

УДК 004.93

О. Н. Романюк, О. О. Дудник, О. В. Романюк, Д. А. Озерчук

## МЕТОД ПЕРСПЕКТИВНО-КОРЕКТНОГО ТЕКСТУРУВАННЯ

Вінницький національний технічний університет, Вінниця

**Анотація.** Текстурювання є одним із найскладніших етапів процесу кінцевої візуалізації, оскільки для кожної точки поверхні визначаються координати в дискретному просторі та значення інтенсивності кольору з урахуванням освітлення. Існуючі методи високоточного накладання текстур характеризуються суттєвою обчислювальною складністю, що в значній мірі впливає на час формування графічних зображень. Тому актуальними є питання підвищення продуктивності методів і засобів текстурювання.

Зменшити кількість операцій ділення при текстурюванні можливо за рахунок растеризації об'єкту в світовій системі координат за умови, що рядки растеризації на кожному етапі розміщено на фіксованій відстані від спостерігача.

Неортогональний по відношенню до координатних осей напрямок растеризації призведе до артефактів – наявності «просікань» і дублювання точок із-за зміщення по абсцисі початку рядків растеризації. Цього можна уникнути за умови, що провідне ребро трикутника буде паралельним осі ординат, однак це передбачає виконання особливої триангуляції поверхні та не відповідає вимогам графічних стандартів. Запропоновано растеризувати не трикутник, а прямокутник, у який його віртуально вписано. Запропоновано структурну схему блоку растеризації.

Аналіз показав, що для трикутника, який включає  $T$  внутрішніх точок, вилучається  $(T-q)$  операцій ділення, де  $q$  – кількість горизонтальних рядків растеризації трикутника.

**Ключові слова:** текстурювання, метод Хесберга, реалістичність, растеризація, перспективно-коректне накладання текстур.

**Аннотация.** Текстурирование является одним из самых сложных этапов процесса конечной визуализации, поскольку для каждой точки поверхности определяются координаты в дискретном пространстве и значения интенсивности цвета с учетом освещения. Существующие методы высокоточного наложения текстур характеризуются существенной вычислительной сложностью, что в значительной степени влияет на время формирования графических изображений. Поэтому актуальными являются вопросы повышения производительности методов и средств текстурирования.

Уменьшить количество операций деления при текстурировании возможно за счет растеризации объекта в мировой системе координат при условии, что строки растеризации на каждом этапе размещены на фиксированном расстоянии от наблюдателя.

Неортогональное по отношению к координатным осям направление растеризации приведет к артефактам - наличию «просечки» и дублирования точек из-за смещения по абсциссе начале строк растеризации. Этого можно избежать при условии, если ведущее ребро треугольника будет параллельным оси ординат, однако это предполагает выполнение особой триангуляции поверхности и не соответствует требованиям графических стандартов. Предложено растеризовать не треугольник, а прямоугольник, в который его виртуально размещено. Предложено структурную схему блока растеризации.

Анализ показал, что для треугольника, который включает  $T$  внутренних точек, исключается  $(T-q)$  операций деления, где  $q$  – количество горизонтальных строк растеризации треугольника.

**Ключевые слова:** текстурирование, метод Хесберга, реалистичность, растеризация, перспективно-корректное наложение текстур.

**Abstract.** Texturing is one of the most difficult stages of the final visualization process, as for each point of the surface the coordinates in the discrete space and the values of color intensity taking into account the lighting are determined. Existing methods of high-precision texture overlay are characterized by significant computational complexity, which significantly affects the time of formation of graphic images. Therefore, the issues of increasing the productivity of texturing methods and tools are relevant.

It is possible to reduce the number of division operations during texturing by rasterizing the object in the world coordinate system, provided that the rasterization lines at each stage are placed at a fixed distance from the observer.

Non-orthogonal in relation to the coordinate axes, the direction of rasterization will lead to artefacts - the presence of "perforations" and duplication of points due to the offset on the abscissa of the beginning of the rasterization lines. This can be avoided provided that the leading edge of the triangle is parallel to the y-axis, but this involves a special triangulation of the surface and does not meet the requirements of graphic standards. It is proposed to rasterize not a triangle, but a rectangle in which it is virtually inscribed. The structural scheme of the rasterization block is offered.

The analysis showed that for a triangle that includes  $T$  inner points, the division operations  $(T-q)$  are removed, where  $q$  is the number of horizontal rasterization rows of the triangle.

**Keywords:** testing, Hесberg method, realism, rasterization, perspective-correct texture overlay.

**DOI:** <https://doi.org/10.31649/1999-9941-2021-50-1-55-63>.

### Вступ

Найвищу реалістичність візуального відтворення об'єктів і процесів реального світу забезпечує тривимірний комп'ютерний графік [1-3].

У комп'ютерних системах [4] високу ефективність візуалізації забезпечують методи та засоби, що дають змогу оптимально використовувати обчислювальні ресурси графічної системи для забезпечення високого рівня реалістичності відтворення графічних зображень. Одним із таких засобів є текстурювання, яке дозволяє врахувати локальні особливості поверхні без надмірного ускладнення геометрії моделі. Процедури накладання текстур успішно застосовують для відтворення на поверхні об'єктів кольорів, фактури, нерівностей, імітації відбиття світла та інших візуальних властивостей.

Текстурювання є одним із найскладніших етапів процесу кінцевої візуалізації, оскільки для кожної точки поверхні визначаються координати в дискретному просторі та значення інтенсивності кольору з урахуванням освітлення. Існуючі методи високоточного накладання текстур характеризуються суттєвою обчислювальною складністю, що в значній мірі впливає на час формування графічних зображень. Тому актуальними є питання підвищення продуктивності методів і засобів текстурювання.

### Мета

Мета статті – розробка методу підвищення продуктивності перспективно-коректного нанесення текстури за рахунок використання неортогонального напрямку растеризації ділянки, обмеженої полігоном.

### Аналіз методів текстуровання

Процес накладання текстури на полігон виконується в три етапи: встановлення відповідності між екранними координатами точок, що належать полігону, та координатами у текстурній площині; вибірка текселів за цими координатами; обчислення кольорів пікселів на основі кольорів відібраних текселів.

Для встановлення відповідності між екранними та текстурними координатами існують методи, що враховують перспективу та методи, що її не враховують. Перші називають перспективно-коректними, а другі – афінними.

Афінні методи [5] базуються на лінійній інтерполяції [1, 6] текстурних координат вздовж рядка растеризації. Це забезпечує високий рівень продуктивності, оскільки для визначення координат текселя достатньо дві операції додавання на кожен піксел. Афінне текстуровання використовують при растеризації сцен із ортогональними проекціями. Використання афінного текстуровання при перспективній проекції можливе лише за умови, що полігон розташовано паралельно екранній площині. При растеризації сцен із перспективною проекцією, що містять довільно орієнтовані полігони, афінне текстуровання призводить до появи істотних артефактів (рис. 1). Тому використання афінних методів недоцільне при візуалізації графічних сцен, що вимагають високого ступеня реалістичності вихідного зображення.

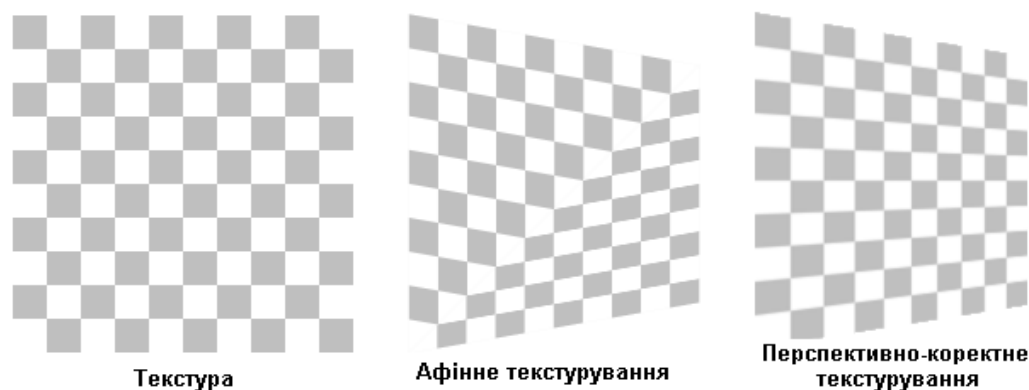


Рисунок 1 – Артефакти при афінному текстурованні

Перспективно-коректне текстуровання, як правило, виконують за методом Хекберта [5, 8], в основі якого лежить виконання зворотного перспективного перетворення  $U$  матричному вигляді зворотна проекція обчислюється за формулою [5, 9]:

$$[uq \quad vq \quad q] = [x \quad y \quad z] \begin{bmatrix} A & D & G \\ B & E & H \\ C & F & I \end{bmatrix} = [x \quad y \quad z] \begin{bmatrix} EI - FH & FG - DI & DH - EG \\ CH - BI & AI - CG & BG - AH \\ BF - CE & CD - AF & AE - BD \end{bmatrix}, \quad (1)$$

де  $u$ ,  $v$  – текстурні координати;  $x$ ,  $y$  – екранні координати;  $A - I$  – коефіцієнти перспективної проекції. Формулу (1) можна записати у вигляді многочленів:

$$u = \frac{Ax + By + C}{Gx + Hy + I}, \quad v = \frac{Dx + Ey + F}{Gx + Hy + I}, \quad (2)$$

З формули (2) видно що зворотне перспективне перетворення вимагає значних обчислювальних витрат, оскільки передбачає виконання «довгих» операцій. Така обчислювальна складність, часто, неприйнятна для використання у високопродуктивних графічних системах [4]. Тому на практиці використовують різні методи апроксимації для вилучення із обчислювального процесу операцій множення та ділення.

Широко поширеною є квадратична апроксимація [6-8], яка для розрахунку текстурних координат використовує рівняння

$$\begin{aligned}u(x) &= A_1 \cdot x^2 + A_2 \cdot x + A_3, \\v(x) &= B_1 \cdot x^2 + B_2 \cdot x + B_3,\end{aligned}$$

де  $A_1-A_3$ ,  $B_1-B_3$  – коефіцієнти апроксимації, які є сталими для кожного рядка растеризації (PP);  
 $x = \frac{x_i - x_1}{x_m - x_1}$ , при  $x_1$ ,  $x_m$  – значення  $x$  – координати відповідно у початковій та кінцевій точках PP.

Формули для розрахунку коефіцієнтів  $A_1 - A_3$  мають вигляд [5]:

$$A_1 = 2u_0 - 4u_1 + 2u_2, \quad A_2 = -3u_0 + 4u_1 - 2u_2, \quad A_3 = u_0,$$

де  $u_0$ ,  $u_1$  і  $u_2$  – значення текстурної координати  $u$  у початковій, середній та кінцевій точках рядка растеризації (PP) відповідно.

Аналогічний вигляд мають формули і для розрахунку коефіцієнтів  $B_1 - B_3$ , однак замість значень координати  $u$  використовуються відповідні значення координати  $v$ .

Застосування квадратичної апроксимації забезпечує досить реалістичне відтворення перспективи порівняно нескладних обчисленнях. Недоліком наведених формул є те, що вони можуть бути використані лише у випадку нормалізованих значень екранних координат. Процедура нормалізації вимагає операції ділення та обчислення квадратного кореня, що суттєво впливає на обчислювальну складність.

Кубічна апроксимація [6-7] використовує залежність

$$\begin{aligned}u(x) &= C_1x^3 + C_2x^2 + C_3x + C_4, \\v(x) &= D_1x^3 + D_2x^2 + D_3x + D_4,\end{aligned}$$

де  $C_1-C_4$ ,  $D_1-D_4$  – коефіцієнти апроксимації, які розраховуються для кожного PP, значення координати  $x$  також нормалізовані.

Для розрахунку коефіцієнтів апроксимації необхідно знайти точні значення текстурних координат у чотирьох опорних точках: початковій, кінцевій та двох внутрішніх точок, які розбивають PP на три рівні відрізки. Кубічна апроксимація забезпечує більш реалістичне відтворення перспективи, ніж квадратична, але обчислювальна складність суттєво зростає, що обмежує застосування даного виду апроксимації у системах комп'ютерної графіки, які працюють у режимі реального часу.

При бі-квадратичній [6-7] апроксимації використовуються рівняння

$$\begin{aligned}u(x) &= A_1x^2 + A_2y^2 + A_3xy + A_4x + A_5y + A_6, \\v(x) &= B_1x^2 + B_2y^2 + B_3xy + B_4x + B_5y + B_6,\end{aligned}$$

де  $A_1-A_6$ ,  $B_1-B_6$  – коефіцієнти апроксимації.

Для розрахунку дванадцяти коефіцієнтів біквадратичної апроксимації потрібно знати точні значення пари текстурних координат у шести точках: у вершинах трикутного полігона та у середніх точках його ребер.

Бікубічна апроксимація [5-7] використовує рівняння

$$\begin{aligned}u(x) &= A_1x^3 + A_2y^3 + A_3x^2y + A_4xy^2 + A_5x^2 + A_6y^2 + A_7xy + A_8x + A_9y + A_{10}, \\v(x) &= B_1x^2 + B_2y^2 + B_3x^2y + B_4xy^2 + B_5x^2 + B_6y^2 + B_7xy + B_8x + B_9y + B_{10},\end{aligned}$$

де  $A_1-A_{10}$ ,  $B_1-B_{10}$  – коефіцієнти апроксимації.

Бікубічна апроксимація вимагає точних значень пари текстурних координат у 10 контрольних точках і передбачає розрахунок 20 коефіцієнтів апроксимації. Отриманий результат виглядає досить реалістично, але обчислювальні витрати дуже великі, тому цей вид апроксимації має обмежене використання.

Таким чином, існуючі методи апроксимації формул (2) не забезпечують достатньо прийняттого співвідношення між підвищенням продуктивності обчислень і збереженням точності обчислень. Тому існує потреба у розробці методів підвищення продуктивності перспективно-коректного текстурування зі збереженням точності обчислення текстурних координат.

Розглянемо реалізацію формули (2).

$$\text{Якщо } x_{i+1} = x_i + I, \text{ то } u_{i+1} = \frac{A_l \cdot (x_i + I) + B_l \cdot y_i + C_l}{D \cdot (x_i + I) + E \cdot y_i + F} = \frac{(A_l \cdot x_i + B_l \cdot y_i + C_l) + A_l \cdot I}{(D \cdot x_i + E \cdot y_i + F) + D \cdot I}.$$

Аналогічну формулу можна записати й для  $v_{i+1}$ . З наведених формул видно, що для розрахунку кожного текселя необхідно виконати 2 операції ділення та 3 операції додавання.

При реалізації текстурування [9-15] передбачає виконання «довгих» операцій, що обмежує швидкість формування графічних сцен і ускладнює апаратну реалізацію.

Зменшити кількість операцій ділення при текстуруванні можливо за рахунок растеризації об'єкту в світовій системі координат [11] за умови, що рядки растеризації на кожному етапі розміщено на фіксованій відстані від спостерігача. Знайдемо коефіцієнт нахилу прямої в екранній системі координат, якому відповідає відрізок прямої у світовій системі координат зі сталим для рядка значенням координати  $z$  (рис. 2). Нехай у світовій системі координат задано трикутник. Він однозначно визначає площину, рівняння якої має такий вигляд [5]

$$AX_w + BY_w + CZ_w = D,$$

де  $A$ ,  $B$ ,  $C$  – коефіцієнти, які визначаються координатами вершин трикутника.

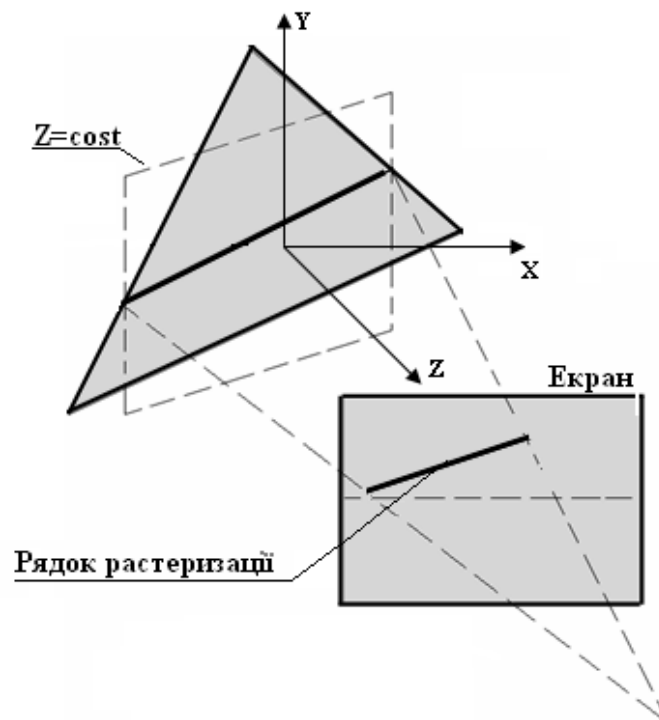


Рисунок 2– Рядки растеризації у світовій та екранній системах координат

З останнього рівняння знаходимо, що

$$Z_w = \frac{A \cdot X_w + B \cdot Y_w + D}{C}. \quad (3)$$

Між екранними та світовими координатами має місце такі співвідношення  $X_v = \frac{X_w}{Z_w}$ ,  $Y_v = \frac{Y_w}{Z_w}$ .

Запишемо останні рівняння у вигляді

$$X_w = X_v \cdot Z_w, \quad Y_w = Y_v \cdot Z_w.$$

Будемо шукати кут нахилу прямої, рівняння якої має вигляд  $Y_v = k \cdot X_v + h$ .

З урахуванням цього запишемо, що  $Y_w = Y_v \cdot Z_w = (k \cdot X_v + h) \cdot Z_w$ .

Підставивши у рівняння (3) отримані значення  $X_w$ ,  $Y_w$ , отримуємо

$$A \cdot X_v \cdot Z_w + B \cdot (k \cdot X_v + h) \cdot Z_w + C \cdot Z_w = D.$$

З останнього рівняння знаходимо, що

$$Z_w = \frac{D}{X_v \cdot (A + B \cdot k) + B \cdot h + C}. \quad (4)$$

За умови, що для рядка растеризації трикутника в екранній системі координат  $Z_w = const$ , то для любого  $j$  має місце рівняння

$$\frac{D}{X_v \cdot (A + B \cdot k) + B \cdot h + C} = \frac{D}{(X_v + j) \cdot (A + B \cdot k) + B \cdot h + C}.$$

Останнє рівняння має єдиний розв'язок  $k = -A/\hat{A}$ . Оскільки  $j$  і  $X_v$  було вибрано довільно, то можна констатувати, що нахил скануючого рядка растеризації не змінюється для всього трикутника, який розглядається в екранній системі координат. Значення координати  $Z_w$  для заданого рядка растеризації легко знайти, підставивши отримане значення  $k$  у рівняння (4).

$$Z_w = \frac{D}{B \cdot h + C}.$$

Між екранним та текстурним координатним простором існує така залежність [9-14]:

$$u = \frac{a \cdot X_v + b \cdot Y_v + c}{A \cdot X_v + B \cdot Y_v + C}, \quad v = \frac{d \cdot X_v + e \cdot Y_v + f}{A \cdot X_v + B \cdot Y_v + C}.$$

Позначимо знаменник наведених виразів через  $T$ , а  $\frac{1}{T} = \mathfrak{R}$ , тоді

$$u = (a \cdot X_v + b \cdot Y_v + c) \cdot \mathfrak{R}, \quad v = (d \cdot X_v + e \cdot Y_v + f) \cdot \mathfrak{R}.$$

Знаменник для визначення текстурних координат  $u$ ,  $v$  є сталим для рядка растеризації, у той час, як при звичайному підході він обчислюється для кожної точки поверхні. У подальшому будемо розглядати тільки одну з координат, наприклад  $u$ , оскільки вирази для їх розрахунку аналогічні. Виразимо  $Y_v$  через  $k$  і підставимо його в попередній вираз. Отримаємо

$$u = (a \cdot X_v + b \cdot (k \cdot X_v + h) + c) \cdot \mathfrak{R} = [X_v \cdot (a + b \cdot k) + (b \cdot h + c)] \cdot \mathfrak{R}.$$

Для початкової точки рядка растеризації  $X_v = 0$ . Із урахуванням цього

$$u_0 = \mathfrak{R} \cdot (b \cdot h + c).$$

Розглянемо, як змінюються  $u$  при зміні координати  $X_v$  на одиницю.

$$u_{i+j} = [(X_v + 1) \cdot (a + b \cdot k) + (b \cdot h + c)] \cdot \mathfrak{R} = u_i + \mathfrak{R} \cdot (a + b \cdot k).$$

Отримане співвідношення легко обчислити апаратним шляхом за умови, що  $\mathfrak{R}$  – відомо.

Важливим є питання растеризації трикутника, оскільки неортогональний по відношенню до координатних осей напрямок растеризації неминуче призведе до артефактів – наявності «просікань» і дублювання точок із-за зміщення по абсцисі початку рядків растеризації. Цього можна уникнути за умови, що провідне ребро трикутника буде паралельним осі ординат, однак це передбачає виконання особливої триангуляції поверхні і не відповідає вимогам графічних стандартів. Виключити артефакти можна за рахунок адаптивного фазування послідовності крокових приростів рядка сканування, що передбачає встановлення при інтерполюванні різних початкових значень оцінювальної функції, що суттєво ускладнює лінійний інтерполятор.

Найбільш просто вирішити колізію можна за рахунок растеризації не трикутника, а прямокутника, у який його віртуально вписано (рис. 3, а). Визначити параметри такого прямокутника можна шляхом порівняння координат вершин трикутника (крайні ліва та права вершини трикутника визначають абсцису лівої та правої сторін прямокутника, а нижня – ординати нижньої сторони прямокутника). Параметр  $r$  легко знайти, підставивши у рівняння прямої, яку використано для растеризації, значення абсциси верхньої вершини трикутника. Зрозуміло, що ордината верхньої лівої вершини прямокутника дорівнює суммі  $r$  і ординати верхньої вершини трикутника.

При растеризації прямокутника за умови неортогонального напрямку растеризації такі артефакти, як «просікання» і «залипання», будуть відсутні, оскільки крокові прирости сусідніх рядків будуть ідентичні вздовж ординатних рівнів.

Растеризації прямокутника виконується з метою визначення координат лівої і правої точок ребра трикутника, які перетинають рядок растеризації, без обчислень, які при реалізації потребують «довгих» операцій. При досягненні правого ребра трикутника здійснюється перехід на новий рядок растеризації прямокутника, який розміщено на один ординатний рівень нижче (рис. 3, а), тобто ділянка прямокутника за правим поточковим ребром не растеризується (рис. 3, б). Визначення координат лівого та правого ребер трикутника досягається шляхом порівняння кольору переднього плану, яким накреслено трикутник, з еталоном. При цьому можна, наприклад, використати принцип критерію парності [4], згідно з яким кількість перетину полігону є парним числом. Для підвищення швидкодії можна запропонувати растеризацію ділянок, які мають колір фону, імпульсною послідовністю підвищеної частоти. Безпосереднє текстуровання виконують з тактовою частотою, на якій працює відеопам'ять.

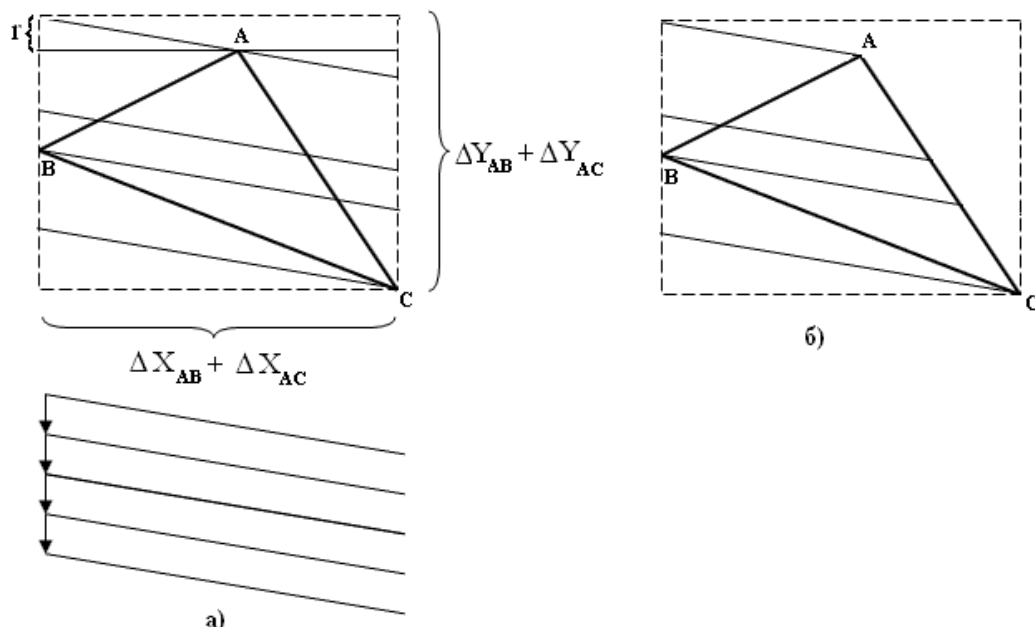


Рисунок 3 – Растрівання ділянки, обмеженої трикутником

Пристрій, який реалізує запропонований підхід до растерування (рис. 4), включає лінійний інтерполятор ЛІ, блок крокових приростів БКП, реєстр переднього плану РГп, реєстр РГХп і лічильник ЛЧ Уп початкової точки вікна, координатні лічильники ЛЧх, Лчу, блок керування текстурованням БКП.

Перед текстурованням у допоміжній пам'яті «викреслюють» кольором переднього плану вихідний трикутник. У лінійний інтерполятор записуються прирости  $\Delta X$ ,  $\Delta Y$  і знаки вектора, який визначає рядок растеризації прямокутника. У блок БКТ заносяться координати вершин трикутника. У регістр РГХп і лічильник ЛЧУп, лічильники ЛЧх і ЛЧу записуються координати лівої вершини прямокутника, з якої буде починатися растеризація прямокутника. В регістр РГпп заносяться значення інтенсивності кольору, яким накреслено трикутник. Ці дії утворюють цикл підготовки.

У подальшому лінійний інтерполятор і блок БКП генерують крокові прирости, під дією яких формуються координати точок рядка растеризації.

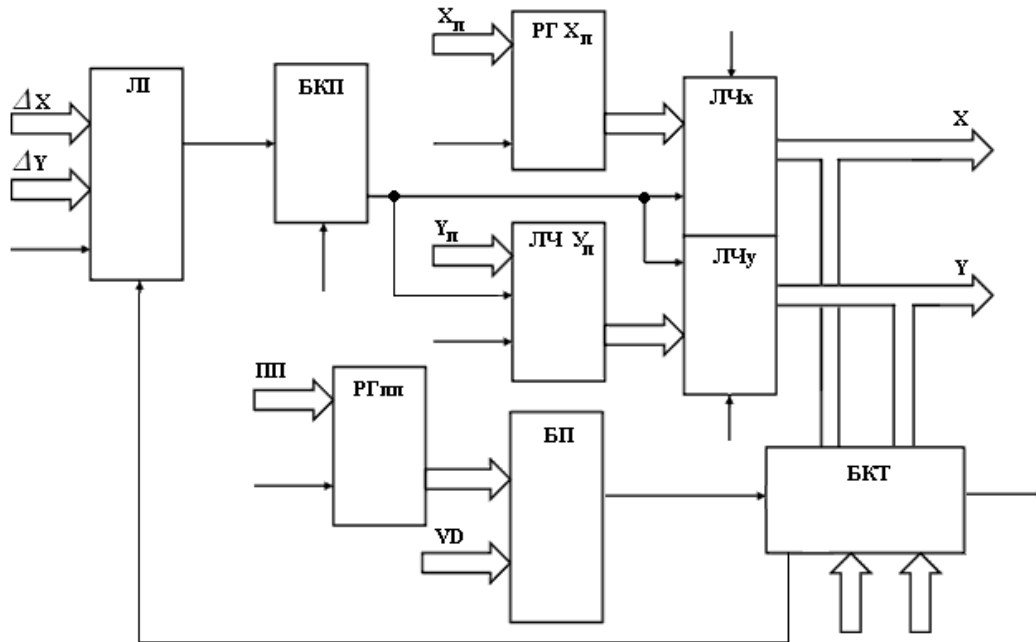


Рисунок 4 – Структурна схема блоку растеризації

Значення крокових приростів поступають на лічильники ЛЧх, Лчу, на виходах яких формуються координати точок вікна, які є одночасно адресами пам'яті. З виходу VD пам'яті зчитуються значення точок, які порівнюються блоком БП зі значенням регістра переднього плану. При співпадінні кольорів формується відповідний сигнал, який поступає в блок БКТ. Блок БКТ дозволяє текстуровання в проміжок часу між формуванням блоком порівняння сигналів рівності операндів із шини VD і регістра РГпп. Ці дії відповідають заповненню по критерію парності [4]. При цьому в блоці БКТ порівнюються значення адрес X і Y зі значеннями вершин трикутника, оскільки при перетині рядком растеризації тільки однієї вершини кількість перетинів ребер трикутника буде парним. У цьому випадку текстуровання не проводять. Растеризація рядка трикутника, який є частиною рядка прямокутника, завершують при перетині рядком правого ребра трикутника. При цьому блоком БКТ формується відповідний сигнал лінійному інтерполятору, який генерує разом із БКТ сигнал переходу на один ординатний рівень униз. Растеризацію трикутника завершують за умови, що потоковий рядок растеризації лежить нижче нижнього рядка трикутника. Після формування рядка растеризації трикутника стан лічильника ЛЧУп зменшують на одиницю. Після цього в лічильники ЛЧх, Лчу, заносять значення операндів, які зберігаються в регістрі РГХп та лічильнику ЛЧУп.

### Висновки

Запропоновано метод підвищення продуктивності перспективно-коректного нанесення текстури за рахунок використання неортогонального напрямку растеризації ділянки, обмеженої полігоном. Растеризування об'єкту в світовій системі координат здійснюється за умови, що рядки растеризації розміщено на фіксованій відстані від спостерігача. Аналіз показав, що для трикутника, який включає T внутрішніх точок, вилучається (T-q) операцій ділення, де q- кількість горизонтальних рядків растеризації трикутника.

### Список літератури

- [1] О. Н. Романюк, *Комп'ютерна графіка: навч. Посіб.* Вінниця, Україна: ВДТУ, 2001, 130 с.
- [2] О. Н. Романюк, та А. В. Чорний, *Високопродуктивні методи та засоби зафарбовування тривимірних графічних об'єктів. Монографія.* Вінниця, Україна: УНІВЕСУМ-Вінниця, 2006, 190 с.

- [3] О. Н. Романюк, та О. О. Дудник, «Підвищення реалістичності зафарбовування тривимірних графічних об'єктів», *Вісник ХНТУ*, № 3, с. 269-272, 2016.
- [4] О. Н. Романюк, М. Д. Обідник, О. В. Романюк, та Н. С. Костюкова, «Особливості архітектурної побудови систем формування тривимірних зображень», *Наукові праці Донецького національного технічного університету. Серія «Інформатика, кібернетика та обчислювальна техніка»*, 2010, № 12(165), с. 87-93.
- [5] Д. Херн, и М. П. Бейкер, *Компьютерная графика и стандарт OpenGL, 3-е издание*. Издательский дом «Вильямс», 2005, 1168 с.
- [6] А. М. Данилов, и И. А. Гарькина, *Интерполяция, аппроксимация, оптимизация: анализ и синтез сложных систем. Монография*. 2014.
- [7] Е. Янке, Ф. Эмде, и Ф. Леш, *Специальные функции: Формулы, графики, таблицы*. Directmedia, 2016.
- [8] Д. С. Яковлев, и М. Н. Фаворская, «Технологии фильтрации текстур», *Актуальные проблемы авиации и космонавтики*, 2011.
- [9] О. О. Дудник, та О. Н. Романюк, «Аналіз методів фільтрації текстур», *Матеріали міжнародної науково-практичної Інтернет-конференції «Молодь в технічних науках: дослідження, проблеми, перспективи» (МТН-2015)*, Вінниця, 16-17 квітня 2015 р. [Електронний ресурс]. Режим доступу: <http://conf.inmad.vntu.edu.ua/fm/index.php?page=materials&line=11&mat=115>. Дата звернення: Лют. 10, 2021.
- [10] О. Н. Романюк, та О. О. Дудник, «Метод підвищення продуктивності перспективно-коректного текстурування», *Наукові праці ДонНТУ*, № 1 (22), с. 43-46, 2016.
- [11] О. Н. Романюк, та О. О. Дудник, «Підвищення продуктивності текстурування з виконанням процедурних операцій в об'єктному просторі», *Наукові праці ДонНТУ. Серія «Інформатика, кібернетика та обчислювальна техніка»*, № 2 (23), с. 45-51, 2016.
- [12] О. Н. Романюк, та О. О. Дудник, «Розробка методів текстурування для задач фотореалістичного рендерингу», *Матеріали сьомої міжнародної науково-технічної конференції «Моделювання і комп'ютерна графіка»*, 18-24 вересня 2017 р, с. 26-33, 2017.
- [13] О. Н. Романюк, та О. О. Дудник, «Підвищення реалістичності зафарбовування тривимірних графічних об'єктів», *Вісник ХНТУ*, № 3 (58), с. 269-272, 2016.
- [14] A. N. Romanyuk, and O. O. Dudnyk, «Ways to improve performance of anisotropic texture filtering», *Control and Communications (SIBCON), 2017 International Siberian Conference on. IEEE, 2017*, [Online]. Available: <https://ieeexplore.ieee.org/document/7998589>. Accessed on: February 01, 2021.
- [15] О. Н. Романюк та О. О. Дудник, «Підвищення продуктивності перспективно-коректного текстурування з використанням анізотропної фільтрації», *Вимірювальна та обчислювальна техніка в технологічних процесах*, № 3, с. 192-195, 2016.

Стаття надійшла: 15.02.2021.

#### References

- [1] O. N. Romaniuk, *Kompiuterna hrafika: navch. Posib*. Vinnytsia Ukraina: VDTU, 2001, 130 s.
- [2] O. N. Romaniuk, та A. V. Chorny, *Vysokoproduktyvni metody ta zasoby zafarbovuvannia tryvy mirnykh hrafichnykh obiektiv. Monohrafiia*. Vinnytsia, Ukraina: UNIVESUM-Vinnytsia, 2006, 190 s.
- [3] O. N. Romaniuk, та O. O. Dudnyk, «Pidvyshchennia realistychnosti zafarbovuvannia tryvymirnykh hra-fichnykh obiektiv», *Visnyk KhNTU*, № 3, s. 269-272, 2016.
- [4] O. N. Romaniuk, M. D. Obidnyk, O. V. Romaniuk, та N. S. Kostyukova, «Osoblyvosti arkhitekturoi pobudovy system formuvannia tryvymirnykh zobrazhen», *Naukovi pratsi Donetskoho natsionalno-ho tekhnichnoho univertsytetu. Serii «Informatyka, kibernetyka ta obchysliuvalna tekhnika»*, 2010, № 12(165), с. 87-93.
- [5] D. Hern, i M. P. Bejker, *Komp'juternaja grafika i standart OpenGL, 3-e izdanie*. Izda-tel'skij dom «Vil'jams», 2005, 1168 s.
- [6] A. M. Danilov, i I. A. Gar'kina, *Interpoljacija, approksimacija, optimizacija: analiz i sintez slozhnyh sistem. Monografija*. 2014.
- [7] E. Janke, F. Jemde, i F. Lesh, *Special'nye funkcii: Formuly, grafiki, tablicy*. Directmedia, 2016.
- [8] D. S. Jakovlev, i M. N. Favorskaja, «Tehnologii fil'tracii tekstur», *Aktual'nye problemy aviacii i kosmonavтики*, 2011.
- [9] O. O. Dudnyk, та O. N. Romaniuk, «Analiz metodiv filtratsii tekstur», *Materialy mizhnarodnoi naukovo-praktychnoi Internet-konferentsii «Molod v tekhnichnykh naukakh:doslidzhennia, problemy, perspektivy» (MTN-2015)*, Vinnytsia, 16-17 kvitnia 2015 r. [Elektronnyi resurs]. Rezhym dostupu: <http://conf.inmad.vntu.edu.ua/fm/index.php?page=materials&line=11&mat=115>. Data zvernennia: Liut. 10, 2021.
- [10] O. N. Romaniuk, та O. O. Dudnyk, «Metod pidvyshchennia produktyvnosti perspektyvno-korektnoho teksturuвання», *Naukovi pratsi DonNTU*, № 1 (22), s. 43-46, 2016.



- [11] O. N. Romaniuk, ta O. O. Dudnyk, «Pidvyshchennia produktyvnosti teksturuвання z vykonanniam pro-tsedurnykh operatsii v obiektnomu prostori», *Naukovi pratsi DonNTU. Seriiia «Informatyka, kiber-netyka ta obchysliuvalna tekhnika»*, № 2 (23), s. 45-51, 2016.
- [12] O. N. Romaniuk, ta O. O. Dudnyk, «Rozrobka metodiv teksturuвання dlia zadach fotorealistychnoho renderynhu», *Materialy somoi mizhnarodnoi naukovo-tekhnichnoi konferentsii «Modeliuвання i kompiuterna hrafiika»*, 18-24 veresnia 2017 r, s. 26-33, 2017.
- [13] O. N. Romaniuk, ta O. O. Dudnyk, «Pidvyshchennia realistychnosti zafarbovuvannia tryvymirnykh hra-fichnykh obiektyv», *Visnyk KhNTU*, № 3 (58), s. 269-272, 2016.
- [14] A. N. Romanyuk, and O. O. Dudnyk, «Ways to improve performance of anisotropic texture filtering», *Control and Communications (SIBCON), 2017 International Siberian Conference on. IEEE*, 2017, [Online]. Available: <https://ieeexplore.ieee.org/document/7998589>. Accessed on: February 01, 2021.
- [15] O. N. Romaniuk ta O. O. Dudnyk, «Pidvyshchennia produktyvnosti perspektyvno-korektnoho teksturuвання z vykorystanniam anizotropnoi filtratsii», *Vymiriuvalna ta obchysliuvalna tekhnika v tekhnolohichnykh protsesakh*, № 3, s. 192-195, 2016.

#### **Відомості про авторів**

**Романюк Олександр Нікіфорович** – доктор технічних наук, професор, завідувач кафедри програмного забезпечення.

**Дудник Олександр Олександрович** – кандидат технічних наук, доцент, доцент кафедри програмного забезпечення.

**Романюк Оксана Володимирівна** – кандидат технічних наук, доцент, доцент кафедри програмного забезпечення.

**Озерчук Дмитро Анатолійович** – магістрант кафедри програмного забезпечення.

A. N. Romanyuk, A. A. Dudnik, O. V. Romanyuk, D. A. Ozerchuk

## **МЕТОД ПЕРСПЕКТИВНО-КОРЕКТНОГО ТЕКСТУРИВАННЯ**

Винницький національний технічний університет, Вінниця

O. N. Romanyuk, O. O. Dudnik, O. V. Romanyuk, D. A. Ozerchuk

## **PERSPECTIVE CORRECT TEXTURING METHOD**

Vinnitsia National Technical University, Vinnitsia