

Література

1. Автоматизация тестирования: минусы [Электронный ресурс] // Режим доступа до файлу: <http://habrahabr.ru/post/111292/>
2. Автоматизация тестирования A1Q1 [Электронный ресурс] // Режим доступа до файлу: http://www.a1qa.by/service/automatic_tests_development/automation_process/
3. Мартин Фаулер. Архитектура корпоративных программных приложений / Фаулер Мартин. – М. : Вильямс. – 2007. – 544 с.
4. Len Bass, Paul C. Clements and Rick Kazman Software Architecture in Practice, Third Edition/ Addison-Wesley Professional – 2012. – 640p.
5. Len Bass, Rick Kazman, Categorizing Business Goals for Software Architectures, Technical Report CMU/SEI-2005-TR-021, 2005.
6. Doran, G. T., There's a S.M.A.R.T. way to write management's goals and objectives. Management Review, Volume 70, Issue 11 (AMA FORUM), pp. 35– 36, 1981.

References

1. Avtomatyzatsyya testyrovannya: mynusy <http://habrahabr.ru/post/111292/> (Jul 04 2013).
2. Avtomatyzatsyya testyrovannya A1Q1 http://www.a1qa.by/service/automatic_tests_development/automation_process/ (Jul 04 2013)
3. Martyn Fauler, Arkhytektura korporatyvnykh prohrammnykh prylozheny/ Vyl'yams – 2007. – 544 p.
4. Len Bass, Paul C. Clements and Rick Kazman Software Architecture in Practice, Third Edition/ Addison-Wesley Professional – 2012. – 640p.
5. Len Bass, Rick Kazman, Categorizing Business Goals for Software Architectures, Technical Report CMU/SEI-2005-TR-021, 2005.
6. Doran, G. T., There's a S.M.A.R.T. way to write management's goals and objectives. Management Review, Volume 70, Issue 11 (AMA FORUM), pp. 35– 36, 1981.

Надійшла до редакції
4.7.2013 р.

УДК 004.5

О. Н. РОМАНИЮК, С. О. РОМАНИЮК, М. П. ПІДДУБЕЦЬКА

Вінницький національний технічний університет, Україна

АНАЛІЗ МЕТОДІВ АНІЗОТРОПНОЇ ФІЛЬТРАЦІЇ ТЕКСТУР

У статті наведено аналіз методів анізотропної фільтрації текстур. Розглянуто особливості базових методів. Наведено формули для розрахунків сліду спроектованого на текстурну площину піксела і процедури визначення кольору відповідних текселів.

Ключові слова: текстурування, анізотропна фільтрація, тексел.

O. N. ROMANIUK, S. O. ROMANIUK, M. P. PIDUBETSKA

Vinnitsia National Technical University, Ukraine

ANALYSIS OF METHODS FOR ANISOTROPIC TEXTURE FILTERING

This paper presents an analysis of methods for anisotropic texture filtering. The futures of the basic techniques are described. The formulas for calculating the track projected on the texture plane from pixel and procedures for determining relevant colour for each texel.

Key words: texturing, anisotropic filtering, texel.

Вступ

Стрімкий розвиток комп'ютерної графіки вимагає відповідного розвитку методів і засобів візуалізації зображень тривимірних об'єктів для забезпечення зростаючих вимог до реалістичності. Для побудови високореалістичних зображень широко використовують текстур, які накладаються на графічні об'єкти. Використання текстур у багатьох випадках дозволяє успішно вирішувати задачі, які надзвичайно трудомістко розв'язати прямими методами. Текстурування дозволяє суттєво зменшити обчислювальні витрати та зробити можливим інтерактивний режим візуалізації.

Головна вимога текстурування – реалістичне відтворення поверхні, яке в першу чергу залежить від правильного відображення об'єктів у двовимірну площину і урахування їх перспективи.

У зв'язку з цим актуальними є питання підвищення реалістичності відтворення графічних об'єктів з використанням текстурування шляхом визначення слідів пікселів на текстурну площину.

Аналіз основних методів

Розглянемо базові методи фільтрації текстур, які не потребують значних апаратних затрат та використовують прості процедури для процесу накладання текстур.

Білінійна фільтрація [1, 2] – метод, який полягає у використанні інтерполяційної техніки. Для визначення текселів, які повинні бути задіяні для апроксимації, використовується основна форма світлової плями – коло. По суті, коло апроксимується чотирма текселями. Результуючий колір пікселя визначається в результаті операції змішування: спочатку змішуються значення кольору двох пар текселів по осі X, а потім змішуються отримані значення по осі Y. Головний недолік білінійної фільтрації полягає в тому, що апроксимація виконується коректно тільки для полігонів, які розташовані паралельно до екрана або до точки спостереження. Якщо полігон розвернуто під кутом, то має місце некоректне накладання текстури.

Трилінійна фільтрація, як і білінійна, використовує блоки з 4 текселів. Колір пікселя, який повинен бути виведений на екран, визначається в результаті інтерполяції за кольорами двох мір-текстур. Рівні мір-текстури являють собою заздалегідь розраховані зменшені версії вихідної текстури. У даний час підтримка трилінійної фільтрації стала стандартною функцією графічних чипів. На жаль, при трилінійній фільтрації також мають місце артефакти при відтворенні об'єктів.

Текстурування з використанням методів білінійної і трилінійної фільтрації є досить ефективними, однак вони не враховують реальну проекцію пікселя з площини екрану на текстурну площину, що не забезпечує адекватне відтворення кольорів пікселів.

Найбільша реалістичність досягається при анізотропній фільтрації текстур, яка порівняно з методами білінійної і трилінійної фільтрації враховує слід пікселя на текстурну площину.

Мета статті - аналіз анізотропної фільтрації текстур.

Аналіз методів анізотропної фільтрації текстур

У комп'ютерній графіці піксель має визначену форму. При білінійній і трилінійній фільтрації текстур не враховуються фізичні властивості пікселя як точки, хоча форма сліду від спроектованого пікселя має велике значення при визначенні кольору відповідного текселя.

На сьогоднішній день найкращі результати за якістю формування тривимірних сцен дає анізотропна фільтрація. Текселі при даній фільтрації відбираються з ділянки, визначеної з урахуванням перспективи (анізотропна форма). При анізотропній фільтрації береться до уваги кут, під яким піксель проектує свій слід на площину текстури (рис. 1), а також сама форма даного сліду [3].

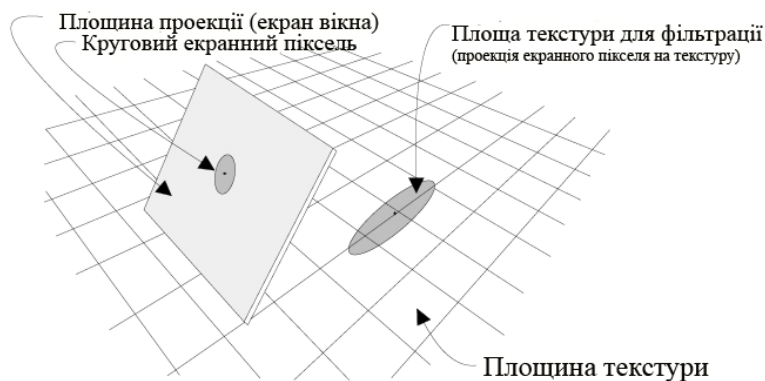


Рис. 1. Еліпсоподібний слід проекції пікселя

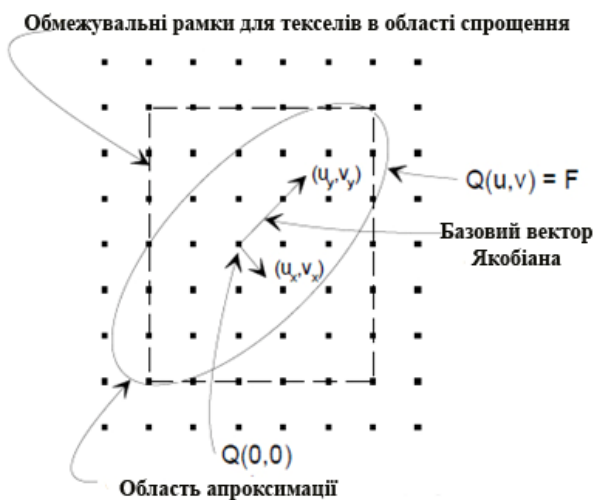


Рис. 2. Апроксимація еліпсоподібної форми сліду

При круговій моделі пікселя його проекція являє собою еліпс довільної форми, як це показано на рис. 1. У деяких випадках для зменшення обчислювальної складності еліпс замінюють на описаний навколо нього паралелограм.

Для апроксимації еліпсоподібної форми сліду використовуються спеціальні фільтри (EWA filtering methods). При цьому виконують такі дії: обчислюють площу еліпса, визначають кількість текселів, які покриваються слідом від пікселя, зважують текселі на даній ділянці та проводять кінцеве визначення кольору пікселя [4].

Схематичне зображення еліпсоподібної форми сліду на площині текстури, а також функції, які застосовуються для опису еліпса зображено на рис. 2.

При фільтрації виконують кілька кроків.

Перший крок – оцінка центру сліду еліпсоподібної форми, який знаходиться у точці (U_0, V_0) і відповідає центру пікселя, який проектується на поверхню. Далі обчислюється інверсна матриця Якобіана J_{U_0, V_0}^{-1} [3] для даної точки, що дає можливість отримати базові вектори Якобіана, які показані на рис. 2.

Апроксимація площини текстури τ еліпсоподібної форми ґрунтується на властивостях матриці Якобіана (якобіан). Оскільки вектори обчисленої матриці Якобіана формують 2D-лінійний просторовий вектор відносно екранних координат, якобіан використовується як засіб для апроксимації площини τ . Необхідно врахувати, що якобіан обчислюється для кожного нового положення точки в площині текстурного фільтра. Якщо координати точки у площині фільтра – (x_s, y_s) , то матриця Якобіана обчислюється для точки з координатами (U_0, V_0) на площині текстури.

Для загального випадку накладання текстур, матриця Якобіана обчислюється за таким виразом:

$$J = \frac{\partial \tau}{\partial (U, V)} = \begin{bmatrix} \frac{\partial}{\partial U} \left(\frac{xw}{w} \right) & \frac{\partial}{\partial V} \left(\frac{xw}{w} \right) \\ \frac{\partial}{\partial U} \left(\frac{yw}{w} \right) & \frac{\partial}{\partial V} \left(\frac{yw}{w} \right) \end{bmatrix} = \frac{1}{w^2} \begin{bmatrix} wx_U - xw_U & wx_V - xw_V \\ wy_U - yw_U & wy_V - yw_V \end{bmatrix}, \quad (1)$$

де часткові похідні xw , yw і w - елементи текстурної матриці, τ , xw , yw і w - відповідні екранні значення, обчислені за допомогою матриці відповідно до значень на площині текстури з параметрами (U, V) . Для випадку текстурування, коли площина текстури знаходиться паралельно до площини екрану або до точки спостерігача, значення $w_U = w_V = 0$ і $w = 1$. Тоді матриця Якобіана може бути спрощена до такого виразу:

$$J = \frac{\partial \tau}{\partial (U, V)} = \begin{bmatrix} \partial x / \partial U & \partial x / \partial V \\ \partial y / \partial U & \partial y / \partial V \end{bmatrix} = \begin{bmatrix} x_U & x_V \\ y_U & y_V \end{bmatrix} \quad (2)$$

Для параметризації еліптичної ділянки за допомогою аналітичних функцій найбільше підходить рівняння для конуса з вершиною у початковій точці координат:

$$(U, V) \begin{bmatrix} A & B/2 \\ B/2 & C \end{bmatrix} \begin{pmatrix} U \\ V \end{pmatrix} = F \quad (3)$$

Параметри A, B, C, F з рівняння 3 оцінюються чотирма рівняннями зворотної матриці Якобіана [3]. Результат розширює єдину маску фільтра, визначену у рівнянні 1 (рис. 2). За допомогою параметрів A, B, C, F обчислюються обмежувальні рамки на текстурній площині, в які входять усі текселі з еліпсоподібної маски. Це виконується для того, щоб швидко визначити зону апроксимації, на якій відбувається фільтрація.

Для кожного текселя, який входить до обмежувальної ділянки, обчислюється значення функції $Q(U, V)$ і порівнюється з порогом функції F . Якщо Q менше за значення F , то тексель знаходиться у зоні апроксимації і зважується у єдиному фільтрі p' . В іншому випадку тексель пропускається і над ним не виконуються дії. Суму всіх зважених текселів також зберігають для подальшої нормалізації фінального значення фільтрації, після перевірки всіх текселів. Результат нормалізації направляється для виведення остаточного пікселя на екран.

Еліпсоподібний слід пікселя в площині текстури, коли центр пікселя проектується в точку $(0;0)$, може бути визначено такими рівняннями:

$$d^2(u, v) = Au^2 + Buv + Cv^2, \quad (4)$$

де

$$A_{nn} = (\partial v / \partial x)^2 + (\partial v / \partial y)^2 \quad (5)$$

$$B_{nn} = -2 * (\partial u / \partial x * \partial v / \partial x + \partial u / \partial y * \partial v / \partial y) \quad (6)$$

$$C_{nn} = (\partial u / \partial x)^2 + (\partial u / \partial y)^2 \quad (7)$$

$$F = A_{nn} C_{nn} - B_{nn}^2 / A \quad (8)$$

$$A = A_{nn} / F \quad (9)$$

$$B = B_{nn} / F \quad (10)$$

$$C = C_{nn} / F \quad (11).$$

Часткові похідні $(\partial u / \partial x, \partial u / \partial y, \partial v / \partial x, \partial v / \partial y)$ визначають зміну координат текстури в результаті зміни точки зору на екранній площині. Величина d^2 показує відстань текселя з координатами $(u;v)$ від центру пікселя при зворотній проекції на екранний простір. Згідно з алгоритмом виконується сканування прямокутної ділянки еліптичного сліду $d^2 \leq 1$ у просторі текстури та обраховується вага текселів усередині цієї ділянки пропорційно до відстані d .

Існує багато альтернативних способів апроксимації еліпсоподібного сліду.

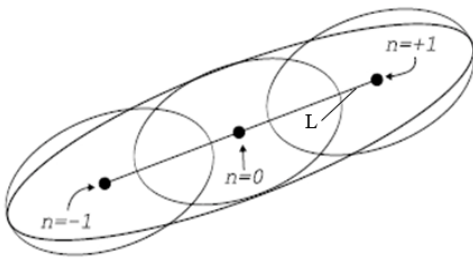


Рис. 3. Апроксимація еліпса за допомогою еліпсів меншого діаметра

У [5] наводиться метод, який використовує прості геометричні форми для апроксимації сліду максимально наближеного до форми еліпса. Пропонується застосування спеціальних анізотропних семплів. Головна ідея методу полягає в тому, що апроксимація еліпса витягнутої форми виконується за допомогою еліпсів з меншої довжиною головної осі, які містяться у вихідному еліпсі. Таким чином можна обійти деякі обмеження апаратної частини і якісніше відфільтрувати еліпсоподібний слід з високим ступенем анізотропії.

Анізотропні проби (семпли) еліпсів із меншою довжиною головної осі розташовуються вздовж лінії L, як показано на рис. 3.

Довжина лінії L і число семплів $N_{проб}$ обраховуються за такими рівняннями:

$$N_{проб} = 2 * (R_{головна} / (\alpha R_{побічна})) - 1 \quad (12)$$

$$L = 2 * (R_{головна} - \alpha R_{побічна}), \quad (13)$$

де α - степінь анізотропії базового сліду.

Для спрощення обрахунків, визначається непарна кількість семплів. Семпли розміщуються відносно центра (u_m, v_m) фільтра та обраховуються за формулами:

$$\theta = a \tan(B / (A - C)) / 2 \quad (14)$$

$$du = \cos(\theta) * L / (N_{проб} - 1) \quad (15)$$

$$dv = \sin(\theta) * L / (N_{проб} - 1) \quad (16)$$

$$(u_n, v_n) = (u_m, v_m) + n / 2 * (du, dv), \quad n = 0, \pm 2, \pm 4... \quad (17)$$

де (u_n, v_n) - позиція n-ого семпла.

Для кращого відтворення форми еліпса, семпли зважуються і апроксимуються пропорційно до центру еліпса. У рівнянні 17 анізотропні семпли n розподілені в двох кадрах, які йдуть підряд. Перший кадр використовує семпли 0; 4; 8... , а наступний кадр 0; 2; 6... Нульовий семпл включається у обидва кадри для мінімізації дисперсії між двома кадрами. Коли фрейми відображаються в швидкій послідовності, людське око сприймає середнє з двох кадрів. Для подальшого покращення змішування між кадрами, набір семплів, які використовуються для фільтрації, повинні варіювати залежно від позиції пікселя.

Метод Текграм (фільтрація за допомогою трилінійних семплів) [6] забезпечує кращу візуальну якість, ніж трилінійна фільтрація, але меншу ніж метод, який було описано вище. Такий вид фільтрації застосовує серію трилінійних семплів уздовж головної осі еліпсоподібного сліду.

Фільтрація за допомогою трилінійних семплів використовує чотири часткові похідні для обрахунку двох векторів у просторі текстури: $(\partial u / \partial x, \partial v / \partial x)$ і $(\partial u / \partial y, \partial v / \partial y)$. Апроксимація відбувається усередині площі паралелограма, який утворюється цими двома векторами, шляхом фільтрації трилінійних семплів вздовж лінії, яка має довжину і нахил більшої сторони паралелограма. Ця лінія може відхилитись від головної осі еліпса до 45 градусів і може бути коротшою за головну ось еліпса майже в чотири рази.

Конкретний випадок фільтрації показано на рис. 4, де $(\partial u / \partial x, \partial v / \partial x)$ має значення (13, 0) і $(\partial u / \partial y, \partial v / \partial y)$ має значення (12, 5).

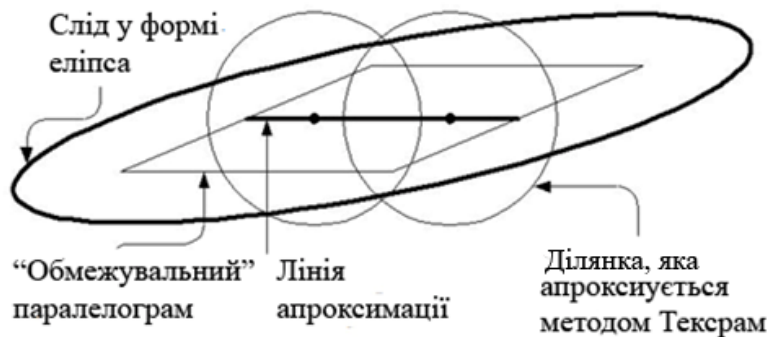


Рис. 4. Метод анізотропної фільтрації за допомогою трилінійних семплів

Анізотропна фільтрація з точністю до суб-текселя (SPAF – Sub-texel Precision Anisotropic Filtering) визначає вагу конкретного текселя залежно від того, на скільки близько його центр знаходиться від спроектованого сліду [7]. Якщо центр текселя покривається проекцією сліду – його вага дорівнює 1, в протилежному випадку вага дорівнює 0. Для підвищення якості зображення в даному методі тексель поділяється на суб-текселі і обчислення виконуються для кожного з цих суб-текселів.

Метод SPAF фільтрує текселі в ділянці, що охоплює чотирикутний слід, який формується залежно від ваги текселів. Вага текселя відіграє ключову роль в отриманні високої якості зображення на площині з обмеженим числом текселів. Алгоритм генерує зону сліду з суб-тексельною точністю і формує таблицю ваг текселів невеликих розмірів без втрати якості. Вага визначається за допомогою Гаусівського алгоритму і обраховується для кожного текселя, в той час як більшість методів фільтрації сприймають вагу однаковою для кожного текселя.

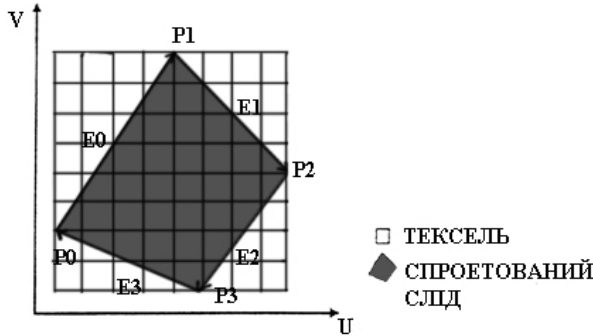


Рис. 5. Проекція сліду пікселя на текстурну площину

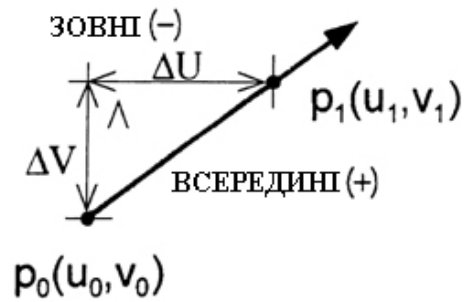


Рис. 6. Виразення напрямлюючої P₀P₁ через систему координат VOU

На рис. 5 зображено схему сліду, спроекційованого від пікселя на площину текстури.

Спроектований слід (рис. 5), має чотирикутну форму і кожна сторона задається чотирма рівняннями (E0, E1, E2, E3), P0, P1, P2, P3 – вершини сліду. На рис. 5 прямокутна ділянка, яка відображає спроектований слід, включає в себе 64 (8×8) текселів. Направляюча лінія від точки P0 до точки P1 виражається через рівняння 18-22.

$$E(u, v) = (u - u_0) * deu + (v - v_0) * dev \quad (18)$$

Коефіцієнти, наведені в формулі 18, визначаються за формулами:

$$deu * \Delta U + dev * \Delta V = 0, \quad (19)$$

$$deu = (\Delta V) / (|\Delta U| + |\Delta V|), \quad (20)$$

$$dev = (-\Delta U) / (|\Delta U| + |\Delta V|), \quad (21)$$

$$\Delta U = u_1 - u_0 \quad \text{і} \quad \Delta V = v_1 - v_0 \quad (22)$$

Значення отримується в результаті введення координат довільної точки у лінійне рівняння, що обраховує відстань від точки до лінії. Іншими словами, точка знаходиться всередині, якщо знак додатний і зовні, якщо знак від'ємний. Чотирикутник задано чотирма лінійними рівняннями, а отже коли вводяться координати точки і значення усіх лінійних рівнянь для цієї точки додатна, то точка знаходиться всередині чотирикутника.

На рис. 7 зображено взаємозв'язок форми і розміру покритого спроектованим слідом текселя. Нахил лінії і значення лінійного рівняння отримано в результаті введення у рівняння координат точки центру текселя.



Рис. 7. Взаємозв'язок форми і покритої площі текселя

Як показано на рис. 7, якщо значення лінійного рівняння менше ніж -0.5 або більше ніж +0.5, площа текселя або зовсім не покрита, або повністю покрита. Іншими словами, значення маски текселів задається комбінацією значень суб-текселів, які можуть мати значення або 1 (тобто покриті), або 0 (не покриті). Під час фільтрації, нахил і значення лінійного рівняння отримуються з адресної таблиці. Для вирахування ваги покритої площі, по-перше, отримується суб-тексельна маска шляхом вирахування нахилу і значення

чотирьох лінійних рівнянь, а потім біт за бітом виконується логічне перемноження і, нарешті, визначається загальне число суб-текселів, значення яких дорівнює 1.

Даний метод концептуально схожий з супер-семплінгом, який використовується для згладжування текстур, і має недолік – низьку точність через наближені обрахунки. Недоліком даного методу також є велика обчислювана складність, адже обрахунки здійснюються для кожного текселя і суб-текселя, що покриваються спроектованим слідом.

Головний недолік методу, який проявляється в ефектах алайзингу, це те, що лінія апроксимації, яка проходить через центр еліпса набагато коротша за головну ось еліпса і трилінійні семпли можуть розташовуватись занадто далеко один від одного.

Висновки

У статті наведено аналіз методів анізотропної фільтрації, яка широко використовується при формуванні високореалістичних зображень. Показано, що для високоякісного текстуровання необхідно враховувати позицію точки спостереження, що обумовлює розрахунок площини сліду пікселя на текстурній площині. Розглянуто різні підходи для визначення текселів за умови, що слід пікселя має форму еліпса. Наведено формули для розрахунків сліду пікселя на текстурну площину і процедури визначення кольору текселів.

Література

1. BJORKE K. High-quality filtering. In GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics. / K. BJORKE. Addison Wesley. – 2004.
2. 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.
3. Landsdale Robert C. Texture Mapping and Resampling for Computer Graphics. / Robert C. Landsdale. Department of Electrical Engineering, University of Toronto, Toronto, Canada. – 1991.
4. Green Ned, Heckbert Paul. Creating Raster Omnimax Images from Multiple Perspective Views Using the Elliptical Weighted Average Filter. / Ned Green, Paul Heckbert. IEEE Computer Graphics and Applications, 6 (6). – 1986.
5. Mavridis Pavlos, Papaioannou Georgios. High Quality Elliptical Texture Filtering on GPU. / Pavlos Mavridis, Georgios Papaioannou. Departments of Informatics, Athens University of Economics & Business. – 2004.
6. McCormack Joel, Perry Ronald, Farkas Keith, Jouppe Norman P. Feline: Fast Elliptical Lines for Anisotropic Texture Mapping. / Joel McCormack, Ronald Perry, Keith I. Farkas, Norman P. Jouppe. Compaq Computer Corporation's Western Research Laboratory and Mitsubishi Electric Research Laboratory. – 1999.
7. Shin Hyun-Chul, Lee Jin-Aeon, Kim Lee-Sup. SPAF: sub-texel precision anisotropic filtering. / Hyun-Chul Shin, Jin-Aeon Lee, Lee-Sup Kim. HWWS '01: Processing of the ACM Siggraph/Eurographics workshop on Graphic hardware. – 2001.

Надійшла до редакції
8.6.2013 р.

УДК 004.925

С.О. РОМАНЮК, О.В. РОМАНЮК, Д.Л. БЛАГОДИР

Вінницький національний технічний університет

ВИЗНАЧЕННЯ СКАЛЯРНОГО ДОБУТКУ ДВОХ ВЕКТОРІВ, ЗАДАНИХ У ПОЛЯРНИХ КООРДИНАТАХ, ДЛЯ ЗАДАЧ КОМП'ЮТЕРНОЇ ГРАФІКИ

Запропоновано нову формулу для визначення скалярного добутку двох векторів, заданих у полярних координатах, яка дозволяє зменшити обчислювальні витрати та прискорити процес формування реалістичних зображень. Запропоновано структурну схему блока визначення скалярного добутку двох векторів, у якому реалізовано нову формулу.

Ключові слова: вектор, скалярний добуток, полярна система координат.

S.O. ROMANIUK, O.V. ROMANIUK, D.L. BLAGODYR

Vinnitsia National Technical University

PROCEDURE OF DETERMINING SCALAR PRODUCT OF TWO VECTORS GIVEN IN POLAR COORDINATES FOR THE TASKS OF COMPUTER GRAPHICS

There had been proposed new formula for determining the scalar product of two vectors given in polar coordinates, which reduces the computational cost and accelerate the formation of realistic images. There had been proposed a block diagram