

**ПІДВИЩЕННЯ РЕАЛІСТИЧНОСТІ ФОРМУВАННЯ ГРАФІЧНИХ
ЗОБРАЖЕНЬ ЗА РАХУНОК ПЕРСПЕКТИВНО КОРЕКТНОГО ПРОЕКТУВАННЯ**

У даній статті розглядаються шляхи підвищення реалістичності формування графічних сцен за рахунок перспективно коректного проектування.

Ключові слова: інтерполяція, метод Гуро, афінне текстурування, білінійне текстурування, проекційне текстурування.

Проекціювання 3D-моделей на площину – невід’ємна частина комп’ютерного моделювання. Проекціювання задає спосіб відображення об’єктів на екрані графічного пристрою. Для формування реалістичних 3D-сцен надзвичайно важливо проводити перспективно-коректне проектування. Перспектива надзвичайно важливий аспект моделювання, наприклад, об’єкти, які знаходяться далі, здаються меншими за розміром, ніж ті, що знаходяться ближче до точки спостереження. Без коректного прорахунку сцена не буде мати реалістичності.

Пряма діагональна лінія на рис.1, проходить від лівого нижнього кута до правого верхнього. Можна помітити, що пара пікселів в лівому нижньому куті відображають набагато меншу довжину в площині текстури ніж пікселі в кінці прямої. Це означає, що проводити просту лінійну інтерполяцію в цій ділянці не можна, бо це призведе до перспективно некоректної побудови сцени. Коректна побудова перспективи реалізується шляхом інтерполяції $u(z)$ і v/z замість u і v [1]. $U(z)$ і v/z можуть бути обчислені лінійно, але потрібно отримати значення u і v для кожного пікселя. Це можна зробити шляхом ділення цих значень на координату z . Тому для інтерполяції v/z , u/z , z необхідно провести дві операції ділення для кожного пікселя. В той час як вздовж діагоналі лінійна інтерполяція не проводиться, в просторі текстури будь-яка горизонтальний рядок може бути прорахований лінійно, адже вздовж горизонтального рядка кожен квадрат має однаковий розмір, хоча цей розмір і різниться для кожного рядка. Даний ефект називається постійним z -нахилом. Для кожного пікселя на горизонтальному рядку глибина сцени однакова. Отже, в будь-якому випадку необхідно інтерполювати v/z і u/z на межах полігонів, але коли необхідно текстурувати однакові елементи в горизонтальному напрямку, наприклад, сходи, можна дублювати інтерпольовані значення u і v [2].

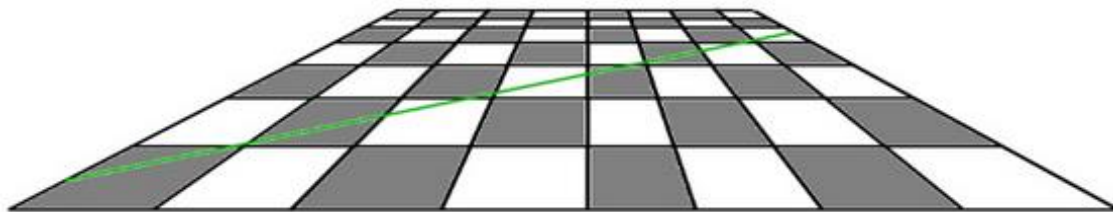


Рисунок 1 – Демонстрація перспективного текстурування

У статті буде розглянуто методики та технології для прорахунку перспективно-коректного текстурування.

При класичній реалізації текстурування та зафарбування пікселів за Гуро z -координата враховується тільки у вершинах трикутника при визначенні інтенсивностей їх кольору. У подальшому z -координата використовується тільки для видалення невидимих поверхонь. Таким чином, при зафарбовуванні Гуро перспектива об’єкта не враховується [3].

Розглянемо пряму з кінцевими точками $(X_0; Z_0)$ і $(X_1; Z_1)$. Вони проектується на

площину екрану в точках S_0, S_1 .

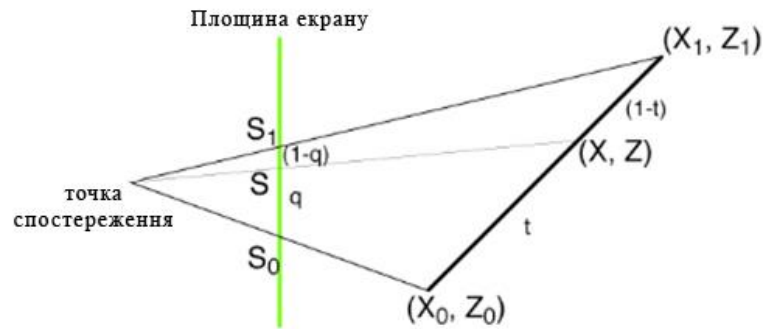


Рисунок 2 – Проекція прямої на екранну площину з урахуванням перспективи

При виконанні лінійного інтерполювання для розрахунку інтенсивностей кольору вздовж ребра $(X_0, Z_0; X_1, Z_1)$ за методом Гуро використовується формула [3]:

$$w = \frac{u \cdot z_1}{z_2 - u \cdot (z_2 - z_1)} \quad (1)$$

Згідно з методом Гуро інтенсивність кольору вздовж ребра визначається за формулою: $I_w = I_A + w \cdot (I_B - I_A)$. Підставивши у наведений вираз значення w , отримуємо:

$$I_w = I_A + \frac{u \cdot z_1}{z_2 - u \cdot (z_2 - z_1)} (I_B - I_A). \quad (2)$$

Афінне текстурування включає в себе обертання, розтягування та деформацію текстури. Текстурування $T(x)$ називають лінійним, якщо $T(x+y)=T(x)+T(y)$ і $T(ax)=aT(x)$ для будь-якого скалярного a ; текстурування $T(x)$ називається афінними, якщо існує константа c і лінійне текстурування $L(x)$, таке що $T(x)=L(x)+c$ для будь-якого x . Можна сказати, що лінійне текстурування – це підмножина афінних перетворень [4]. Приклад афінного текстурування можна побачити на рис. 3.

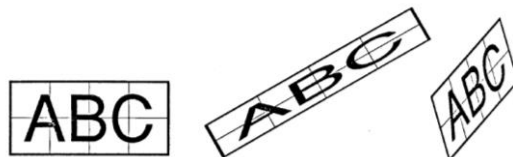


Рисунок 3 – Афінне текстурування

Афінне текстурування зберігає паралельні лінії і рівномірне розподілення точок вздовж ліній (це означає, що рівномірно розподілені точки вздовж рядка в початковому просторі перетворюються на рівновіддалені точки вздовж лінії в просторі текстури, хоча інтервал між цими точками у цих площинах може бути різним).

Білінійне текстурування спрощено визначається як лінійне перетворення квадрата на площині екрану у чотирикутник на площині текстури. Як показано на рис. 4, даний вид текстурування може бути обчислений шляхом лінійної інтерполяції параметра u вздовж верхнього та нижнього країв чотирикутника, а потім лінійної інтерполяції параметра v між двома інтерпольованими точками, в результаті чого й отримують кінцеву точку (x, y) [5].

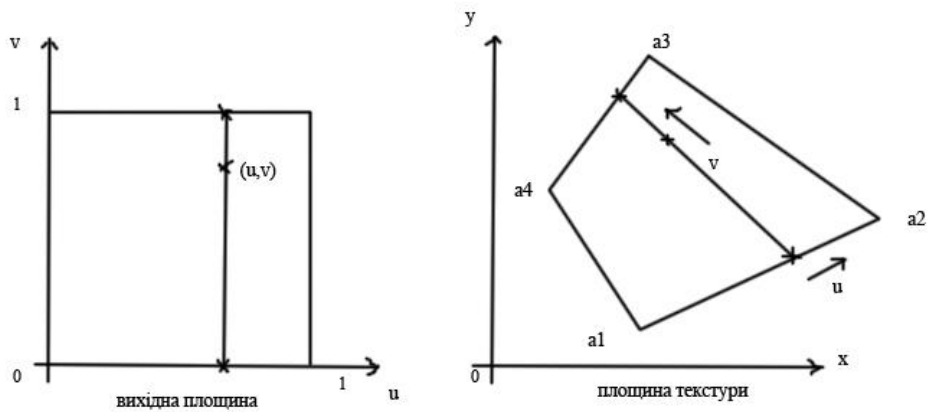


Рисунок 4 – Білінійне текстуровання

Білінійне текстуровання має ряд переваг: через його лінійні властивості, трансформація з вихідної площини до кінцевої зберігає вертикальні та горизонтальні лінії без змін, а також зберігає чотирикутне розташування точок вздовж цих ліній, але не зберігає діагональні лінії, що й показано на рис. 4.

Проекційне текстуровання, також відоме як перспективна або однорідна трансформація – це проекція однієї площини через певну точку на іншу площину. Однорідні трансформації широко використовуються для 3D-афінних перетворень і для перспективних проекцій [6].

Загальна форма проекційного текстуровання подібна до раціонального лінійного текстуровання [6] :

$$x = \frac{au + bv + c}{gu + hv + i}, y = \frac{du + eu + f}{gu + hv + i}$$

(4)

На відміну від білінійного текстуровання, що зберігає рівновіддалені точки вздовж певних ліній, проекційне текстуровання в загальному не зберігає рівновіддаленість точок на прямій (рис. 5). Замість цього зберігається величина, яка називається «подвійне розміщення» точок. Як і афінне текстуровання, проекційне зберігає лінії у всіх напрямках, в тому числі й діагоналі. Проекційне текстуровання афінне, коли: $g=h=0$. Насправді, проекційне текстуровання найкраще зберігає перспективно коректну орієнтацію ліній.

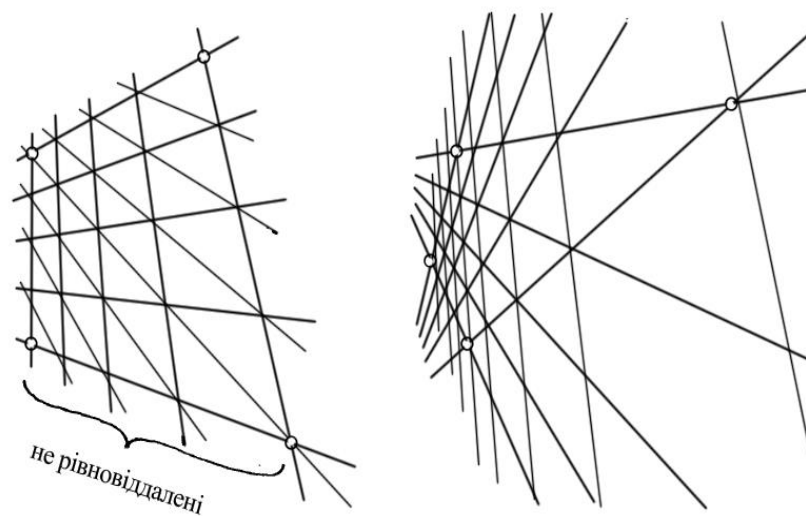


Рисунок 5 – Трансформація точок в результаті проекційного текстуровання

У таблиці 1 наведено порівняльну характеристику афінного, білінійного та проекційного текстуровання.

Таблиця 1 – Різні види перспективного проектування

Властивість	Афінне	Білінійне	Проекційне
Зберігає паралельні лінії	Так	ні	ні
Зберігає лінії	Так	ні	так
Зберігає точки, розташовані в межах чотирикутника	Так	ні	ні
Трансформує розташовані в межах чотирикутника точки у	Паралелограм	чотирикутник	чотирикутник
Однозначно зворотне	Так	ні	так

Отже, при текстуруванні необхідно враховувати перспективу для реалістичності формування 3D-сцен. Для того, щоб текстуровані об'єкти виглядали коректно, необхідно проводити перспективно коректне проектування. Воно гарантує, що текстура накладеться правильно на різні частини об'єкта. Коректне відтворення кольорів має місце за умови, що складові інтенсивностей кольору відповідних точок поверхні в світовій (об'єктній) та екранній системах координат співпадають. Перспективна корекція – ресурсомістка процедура (одна операція ділення на кожен піксель), що вимагає від 3D-прискорювачів реалізації цієї процедури апаратно [7].

ВИКОРИСТАНА ЛІТЕРАТУРА:

1. Вельтмандер П.В. Основные алгоритмы компьютерной графики / П. В. Вельтмандер. Новосибирский Государственный Технический Университет. – 1998.
2. Everitt Cass. Projective Texture Mapping. / Cass Everitt. In Computer Graphics (SIGGRAPH '95 Proceedings). – 1995.
3. Bala Kavita. Perspective correct texturing and quaternion interpolation./ Kavita Bala. 4629/5620 Fall. – 2012.
4. Heckbert Paul. Fundamentals of Texture Mapping and Image warping (Masters Thesis). / Paul Heckbert. Report No. UCB/CSD 89/516, Computer Science Division, University of California, Berkley. – 1989.
5. Faux I. D., Pratt M. J. Computational Geometry for Design and Manufacture. / I. D. Faux, M. J. Pratt. Ellis Horwood Ltd., Chichester, England. – 1979.
6. Maxwell E. A. The Methods of Plane Projective Geometry, Based on the Use of General Homogeneous Coordinates. / E. A. Maxwell. Cambridge U. Press, London. – 1946.
7. Blinn James F. Hyperbolic interpolation. / James F. Blinn. IEEE Computer Graphics (SIGGRAPH) and Applications, 12(4). – 1992.