому обсяг необхідної для зберігання кешу даних зростає пропорційно площі екрану.

Отже, використання кешування параметрів рівняння еліпса дає можливість істотно підвищити продуктивність усього процесу анізотропної фільтрації. Проте, для окремих задач, що вимагають економії оперативної пам'яті, такий підхід може бути неефективним. Тому існує потреба у використанні складніших методів кешування, що вимагають меншого обсягу пам'яті.

Відповідно до методу прискореного обчислення частинних похідних для визначення коефіцієнтів рівняння еліпса за анізотропної фільтрації, частинні похідні можуть бути визначені за формулами [5]

$$\frac{\partial u}{\partial x} = \frac{AI - CG + AHy - BGy}{(I + Gx + Hy)(G + I + Gx + Hy)}; \quad (3)$$

$$\frac{\partial v}{\partial x} = \frac{DI - FG - DHy + EHy}{(I + Gx + Hy)(G + I + Gx + Hy)}; \quad (4)$$

$$\frac{\partial u}{\partial y} = \frac{BI - CH - AHx + BGx}{(I + Gx + Hy)(G + I + Gx + Hy)}; \quad (5)$$

$$\frac{\partial v}{\partial y} = \frac{EI - FH - DHx + EGx}{(I + Gx + Hy)(G + I + Gx + Hy)}. \quad (6)$$

Чисельники формул можуть бути обчислені ітеративно.

З формул (3)—(6) видно, що чисельники формул (3) і (4) не залежать від x координати пікселя, а чисельники формул (5) і (6) не залежать від y координати пікселя. Таким чином, за умови, що параметри проекції однакові для всіх полігонів сцени, значення чисельників формули (3) і (4) рівні при опрацюванні будь-якого конкретного рядка екрану, а значення чисельників формул (5) і (6) рівні для будь-якого стовпця екрану, незалежно від конкретного полігону (рис. 2).

Таким чином, значення чисельників для формул (3)—(6) можуть бути розраховані до початку растеризації сцени. При цьому достатньо розрахувати їх значення лише в одній точці, а для інших отримати ітеративно [5].

Формули (3)—(6), для обчислення похідних в реальному часі матимуть вигляд

$$\frac{\partial u}{\partial x} = \frac{num_1}{(I + Gx + Hy)(G + I + Gx + Hy)};$$

$$\frac{\partial v}{\partial x} = \frac{num_2}{(I + Gx + Hy)(G + I + Gx + Hy)};$$

$$\frac{\partial u}{\partial y} = \frac{num_3}{(I + Gx + Hy)(G + I + Gx + Hy)};$$

$$\frac{\partial v}{\partial y} = \frac{num_4}{(I + Gx + Hy)(G + I + Gx + Hy)};$$

Значення чисельника для формул (5), (6) вздовж рядка рівні для обох полігонів

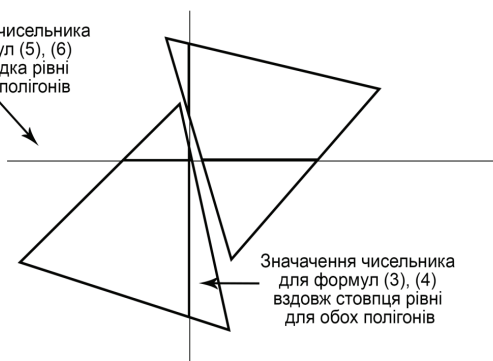


Рис. 2. Рівні значення чисельників для різних полігонів

де $num_1 \dots num_4$ — значення чисельників, що закешовані.

Оскільки закешоване значення чисельника для формул (3) і (4) спільне для кожного пікселя в екранному рядку, а закешоване значення чисельника для формул (5) і (6) спільне для кожного пікселя в екранному стовпці розрахувати необхідний обсяг додаткової пам'яті для чисел одинарної точності можна за формулою

$$V = 8w + 8h = 8(w + h),$$

а для чисел подвійної точності за формулою

$$V = 16w + 16h = 16(w + h),$$

де V — необхідний обсяг додаткової пам'яті; w — ширина екрану; h — висота екрану.

Таким чином обсяг необхідної пам'яті зростає пропорційно периметру екрану, а не його площі, що істотно знижує вимоги до обсягу додаткової пам'яті. Проте, частина обчислень буде проводитись у реальному часі, що знижує продуктивність порівняно з попереднім методом.

Ефект від використання такого методу кешування для чисел з подвійною точністю на екранах із різною роздільною здатністю відображено у табл. 2.

Таблиця 2

Показники ефективності кешування чисельників формул (3)—(6)

Роздільна здатність	Час обчислення без кешування класичним методом (с)	Час обчислення без кешування за методом прискореного обчислення похідних (с)	Час обчислення з кешуванням (с)	Обсяг додаткової пам'яті (Кб)	Приріст продуктивності відносно класичного методу (%)	Приріст продуктивності відносно класичного методу з прискореним обчисленням похідних (%)
VGA 640×480	0,0073	0,0059	0,0051	17,5	30,7	13,97
XGA 1024×768	0,0191	0,0151	0,0130	28	31,74	13,49
HD 720p 1280×720	0,0225	0,0175	0,0152	31,25	32,5	13,5
Full HD 1080p 1920×1080	0,0496	0,0396	0,0337	46,88	31,96	14,87

Як видно з табл. 2, запропонований метод кешування забезпечує достатньо високий приріст продуктивності порівняно з методами без кешування. При цьому затрати оперативної пам'яті значно менші порівняно з попереднім методом.

Значення знаменника для формул (3)—(6) також може бути розраховано до початку растеризації. У такому випадку обсяг необхідної пам'яті зростає пропорційно площі екрану, оскільки знаменник залежить як від x координати, так і від y координати. Так як в формулах (3)—(6), чисельники рівні, необхідно збільшити розмір кешу лише на 4 байти, для чисел одинарної точності, або на 8 байт, для чисел подвійної точності, на кожен піксель екрану. Необхідний обсяг додаткової пам'яті для чисел одинарної точності можна обчислити за формулою:

$$V = 8(w + h) + 4(w \cdot h),$$

а для чисел подвійної точності — за формулою:

$$V = 16(w + h) + 8(w \cdot h),$$

де V — необхідний обсяг додаткової пам'яті; w — ширина екрану; h — висота екрану.

Ефект від використання кешування чисельників і знаменників формул (3)—(6) для чисел з подвійною точністю на екранах з різною роздільною здатністю відображено у табл. 3.

Таблиця 3

Показники ефективності кешування чисельників та знаменників формул (3)—(6)

Роздільна здатність	Час обчислення без кешування класичним методом (с)	Час обчислення без кешування за методом прискореного обчислення похідних (с)	Час обчислення з кешуванням (с)	Обсяг до додаткової пам'яті (Кб)	Приріст продуктивності відносно класичного методу (%)	Приріст продуктивності відносно класичного методу з прискореним обчисленням похідних (%)
VGA 640×480	0,0073	0,0059	0,0044	2417,5	40,43	26,05
XGA 1024×768	0,0191	0,0151	0,0110	6172	42,2	26,75
HD 720p 1280×720	0,0225	0,0175	0,0128	7231,25	43,13	27,12
Full HD 1080p 1920×1080	0,0496	0,0396	0,0284	16246,88	42,7	28,31

Як видно з табл. 3, запропонований метод кешування забезпечує достатньо високий приріст продуктивності порівняно з методами без кешування та методом з кешуванням лише чисельників формул (3)—(6), однак вимагає вищих обсягів оперативної пам'яті порівняно з попереднім методом, але істотно нижчих порівняно з методом кешування параметрів рівняння еліпса.

Отже, запропоновано три методи кешування проміжних результатів обчислень під час визначення коефіцієнтів рівняння еліпса для анізотропної фільтрації текстур. Метод кешування параметрів рівняння еліпса забезпечує найвищий приріст продуктивності, проте може вимагати значних затрат оперативної пам'яті для рендерингу зображень з високою роздільною здатністю. Методи, що базуються на кешуванні проміжних обчислень у процесі визначення похідних, забезпечують менший приріст продуктивності. Проте вимагають значно менших затрат оперативної пам'яті і можуть бути використані в системах рендерингу реального часу з жорсткими обмеженнями оперативної пам'яті або шини даних. Графік залежності необхідного обсягу додаткової пам'яті від роздільної здатності екрану для трьох методів зображено на рис. 3.

Методи, що базуються на кешуванні проміжних обчислень у процесі визначення часткових похідних, доцільно використовувати для рендерингу в реальному часі анімованих графічних сцен, у яких можуть змінюватись параметри проекції або роздільна здатність на різних кадрах. У такому випадку кешування доцільно здійснювати в ході рендерингу сцени з використанням методів динамічного програмування [7], здійснюючи очищення кешу у разі зміни параметрів.

Висновки

Запропоновано методи кешування проміжних результатів обчислень для визначення коефіцієнтів рівняння еліпса для анізотропної фільтрації текстур. Використання кешування під час обчислення коефіцієнтів рівняння еліпса дало можливість істотно підвищити продуктивність анізотропного текстурування.

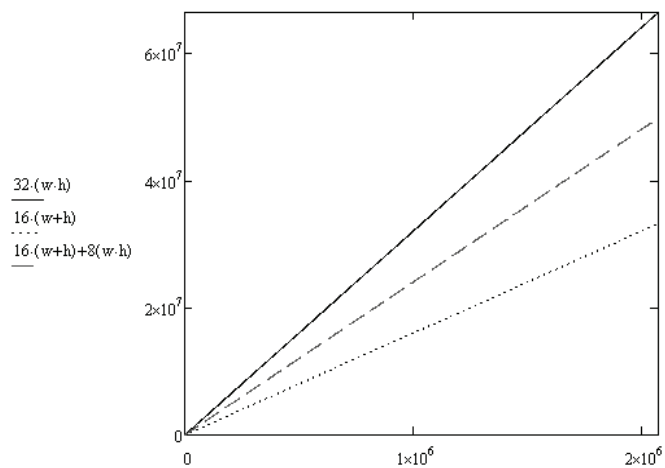


Рис. 3. Графік залежності необхідного обсягу пам'яті від роздільної здатності екрану

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Paul S. Heckbert, Survey of texture mapping [Electronic resource] / Paul S. Heckbert // Graphics Interface '1986. — Access mode: <https://www.cs.cmu.edu/~ph/textsurv.pdf>.
2. Романюк О. Н. Неортогональна растеризація при перспективно-коректному текстурованні / О. Н. Романюк, О. О. Дудник, О. В. Мельник, // Моделювання та комп'ютерна графіка : VI міжнародна конференція, ДВНЗ «ДонНТУ», м. Красноармійськ, 25—29 травня 2015 р.
3. Романюк О. Н. Метод підвищення продуктивності перспективно-коректного текстуровання / О. Н. Романюк, О. О. Дудник // Наукові праці ДонНТУ. — 2016. — № 1 (22).
4. Szeliski R. Computer Vision: Algorithms and Applications [Electronic resource] / Richard Szeliski // September 3, 2010 draft © 2010 Springer. — Access mode: <http://szeliski.org/Book/>.
5. Романюк О. Н. Підвищення продуктивності перспективно-коректного текстуровання з використанням анізотропної фільтрації / О. Н. Романюк, О. О. Дудник // Вимірювальна та обчислювальна техніка в технологічних процесах. — 2016. — № 3 (56).
6. 754-2008 — IEEE Standard for Floating-Point Arithmetic. Revision of ANSI/IEEE Std 754-1985 [Electronic resource] / IEEE Explore® Digital Library. — Access mode: <http://ieeexplore.ieee.org/document/4610935/>.
7. Дасгупта С. Алгоритмы / С. Дасгупта, Х. Пападимитриу, У. Вазирані. — М. : МЦНМО, 2014. — 320 с.

Рекомендована кафедрою програмного забезпечення

Стаття надійшла до редакції 2.11. 2016

Романюк Олександр Никифорович — д-р техн. наук, професор, перший проректор, e-mail: ran12345@mail.ru ;

Дудник Олександр Олександрович — аспірант кафедри програмного забезпечення. e-mail: dudnyk1@inmad.vntu.edu.ua .

Вінницький національний технічний університет, Вінниця

O. N. Romaniuk¹
O. O. Dudnyk¹

Anisotropic Texture Filtering Using Caching Methods

¹Vinnitsia National Technical University

The article suggests methods for caching intermediate results of calculations in the determination of the coefficients of the equation of an ellipse to anisotropic filtering textures to improve the performance of texture mapping.

Keywords: texture mapping, texture filtering, anisotropic filtering.

Romaniuk Oleksandr N. — Dr. Sc. (Eng.), Professor, First Vice-Rector, e-mail: ran12345@mail.ru ;

Dudnyk Oleksandr O. — Post-Graduate Student of the Chair of Software, e-mail: dudnyk1@inmad.vntu.edu.ua

A. N. Романюк¹
A. A. Дудник¹

Анизотропная фильтрация текстур с использованием методов кэширования

¹Винницкий национальный технический университет

Предложены методы кэширования промежуточных результатов вычислений при определении коэффициентов уравнения эллипса для анизотропной фильтрации текстур.

Ключевые слова: текстурирование, фильтрация текстур, анизотропная фильтрация.

Романюк Александр Никифорович — д-р техн. наук, профессор, первый проректор, e-mail: ran12345@mail.ru ;

Дудник Александр Александрович — аспирант кафедры программного обеспечения, e-mail: dudnyk1@inmad.vntu.edu.ua