

Хотя, радиационно-стойкие ПЛИС высокой интеграции сейчас весьма недешевы, равно как и описанные выше средства разработки, эти затраты сопоставимы со стоимостью решений на основе радиационно-стойких процессоров и компенсируются сокращением косвенных затрат на разработку и поддержанием проектов в долгосрочной перспективе, и сокращением времени готовности новых решений от момента получения технических требований на разработку (time-to-market), что становится особенно важным в условиях рыночной конкуренции разработчиков электронной аппаратуры.

### Заключение

Существенное снижение в последние годы стоимости ПЛИС делает их использование экономически эффективными даже в серийном производстве. Развитие технологии производства интегральных микросхем к настоящему времени сделало доступными для массового потребителя технически совершенные и производительные программируемые логические матрицы высокой интеграции. Появление подобных изделий позволяет пересмотреть роль и значение ПЛИС в современной электронной системе сбора и обработки информации. Превращение программируемых матриц из вспомогательного средства компактного размещения цифровой «логики» в системообразующий элемент открывает новые технические возможности в разработке бортовых систем и дает возможность решить ряд экономических и организационных проблем, связанных с применением традиционных решений типа «система на плате». Применение ПЛИС позволяет существенно уменьшить зависимость предприятий, разработчиков электронной аппаратуры, от производителей и заложить крепкий фундамент для развития современных вычислительных комплексов реального времени.

### Литература

1. Косткин М. Архитектурные и схмотехнические решения вычислительно-управляющих комплексов на основе микросхем программируемой логики / Косткин М., Поздняков П., Попович А // Компоненты и технологии. – 2008. – № 5. С. – 62 – 65.
2. Spartan-6. Lowest Power High-Performance FPGAs. Data Sheet. – ver. 2. – 2009, February. – P. 5-12.
3. Леклидер Т. Погружаясь в ПЛИС / Леклидер Т // Компоненты и технологии. – 2006. – № 12. – С. – 59-64.
4. Карпов С. Разработка систем на кристалле на базе ПЛИС Actel [Режим доступа] – <http://www.actel.com>
5. Муравьев В. Микроспутники. Краткое описание // Новости космонавтики. – 2004. – № 4. – С.24-36.

Надійшла до редакції  
10.6.2010 р.

УДК 681.323

**В.М. ГУСЯТИН, Я.В. ЧАГОВЕЦ, Д.Г. КОЖУШКО**

Харьковский национальный университет радиоэлектроники

## МЕТОД БЫСТРЫХ ВЫЧИСЛЕНИЙ ДЛЯ ЗАДАЧИ АНИЗОТРОПНОЙ ФИЛЬТРАЦИИ ПРИ СИНТЕЗЕ ИЗОБРАЖЕНИЙ ОБРАТНЫМ ТРАССИРОВАНИЕМ

В статье предложен метод быстрых вычислений для задачи анизотропной фильтрации текстур. Суть метода заключается в представлении текстуры в виде кватернарного дерева предвычисленных значений взвешенных цветов, последующем проходе по такому дереву с суммированием значений в вершинах, попавших внутрь проекции пикселя на плоскость текстуры и нахождении интегрального цвета пикселя. Предложены соотношения для расчета взвешенных цветов в вершинах кватернарного дерева для случая выполнения фильтрации функцией *Box* и функцией *Courant* (Курант).

*The fast computation method for the anisotropic texture filtering is proposed. According to the method the texture is represented as a quad-tree of the precomputed values of weighted colors and thereafter quad-tree is traversed with a summation of values at the vertices lain inside the projection of a pixel on the texture plane, thus computing an integral pixel color. Proportion of the amount of such colors to the volume of the filter function is an integral color of pixel projection. Relations for computation of weighted colors at the vertices of a quad-tree are proposed for the filtering function of the Box and the Kurant.*

Ключевые слова: текстура, алиасинг, пиксел.

### Введение

Исследования ведущих специалистов в области компьютерной графики показывают, что метод обратного трассирования (ray tracing) позволяет синтезировать изображения высокой реалистичности [1,2,3]. Однако, при использовании этого метода, как и при использовании метода прямого трассирования, существует проблема устранения эффекта алиасинга (aliasing), явление которого в процессе синтеза

ізображений різко знижає їх реалістичність. В зв'язі з цим розробка методів усунення аліасинга при синтезі ізображень обратним трассированием являється актуальною проблемою.

В роботах [4,5] використовуються методи фільтрації текстур, які не учитують форму проєкції пікселя на площину текстури, що призводить до надлишкового розмиття ізображення. В роботі [6] використовується підхід, оснований на методі анізотропної фільтрації, при якому використовуються додаткові вибірки, розташовані залежно від положення спостерігача відносно сцени. Такий підхід в разі використання обратного трассирования неадекватен, так як вимагає додатково трассированих променів [2]. В відміння від [2,6], метод анізотропної фільтрації, викладений в статті авторів [7], не вимагає додатково трассированих променів. В той же час, при виконанні фільтрації в площині текстури кількість вибірок з пам'яті та арифметических операцій зростає пропорційно збільшенню роздільності текстури. Метою статті являється розробка методу швидких обчислень, що зменшує кількість вибірок з пам'яті та арифметических операцій при виконанні анізотропної фільтрації на основі методу, викладеного в статті авторів [7].

Згідно з запропонованим в роботі [7] методом колір кожного пікселя визначається як інтегральний колір проєкції пікселя на площину текстури згідно з співвідношенням:

$$R^* = \int_{-\infty-\infty}^{+\infty+\infty} T(x_T, z_T) H^{Ys, Zs}(x_T, z_T) dx_T dz_T. \quad (1)$$

Авторами розглянуті два варіанти фільтрації, для кожного з яких були отримані відповідні співвідношення вагової функції  $H^{Ys, Zs}$ .

В комп'ютерній графіці текстура задається, як правило, на основі матриці кольорів  $M = \{C_{i,j}\}$ , де  $C_{i,j}$  – значення кольорів для  $i, j$ -ячейки;  $i = 0..N-1, j = 0..N-1$  – цілі числа;  $N = 2^n$ ,  $n$  – ціле число,  $n \geq 0$ . Використовуючи елементи матриці, задамо функцію  $T(x_T, z_T)$ , для якої координати  $\{x_T, z_T\}$  нормалізовані і знаходяться в інтервалі  $[0,1)$  (єдинична текстура) в вигляді співвідношення:

$$T(x_T, z_T) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C_{i,j} b_0(x_T N - i) b_0(z_T N - j), \quad (2)$$

де:

$$b_0(t) = \begin{cases} 1, & \text{якщо } t \in [0,1), \\ 0, & \text{якщо } t \notin [0,1). \end{cases}$$

#### Постановка задачі дослідження

Для швидкого обчислення (1) в разі, коли текстура задана згідно (2), скористаємося способом задавання текстури, викладеним в роботі авторів [8] в вигляді кватернарного дерева. Кожна вершина дерева може містити інформацію про чотири дочірні піддерев'я, така вершина називається вузлом. Вершина, яка не містить інформації про дочірні піддерев'я, називається листовою.

Пропонується в кожній вершині дерева зберігати інформацію, необхідну для швидкого обчислення (1). Вершина дерева відповідає області текстури, маючій форму квадрата, яка, в свою чергу, може ділитися на чотири квадрата меншого розміру. Кватернарне дерево складається з кінцевого множини таких вершин. Швидке обчислення інтегрального кольору проєкції пікселя пропонується виконувати наступним чином. Визначаються квадрати, відповідні вершинам, повністю потрапивши всередину проєкції пікселя, і в процесі пошуку визначається сума значень кольорів, збережених в цих вершинах, зважених на відповідну вагову функцію. На основі цієї суми обчислюється інтегральний колір (1).

Вид інформації, збереженої в вершинах дерева, відрізняється в залежності від типу використовуваної вагової функції  $H^{Ys, Zs}(x_T, z_T)$ . Розглянемо запропонований метод для випадку вагової функції нульової (Вох) і першої (функція Куранта) ступеня.

#### Основні співвідношення для випадку фільтрації функцією Вох

В статті [7] при знаходженні проєкції пікселя на площину текстури було запропоновано апроксимувати реальну форму вагової функції, побудованої в координатах площини текстури, функцією виду (рис. 1).

$$H_B^{Ys, Zs}(x_T, z_T) = \begin{cases} K_B, & \text{якщо } f_k(x_T, z_T) = 1 \forall k = \overline{1..4}, \\ 0, & \text{якщо } f_k(x_T, z_T) \neq 1 \exists k = \overline{1..4}, \end{cases}$$

де:  $f_k(x_T, z_T) = \text{sgn}(\mathbf{v}_{oy} \cdot (\mathbf{v}'_k(x_T, z_T) \times \mathbf{v}_k))$  – функція, що показує, з якої сторони від вектора, побудованого вздовж сторони проєкції пікселя, лежить точка  $q = \{x_T, z_T\}$  (рис. 1);

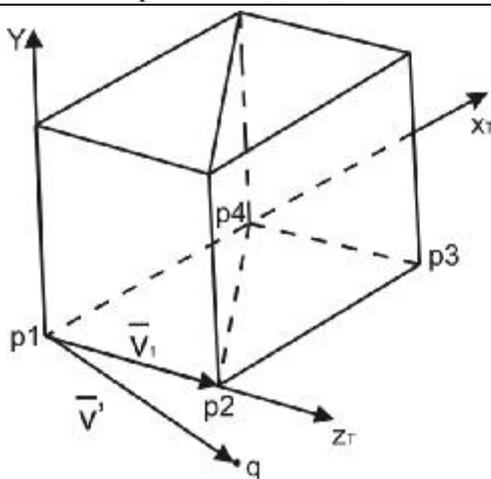


Рис. 1. Весовая функция нулевой степени (Box)

$\mathbf{v}_k = p_{k+1} - p_k$  – вектор, исходящий из  $k$ -й вершины, идущий вдоль стороны проекции пикселя и направленный таким образом, что проекция лежит слева от него;

$\mathbf{v}'_k(x_T, z_T) = q - p_k$  – вектор, исходящий из  $k$ -й вершины, идущий в направлении точки  $q$ ;

$p_1, p_2, p_3, p_4$  – проекции вершин пикселя на плоскость текстуры, взятые для определенности в таком порядке, что при обходе от  $p_k$  до  $p_{k+1}$  проекция пикселя остается слева (рис. 1);

$\mathbf{v}_{oy} = \{0, 1, 0\}$  – единичный вектор, направленный вдоль оси  $Y$ ;

$$\text{sgn}(x) = \begin{cases} 1, & \text{если } x < 0 \\ 0, & \text{если } x \geq 0 \end{cases}$$

Согласно предлагаемому подходу четырехугольник проекции пикселя делится на два треугольника  $p_1, p_2, p_4$  и  $p_2, p_3, p_4$ .

В этом случае (1) принимает вид:

$$\begin{aligned} R^* &= \int_{-\infty-\infty}^{+\infty+\infty} \int_{-\infty-\infty}^{+\infty+\infty} T(x_T, z_T) H_B^{Ys, Zs}(x_T, z_T) dx_T dz_T = \\ &= \int_{-\infty-\infty}^{+\infty+\infty} \int_{-\infty-\infty}^{+\infty+\infty} T(x_T, z_T) H_{\Delta_1}^{Ys, Zs}(x_T, z_T) dx_T dz_T + \int_{-\infty-\infty}^{+\infty+\infty} \int_{-\infty-\infty}^{+\infty+\infty} T(x_T, z_T) H_{\Delta_2}^{Ys, Zs}(x_T, z_T) dx_T dz_T, \end{aligned} \quad (3)$$

где:  $\Delta_1, \Delta_2$  – треугольные области интегрирования, на которые разбивается четырехугольная область  $D$  (проекция пикселя на плоскость текстуры);

$$\begin{aligned} H_{\Delta_1}^{Ys, Zs}(x_T, z_T) &= K_B (f_1(x_T, z_T) \wedge f_4(x_T, z_T) \wedge \text{sgn}(\mathbf{v}_{oy} \cdot (\mathbf{v}'_1(x_T, z_T) \times (p_4 - p_2))))), \\ H_{\Delta_2}^{Ys, Zs}(x_T, z_T) &= K_B (f_2(x_T, z_T) \wedge f_3(x_T, z_T) \wedge \text{sgn}(\mathbf{v}_{oy} \cdot (\mathbf{v}'_4(x_T, z_T) \times (p_2 - p_4))))). \end{aligned} \quad (4)$$

Здесь и далее под знаком  $\wedge$  следует понимать операцию конъюнкции.

Так как функции (4) в пределах треугольных областей  $\Delta_1, \Delta_2$  равны  $K_B$ , а за их пределами нулю, и интегрирование выполняется по треугольным областям, то (3) принимает вид:

$$R^* = K_B \left( \iint_{\Delta_1} T(x_T, z_T) dx_T dz_T + \iint_{\Delta_2} T(x_T, z_T) dx_T dz_T \right) = K_B (C_1^* + C_2^*), \quad (5)$$

$C_1^*, C_2^*$  – цвета, взвешенные по площади областей  $\Delta_1, \Delta_2$ .

В статье [7] было доказано, что  $K_B = 1/S_D = 1/(S_{\Delta_1} + S_{\Delta_2})$ . Подставим его в (5):

$$R^* = (C_1^* + C_2^*) / (S_{\Delta_1} + S_{\Delta_2}), \quad (6)$$

$S_{\Delta_1}, S_{\Delta_2}$  – площади треугольников  $\Delta_1, \Delta_2$ .

С учетом (2)  $C_1^*$  принимает вид:

$$C_1^* = \iint_{\Delta_1} \left( \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C_{i,j} b_0(x_T N - i) b_0(z_T N - j) \right) dx_T dz_T =$$

$$= O_1 + \sum_{k=0}^{P-1} \iint_{\Delta_1} C_{i(k),j(k)} b_0(x_T N - i(k)) b_0(z_T N - j(k)) dx_T dz_T = O_1 + \sum_{k=0}^{P-1} \frac{C_{i(k),j(k)}}{N^2},$$

где:  $O_1$  – сумма интегралов, учитывающая тексели, частично попавшие внутрь области интегрирования  $\Delta_1$ ;

$k$  – номер текселя, полностью попавшего внутрь области интегрирования;

$i(k), j(k)$  – координаты  $k$ -го текселя в текстуре;

$P$  – количество текселей, полностью попавших внутрь области интегрирования.

Аналогичные соотношения справедливы и для  $C_2^*$ .

Предлагается при расчете взвешенных цветов не учитывать тексели, частично попавшие внутрь треугольных областей интегрирования  $\Delta_1, \Delta_2$ . Тогда:

$$C_{1,2}^* \approx C_{1,2}^{**} = \sum_{k=0}^{P-1} \frac{C_{i(k),j(k)}}{N^2}.$$

В этом случае  $R^*$  предлагается вычислять приближенно по формуле:

$$R^* \approx R^{**} = (C_1^{**} + C_2^{**}) / (S_{\Delta_1} + S_{\Delta_2}), \quad (7)$$

Геометрический смысл величин  $C_1^{**}, C_2^{**}$  – объем подынтегральной функции текстуры, отсекаемый областями соответственно  $\Delta_1^*, \Delta_2^*$ , которые являются частью областей треугольников  $\Delta_1, \Delta_2$ , ограниченных площадью текселей, полностью попавших внутрь этих треугольников (рис. 2).

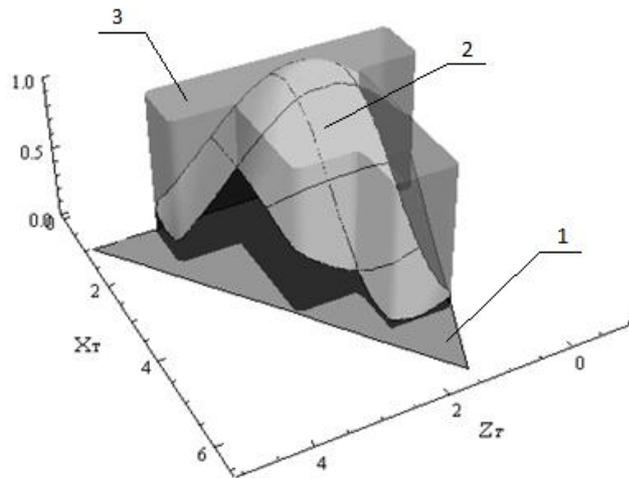


Рис. 2. Геометрический смысл элементов задачи нахождения интегрального цвета для случая функции фильтра  $H_B^{Ys,Zs}$

На рисунке 2 показана область  $\Delta_1$ , обозначенная “1”. Нашей целью является нахождение интегрального цвета этой области. На осях  $x_T, z_T$  заданы координаты в плоскости текстуры, а по оси  $Y$  условно отложен цвет текстуры. Поверхность, обозначенная “2”, является геометрической интерпретацией непрерывной функции текстуры, помноженной на весовую функцию. Через “3” обозначена весовая функция, высота которой равна единице. Учитываются квадраты (тексели текстуры), находящиеся внутри области  $\Delta_1^*$ . Объем, призматического тела, имеющего основанием один из квадратов и отсекаемого сверху функцией текстуры, численно равен взвешенному цвету данного квадрата (текселя текстуры). В процессе вычисления взвешенных цветов  $C_{1,2}^{**}$  находится сумма взвешенных цветов для всех текселей, полностью попавших внутрь треугольника, обозначенного “1”.

#### Основные этапы вычислений метода для случая фильтрации функцией Вох

Этап 1. Формирование дерева. Является подготовительным этапом и выполняется вне реального времени.

Введем обозначение:

$W_{i,j}^m$  – взвешенный цвет  $i, j$ -го текселя в  $m$ -м уровне. Здесь  $i = 0, 2^m - 1, j = 0, 2^m - 1$  – номера

текселей текстуры в пределах уровня.

На первом проходе формируется матрица  $M^* = \{w_{i,j}^n\}$  взвешенных по площади цветов верхнего уровня дерева, где  $w_{i,j}^n = C_{i,j} / 2^{2n}$ .

На втором и последующих проходах формируются последующие уровни дерева, взвешенный цвет  $w_{i,j}^m$  вершин которых вычисляется с помощью соотношения  $w_{i,j}^m = w_{2i,2j}^{m+1} + w_{2i+1,2j}^{m+1} + w_{2i+1,2j+1}^{m+1} + w_{2i,2j+1}^{m+1}$  для всех  $m = \overline{0, n-1}$ .

Таким образом, в каждой вершине образованного дерева хранится информация о взвешенном цвете участка текстуры, соответствующего данной вершине, которая используется для быстрого вычисления интегрального цвета проекции пикселя.

Этап 2. Нахождение интегрального цвета. Выполняется в реальном времени.

Для каждого из треугольников  $p_1, p_2, p_4$  и  $p_2, p_3, p_4$  выполняется алгоритм поиска участков текстуры (текселей), соответствующих вершинам дерева и полностью попавших внутрь треугольников (рис. 3).

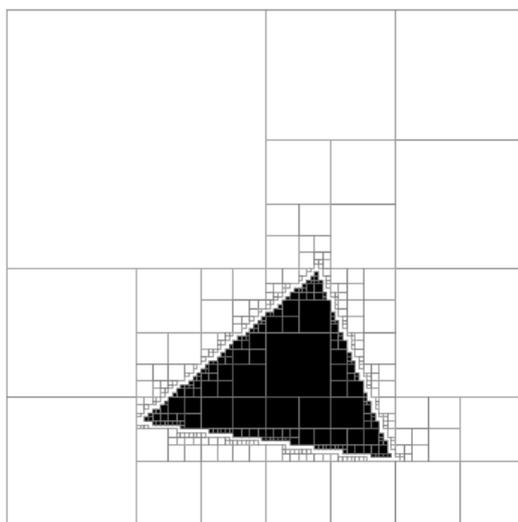


Рис. 3. Формирование области, например  $\Delta_1^*$ , из наибольших КК, попавших внутрь треугольника

Одновременно в процессе поиска вычисляют суммарный взвешенный цвет  $C_1^{**} + C_2^{**}$  проекции пикселя.

$$\begin{aligned} \text{sumBox}(p_1, p_2, p_3, p_4, \text{tree}) &= \text{sumTri}(p_1, p_2, p_4, \text{tree}) + \\ &+ \text{sumTri}(p_2, p_3, p_4, \text{tree}), \end{aligned} \quad (8)$$

где  $\text{sumTri}$  – функция, которая вычисляет суммарный взвешенный цвет текселей, полностью попавших в треугольник. В процессе вычисления функции выполняется проход – от уровней с малой детализацией к уровням с более подробной детализацией.

#### Основные математические соотношения для случая фильтрации функцией Куранта

Рассмотрим случай фильтрации функцией первой степени.

В статье [7] было предложено аппроксимировать реальную форму проекции фильтра на плоскость текстуры функцией вида (рис. 4).

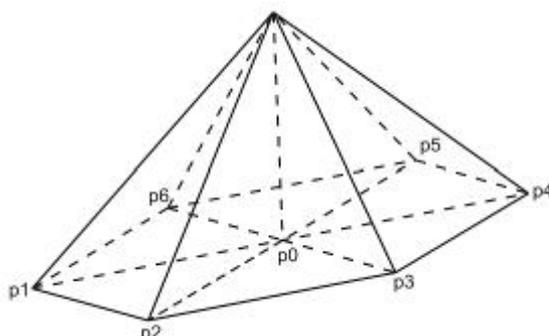


Рис. 4. Функция Куранта  $H_T^{Y_s, Z_s}$

$$H_T^{Ys,Zs}(x_T, z_T) = K_T \cdot \begin{cases} plane_1(x_T, z_T), \text{ если } F_1(x_T, z_T) = 1 \\ plane_2(x_T, z_T), \text{ если } F_2(x_T, z_T) = 1 \\ plane_3(x_T, z_T), \text{ если } F_3(x_T, z_T) = 1 \\ plane_4(x_T, z_T), \text{ если } F_4(x_T, z_T) = 1, \\ plane_5(x_T, z_T), \text{ если } F_5(x_T, z_T) = 1 \\ plane_6(x_T, z_T), \text{ если } F_6(x_T, z_T) = 1 \\ 0, \text{ иначе} \end{cases} \quad (9)$$

где:  $plane_k(x_T, z_T) = D_k \cdot (x_T \cdot A_k + z_T \cdot B_k + C_k)$  – одна из шести боковых граней пирамиды,

$$A_k = z_k - z_{k+1}, B_k = x_{k+1} - x_k, C_k = x_k z_{k+1} - x_{k+1} z_k,$$

$$D_k = \frac{1}{x_{k+1} z_0 - x_{k+1} z_k + z_k x_0 + x_k z_{k+1} - z_{k+1} x_0 - x_k z_0},$$

$x_k, z_k$  – координаты точки  $p_k$ ,

$x_{k+1}, z_{k+1}$  – координаты точки  $p_{k+1}$ ,

$$F_k(x_T, z_T) = f_k(x_T, z_T) \wedge f'_k(x_T, z_T) \wedge f''_k(x_T, z_T),$$

$$f_k(x_T, z_T) = \text{sgn}(\mathbf{v}_{oy} \cdot (\mathbf{v}'_k(x_T, z_T) \times \mathbf{v}_k)),$$

$$f'_k(x_T, z_T) = \text{sgn}(\mathbf{v}_{oy} \cdot (\mathbf{v}'_{k+1}(x_T, z_T) \times (-\mathbf{v}_{0,k+1}))),$$

$$f''_k(x_T, z_T) = \text{sgn}(\mathbf{v}_{oy} \cdot (\mathbf{v}'_0(x_T, z_T) \times \mathbf{v}_{0,k})),$$

$\mathbf{v}_{0,k} = p_k - p_0$  – вектор, исходящий из 0-й вершины, идущий в направлении  $k$ -ой вершины и направленный таким образом, что проекция лежит слева от него.

$\mathbf{v}'_k(x_T, z_T) = q(x_T, z_T) - p_k$  – вектор, исходящий из  $k$ -й вершины, идущий в направлении точки  $q(x_T, z_T)$ .

$\mathbf{v}'_0(x_T, z_T) = q(x_T, z_T) - p_0$  – вектор, исходящий из 0-й вершины, идущий в направлении точки  $q(x_T, z_T)$ .

$p_k$  – проекции центров семи смежных пикселей на плоскость текстуры [7] (рис. 4) ( $k = \overline{0..6}$ ).

Соотношение (9) будет выполняться, если на последнем шаге обхода примем  $p_7 = p_1$ .

В этом случае (1) принимает вид:

$$\begin{aligned} R^* &= \int_{-\infty-\infty}^{+\infty+\infty} \int T(x_T, z_T) H_T^{Ys,Zs}(x_T, z_T) dx_T dz_T = \\ &= \iint_D T(x_T, z_T) H_T^{Ys,Zs}(x_T, z_T) dx_T dz_T = \\ &= K_T \left( \iint_{\Delta_1} T(x_T, z_T) plane_1(x_T, z_T) dx_T dz_T + \dots + \right. \\ &\quad \left. + \iint_{\Delta_6} T(x_T, z_T) plane_6(x_T, z_T) dx_T dz_T \right) = K_T \sum_{k=1}^6 C_k^*, \end{aligned} \quad (10)$$

где:  $D$  – область интегрирования, образованная шестиугольником  $p_1 p_2 p_3 p_4 p_5 p_6$  (рис. 4).

$\Delta_1, \dots, \Delta_6$  – треугольные области интегрирования, на которые разбивается шестиугольная область  $D$ .

Запишем одно из слагаемых из (10):

$$C_k^* = \iint_{\Delta_k} T(x_T, z_T) plane_k(x_T, z_T) dx_T dz_T \quad (11)$$

– взвешенный цвет по площади  $\Delta_k$  с учетом весовой функции  $plane_k(x_T, z_T)$ .

Геометрический смысл величин  $C_1^*, \dots, C_6^*$  – объем подынтегральной функции текстуры, взвешенный на соответствующую весовую функцию  $plane_k(x_T, z_T)$ , отсекаемый областями  $\Delta_1, \dots, \Delta_6$  (рис. 5).

Согласно [7] подставим  $K_T$  в (10):

$$R^* = 3 \sum_{k=1}^6 C_k^* / \sum_{k=1}^6 S_{\Delta_k} \quad (12)$$

$S_{\Delta_k}$  – площади треугольников  $\Delta_k$ .

В нашем случае вместо областей  $\Delta_k$  учитываются их подобласти  $\Delta_k^*$  соответственно. В этом случае интегральный цвет записывается соотношением:

$$R^* \approx R^{**} = 3 \sum_{k=1}^6 C_k^{**} / \sum_{k=1}^6 S_{\Delta_k} \quad (13)$$

где:  $\Delta_k^*$  – область, образованная из текселей текстуры, полностью попавших внутрь  $\Delta_k$ ;

$C_k^{**}$  – имеет тот же смысл, что и  $C_i^*$ , но при их вычислении учитываются подобласти  $\Delta_k^*$ .

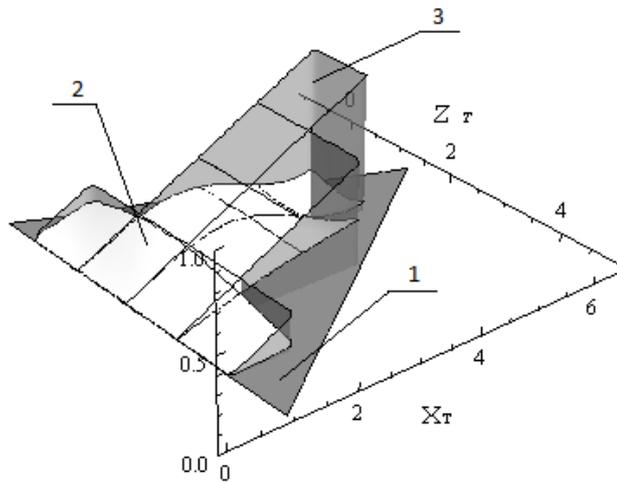


Рис. 5. Геометрический смысл элементов задачи нахождения интегрального цвета для случая функции фильтра  $H_T^{Ys, Zs}$

По аналогии с рис. 2 на рис. 5 показана область  $\Delta_1$ , обозначенная “1”. Нашей целью является нахождение интегрального цвета этой области. На осях  $x_T, z_T$  заданы координаты в плоскости текстуры, а по оси Y условно отложен цвет текстуры. Поверхность, обозначенная “2”, является геометрической интерпретацией функции текстуры, помноженной на весовую функцию. Через “3” обозначена весовая функция. Учитываются исключительно квадраты (тексели текстуры), полностью находящиеся внутри области  $\Delta_1^*$ . Объем призматического тела, имеющий основанием один из квадратов и отсекаемый сверху функцией, обозначенной “2”, является геометрической интерпретацией взвешенного цвета данного квадрата (текселя текстуры). В результате вычисления находится сумма таких взвешенных цветов для всех текселей, полностью попавших внутрь треугольника “1”.

Предлагается быстрый метод вычисления (11). Запишем:

$$plane_k(x_T, z_T) = a_k x_T + b_k z_T + c_k$$

Тогда (11) приобретает вид:

$$C_k^{**} = \sum_{r=0}^{L-1} \iint_{d_r} C_{i,j} (a_k x_T + b_k z_T + c_k) dx_T dz_T \quad (14)$$

где:  $L$  - количество текселей уровня  $n$ , полностью попавших внутрь области интегрирования  $\Delta_k$ ;

$d_r$  – область  $r$ -го текселя текстуры с координатами левой нижней вершины  $\{i/N, j/N\}$ , а правой верхней  $\{(i+1)/N, (j+1)/N\}$ , полностью попавшего внутрь области  $\Delta_k$ ;

$i, j$  – координаты  $r$ -го текселя в уровне  $n$  текстуры.

Вычислим интеграл под суммой в (14):

$$\iint_{d_r} C_{i,j} (a_k x_T + b_k z_T + c_k) dx_T dz_T = \int_{\frac{j}{N}}^{\frac{j+1}{N}} \int_{\frac{i}{N}}^{\frac{i+1}{N}} C_{i,j} (a_k x_T + b_k z_T + c_k) dx_T dz_T =$$

$$= a_k \cdot Ka_{i,j} + b_k \cdot Kb_{i,j} + c_k \cdot Kc_{i,j}$$

где:

$$Ka_{i,j} = C_{i,j} \frac{i + \frac{1}{2}}{N^3}; Kb_{i,j} = C_{i,j} \frac{j + \frac{1}{2}}{N^3}; Kc_{i,j} = \frac{C_{i,j}}{N^2}.$$

Коэффициенты  $Ka_{i,j}, Kb_{i,j}, Kc_{i,j}$  не зависят от наклона плоскости  $plane_k$  и могут быть заранее вычислены для каждого текселя текстуры. При этом предлагается в каждом текселе текстуры хранить кортеж  $K_{i,j} = (Ka_{i,j}, Kb_{i,j}, Kc_{i,j})$ .

### Основные этапы метода для случая фильтрации функцией Куранта

Этап 1. Формирование дерева. Является подготовительным этапом и выполняется вне реального времени.

Дерево в этом случае формируется в два прохода. На первом проходе формируется матрица кортежей  $M^* = \{K_{i,j}^n\}$ . Верхним индексом обозначим номер уровня. На втором проходе формируется кватернарное дерево кортежей. Каждый последующий кортеж дерева  $K_{i,j}^m$  вычисляется с помощью следующего соотношения:

$$K_{i,j}^m = K_{2i,2j}^{m+1} + K_{2i+1,2j}^{m+1} + K_{2i+1,2j+1}^{m+1} + K_{2i,2j+1}^{m+1}, \text{ для всех } m = \overline{0, n-1}.$$

Здесь и далее под операцией суммирования кортежей подразумевается их поэлементное суммирование.

Этап 2. Нахождение интегрального цвета. Выполняется в реальном времени.

Проекция центров семи смежных пикселей [7] (рис. 3) представляет собой шестиугольник  $P_1P_2P_3P_4P_5P_6$ . Согласно предложенному методу этот шестиугольник делится на шесть треугольников:  $P_1P_0P_2, P_2P_0P_3, P_3P_0P_4, P_4P_0P_5, P_5P_0P_6, P_6P_0P_1$ ;

Для каждого из треугольников вычисляется кортеж сумм коэффициентов  $(Ka_{sum}, Kb_{sum}, Kc_{sum})$  при помощи функции  $sumTri$  для дерева кортежей.

Взвешенный цвет каждой треугольной области будет определяться соотношением

$$C_k^{**} = a_k \cdot Ka_{sum} + b_k \cdot Kb_{sum} + c_k \cdot Kc_{sum}$$

Интегральный цвет проекции пикселя будет вычисляться согласно (13).

### Результаты моделирования

Расчет количества операций, необходимых для накопления коэффициентов  $Ka_{i,j}, Kb_{i,j}, Kc_{i,j}$ , приходящихся на один тексель текстуры (классификационный квадрат – КК) для трех составляющих цвета при выполнении анизотропной фильтрации методом, основанным на использовании текстур, заданных растровой матрицей цветов и предложенным методом быстрых вычислений с использованием подготовленного дерева текстур приведен в табл. 1:

Таблица 1

**Количество операций, приходящихся на один КК**

	Функция Vox	Функция Куранта
Метод, основанный на растровом задании текстур, в каждом текселе хранятся значения RGB	1 выборка из памяти 3 суммирования	1 выборка из памяти 6 умножений 11 суммирований
Предложенный метод быстрых вычислений	1 выборка из памяти 3 суммирования	1 выборка из памяти 9 суммирований

Из табл. 1 следует, что при расчете предложенным методом быстрых вычислений взвешенного цвета текселя для случая анизотропной фильтрации функцией Куранта исключаются операции умножения и сокращается количество операций суммирования.

Произведем подсчет количества обрабатываемых текстелей (КК) на примере двух форм треугольных областей, приведенных на рис. 6:

Как показано на рис. 7, выполнение анизотропной фильтрации предложенным методом позволяет с увеличением разрешения текстуры значительно уменьшить количество обрабатываемых КК. При этом существенно увеличивается производительность, так как уменьшается число выборок из памяти и

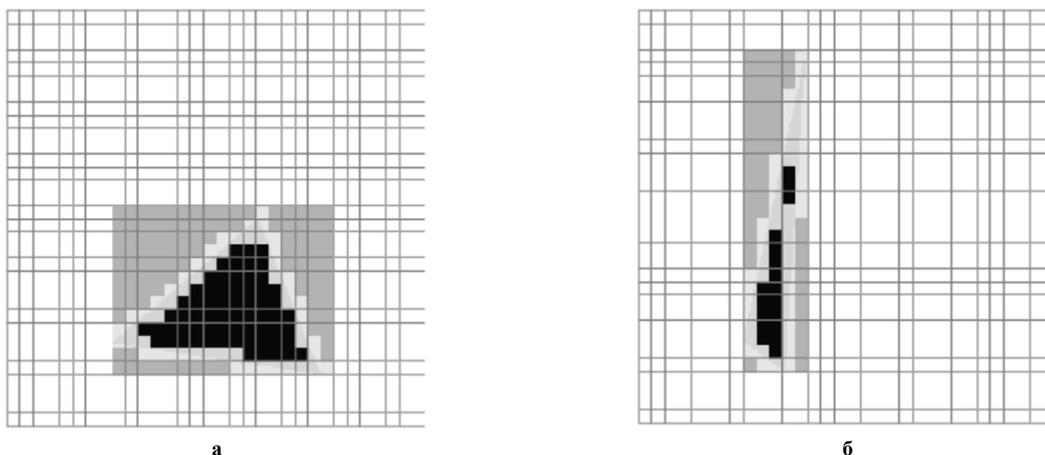


Рис. 6. Примеры расположения обрабатываемых в процессе выполнения фильтрации треугольных подобластей проекции пикселя на плоскость текстуры

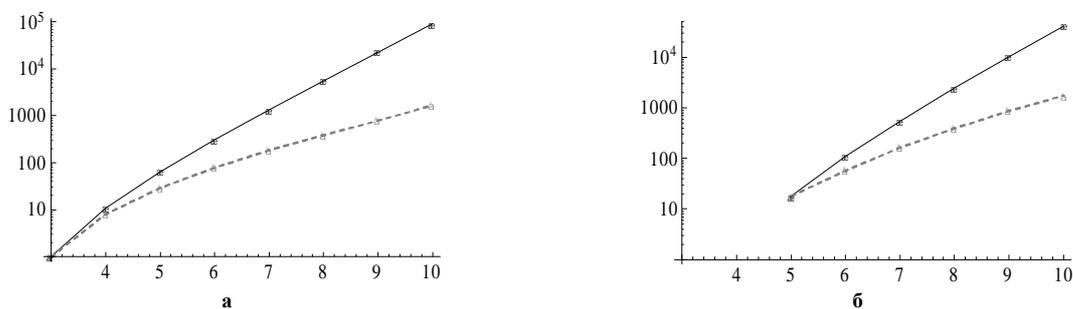


Рис. 7. Зависимость количества обрабатываемых КК текстуры от номера уровня (разрешения текстуры). Ось ординат (количество обрабатываемых КК) показана в логарифмической шкале. По оси абсцисс отложено количество уровней дерева текстуры. Сплошной линией обозначена зависимость для случая задания текстуры в виде матрицы цветов, а пунктирной – для случая задания текстуры в виде кватернарного дерева:  
 а – зависимость, полученная для треугольника, изображенного на рис. 6,а  
 б – зависимость, полученная для треугольника, изображенного на рис. 6,б

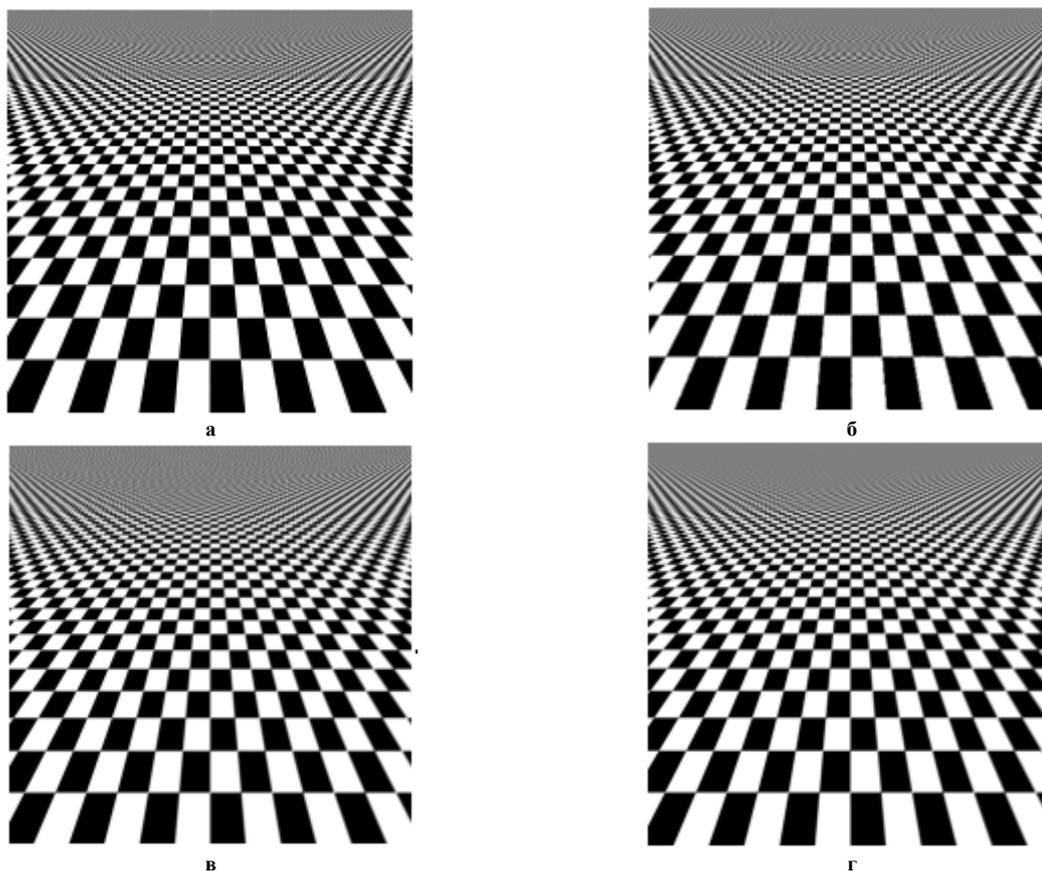


Рис. 8. Результаты моделирования

На рис. 8 а, в приведені результати фільтрації согласно (1) функціями Vox и Куранта текстури (2), которые можно принять в качестве эталонных. На рис. 8 б, г приведені результати фільтрації предложенным методом для фильтров Vox и Куранта этой же текстуры, представленной в виде кватернарного дерева согласно предложенному методу. Результаты моделирования, представленные на рис. 8, показали, что выполнение фильтрации предложенным методом быстрых вычислений позволяет устранять эффект алиасинга так же качественно, как в случае фильтрации согласно (1) в плоскости текстуры. Время, затраченное на синтез изображения предложенным методом, до 100 раз меньше, чем при синтезе эталонных изображений.

#### **Выводы**

Предложен метод быстрых вычислений для задач анизотропной фильтрации текстур при синтезе изображений обратным трассированием. Особенности метода:

- текстура представляется в виде КД. Такое представление текстуры позволяет существенно уменьшить количество обращений к памяти и число арифметических операций;
- в случае фильтрации функцией Куранта предлагается представить информацию о текселе текстуры в виде тройки заранее вычисленных значений, что исключает операции умножения и сокращает количество суммирований в процессе выполнения анизотропной фильтрации.

Предложенный метод позволяет повысить производительность выполнения анизотропной фильтрации при синтезе изображений обратным трассированием в реальном времени.

#### **Литература**

1. Гусятин В.М. Алгоритм геометрических преобразований изображения в системах визуализации тренажеров транспортных средств / Гусятин В.М // *Авиационно-космическая техника и технология*. Труды ХАИ им. Н.Е. Жуковского за 1997, с.467-471.
2. Foley J.D., van Dam A., Feiner S.K., Hughes J.F. *Computer Graphics (principles and practice)* by Addison-Wesley Publishing Company, Inc. 1996, 1175 p.
3. Никулин Е. А. *Компьютерная геометрия и алгоритмы машинной графики*. – СПб.: БВХ–Петербург, 2003. – 560 с.
4. Christopher C. Tanner, Christopher J. Migdal, and Michael T. Jones. The clipmap: A virtual mipmap. In *SIGGRAPH 98 Conference Proceedings*, pages 151.158, July 1998
5. Samet H. The quadtree and related hierarchical data structures // *ACM. Computing Surveys*. – 1984. – Vol. 16. – P. 187-260.
6. Crow, F.C., "Summed-Area Tables for Texture Mapping", *SIGGRAPH 84*, 207-212.
7. Гусятин В.М. Метод анизотропной фильтрации текстур при синтезе изображений обратным трассированием / В.М. Гусятин, Я.В. Чаговец, Д.Г. Кожушко // "Информатика, кибернетика и вычислительная техника" (ИКВТ-2009). Выпуск 10 (153) – Донецк: ДонНТУ– 2009. – С.64– 69.
8. Гусятин В.М. Упаковка векторных текстур в задачах синтеза изображений для систем визуализации/ В.М. Гусятин, Я.В. Чаговец, Д.Г. Кожушко // *Вісник НТУ "ХПІ" "Інформатика і моделювання"*. – 2005. – № 56. – С.9-16.

Надійшла до редакції  
7.6.2010 р.

**УДК 004.89+656.2**

**Т.О. САВЧУК, К.В. ЩЕПАНОВСЬКИЙ**

Вінницький Національний Технічний Університет

### **ПОШУК АСОЦІАТИВНИХ ПРАВИЛ ДЛЯ АНАЛІЗУ НАДЗВИЧАЙНИХ СИТУАЦІЙ НА ЗАЛІЗНИЧНОМУ ТРАНСПОРТІ**

Запропоновано алгоритм пошуку асоціативних правил для аналізу надзвичайних ситуацій на залізниці, що базують на розподілі множини факторів, які описують надзвичайні ситуації.

**A search algorithm of associative rules to analyze emergency situations on the railroad, based on the section set of factors that describe the emergency is offered.**

Ключові слова: асоціативні правила, надзвичайні ситуації, пошук закономірностей, алгоритм апіорі, коефіцієнт підтримки, коефіцієнт довіри.

**Актуальність задачі пошуку асоціативних правил для аналізу надзвичайних ситуацій на залізниці**

Зростаюче антропогенне навантаження на навколишнє середовище обумовлює різке загострення екологічної ситуації в глобальному масштабі. Однією із основних причин цього є техногенні надзвичайні ситуації на залізничному транспорті, які відбуваються при вантажоперевезеннях небезпечних або отруйних