

## ВДОСКОНАЛЕНИЙ МЕТОД РОЗРОБКИ ЕФЕКТУ ВІЗУАЛІЗАЦІЇ 3D-ОБ’ЄКТІВ

© Лисак Т.А., Сернівка Ю.І., 2009

Запропоновано вдосконалений метод візуалізації 3D-об’єктів, що мають властивість відбиття на своїй поверхні навколишнього середовища. Розглянуто процес візуалізації відбиття світла від поверхні об’єкта за законом Ламберта та процес візуалізації відбиття навколишнього середовища від поверхні об’єкта із використанням кубічних карт. Метод забезпечує покращання якості відображення 3D-об’єктів за рахунок комплексного врахування цих двох процесів. Відкориговано вектор відбиття при його обчисленні у процесі візуалізації відбиття навколишнього середовища.

In this article perfected technique of 3D objects visualization which reflects environment on its surfaces is offered. The visualization process of light reflection from its surfaces by Lambert law is considered as well as the visualization process of reflection of environment from its surface which uses Cube maps technique. This method guarantees the 3D objects image quality because there is complicity usage of both processes. Reflection vector corrects when it calculates in process of reflection of environment.

**Вступ.** На основі засобів комп’ютерної графіки та інших прогресивних технологій з’явилися зовсім нові, не схожі ні на що раніш відоме, напрямки виробництва, професії, навіть середовища та взаємостосунки. Наслідками всебічного вторгнення комп’ютеризації у майже всі сфери діяльності людини є її поява нових видів мистецтва. До таких можна віднести комп’ютерну музику та анімацію. Остання з’явилася саме завдяки розвитку одного з видів комп’ютерної графіки, а саме – тривимірної (3D) графіки та анімації.

Реалізація запропонованого методу розроблення ефекту стала актуальною після виходу на ринок потужних графічних адаптерів з апаратною підтримкою багатьох елементів комп’ютерної графіки.

**Огляд літературних джерел.** Процес візуалізації відбиття світла за законом Ламберта [1, 2] був запропонований у статті в’єтнамського інженера Ву Тонг Фонга [3] у 1975р. Цей процес за назвою “Модель відбиття Бліна–Фонга” [4, 5] є доволі популярним і поширеним у комп’ютерній графіці, попри те, що існують точніші фізичні моделі.

Для зображення поверхонь, що відбивають навколишнє середовище, використовують технологію Кубічних карт [6]. Цю технологію запропонував Джим Блін у 1976 р[7]. Обчислення вектора відбиття, який використовується як координати текстурних карт, запропоновано у джерелі [8]. Цей підхід не є досконалим і потребує деякої корекції, що було реалізовано у цій статті.

**Постановка задачі.** Задача цієї статті полягає в розробленні ефекту візуалізації 3D-об’єкта. Вхідні параметри ефекту повинні впливати на кінцевий вигляд об’єкта. Найважливішими є такі вхідні параметри: позиція об’єкта, колір об’єкта, величина самосвічення об’єкта, величина відбиття навколишнього середовища на поверхні об’єкта, позиція джерела світла, колір світла.

### 1. Процес візуалізації відбиття світла від поверхні об’єкта за законом Ламберта

Як уже згадувалось, в’єтнамський інженер Ву Тонг Фонг запропонував модель освітлення, яка швидко набула популярності, і хоча апаратне забезпечення сильно удосконалилось з того часу, ця модель використовується і досі практично у всіх сучасних системах 3D-візуалізації реального часу.



Забарвлення дифузного освітлення обчислюється за формулою:

$$C_{diffuse} = C_d \cdot I_d (\bar{N} \cdot \bar{L}), \quad (2)$$

де  $C_d$  – власне забарвлення об'єкта цієї моделі;  $I_d$  – інтенсивність освітлення;  $\bar{N}$  – нормаль поверхні,  $\bar{L}$  – напрямлений вектор від позиції джерела світла до поверхні;  $\bar{N} \cdot \bar{L}$  – скалярний добуток векторів, що визначає кількість відбитого світла на конкретному фрагменті поверхні.

1.3. Модель освітлення відблиском. Важко уявити таку модель освітлення, не побачивши її. Насправді вона зустрічається доволі часто. Прикладом може бути чисто відполірована пряма металева поверхня, на яку напрямлене джерело світла, і на яку дивляться під кутом, не перпендикулярним до її поверхні. У результаті можна побачити відблиски на поверхні, які істотно збільшують реалістичність зображення. Ці відблиски є результатом відбиття джерела світла від поверхні.

У такій моделі освітлення, крім векторів позиції джерела світла і нормалі (як у випадку з дифузною моделлю), використовуються ще два вектори: оглядовий вектор і вектор відображення.

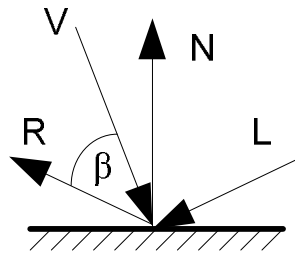


Рис. 3. Поверхня, нормаль ( $N$ ), оглядовий вектор ( $V$ ), відбитий вектор ( $R$ ) та вектор джерела світла ( $L$ )

Формула визначення забарвлення відблиском має вигляд:

$$C_{specular} = C_s \cdot I_s (\bar{L} \cdot \bar{R})^n, \quad (3)$$

де  $C_s$  – власне забарвлення об'єкта моделі;  $I_s$  – інтенсивність освітлення моделі;  $\bar{V}$  – оглядовий вектор;  $\bar{L}$  – вектор джерела світла;  $\bar{R}$  – відбитий від поверхні вектор, що обчислюється за формулою:

$$\bar{R} = (-\hat{V}) - 2 \cdot \bar{N} \cdot ((-\hat{V}) \cdot \bar{N}), \quad (4)$$

$$\hat{v} = \frac{\bar{v}}{\|\bar{v}\|}. \quad (5)$$

1.4. Модель самосвічення є найпростішою із наведених моделей. Вона визначається більшим значенням кольору матеріалу та сумою кольорів усіх моделей. Цю модель застосовують для об'єктів, що світяться, – таких, як джерела світла.

Визначення забарвлення самосвіченням:

$$C = \max(C_{emissive}, C_e). \quad (6)$$

1.5. Комбінування моделей освітлення. Після обчислення кожної з моделей освітлення об'єкт отримує забарвлення за формулою:

$$C = \max(C_{emissive}, (C_{ambient} + C_{diffuse} + C_{specular})). \quad (7)$$

## 2. Процес візуалізації відбиття навколишнього середовища від поверхні об'єкта із використанням кубічних карт

2.1. Попередній обрахунок кубічних карт. Для зображення на поверхні об'єкта інших об'єктів, що оточують його, тобто відображення навколишнього середовища, використовується

технологія Кубічних карт. Кубічні карти складаються з шести різних текстур (рис. 4). Кожна із цих текстур являє собою промальовану сцену з місця знаходження об'єкта, для якого ми генеруємо цю карту, у певному напрямку.



Рис. 4. Кубічна карта деякого навколишнього середовища

2.2. Відбитий вектор обчислюється так:

$$\bar{R} = \hat{P} - 2 \cdot \bar{N} \cdot (\hat{P} \cdot \bar{N}), \quad (8)$$

де  $\bar{P}$  – позиція об'єкта що відображає навколишнє середовище, нормалізація цього вектора ( $\hat{P}$ ) здійснюється згідно з формулою 5;  $\bar{N}$  – нормаль поверхні об'єкта;  $\bar{T}$  – відбитий від поверхні об'єкта вектор, який одночасно і є координатами кубічної текстури для конкретного фрагмента поверхні об'єкта.

2.3. Текстурні координати обчислюються для кубічних карт множенням оберненої матриці спостереження, що є одним із параметрів ефекту, на відбитий вектор:

$$\bar{T} = \bar{R} \times V^{-1}, \quad (9)$$

де  $V^{-1}$  – обернена матриця спостереження.

2.4. Корекція відбитого вектора

Після обчислення відбитого вектора можна отримати забарвлення в кожній точці поверхні об'єкта, що відбиває навколишнє середовище. Цей ефект можна застосовувати для об'єктів будь-яких геометричних форм. Проте такий алгоритм отримання тривимірних координат буде ідеальним тільки для об'єктів, форма яких близька до сфери. У решті випадків, особливо коли об'єкт має дещо плоску форму, за цим алгоритмом обчислюють далеке від ідеального відображення навколишнього середовища. Проблема полягає в тому, що вираховується вектор, що прямує з позиції огляду сцени (А) в конкретну точку поверхні об'єкта (Б) та підставляється у кубічну текстуру. Текстуру визначали в точці центра об'єкта (Б'), а не у поточній позиції поверхні (Б), де обчислюється цей вектор. Отриманий вектор буде дещо неправильно визначати координати (В'), за якими вибирають забарвлення.

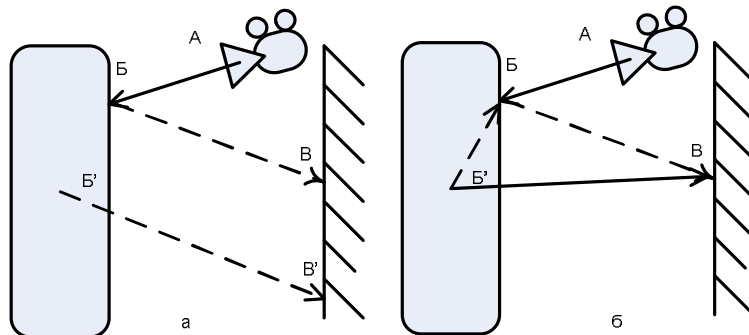


Рис. 5. Зображення відбитого вектора (а) та його корекція (б)

Проте, є просте рішення цієї проблеми за допомогою кількох операцій із векторами. Корекцію відбитого вектора можна здійснити за формулою:

$$\overline{BB'} = (A - B') + \overline{AB} \quad (10)$$

Нижче показані зображення об'єкта з некоригованим та з коригованим векторами відображення:



а



б

Рис. 6. Відображення навколишнього середовища при некоригованому (а) та коригованому векторах (б)

### 3. Остаточний вигляд поверхні об'єкта

У визначенні остаточного вигляду поверхні використовують як обчислення поверхні непрозорого об'єкта, так і обчислення поверхні об'єкта, що відбиває навколишнє середовище. А їх вплив на забарвлення поверхні встановлюється властивістю типу поверхні, що має назву *інтенсивність відбиття*. Чим більше значення цієї опції, тим більша поверхня об'єкта матиме відбитий, а не власний колір і навпаки.



А

Б

В

Г

Д

Рис. 7. Об'єкти з різною величиною відбиття. Кожен наступний об'єкт має більшу величину відбиття відносно сусіднього об'єкта зліва

**Висновки.** Запропонований вдосконалений метод розроблення ефекту візуалізації 3D-об'єктів, який поєднує відомі ефекти, розширяє можливості відображення об'єктів різних типів з погляду природи матерілу. Забезпечує покращання якості відображення на основі корекції вектора відбиття навколишнього середовища. Ефективність методу доведена практичною реалізацією і підтверджена покращанням графіки в уже існуючій програмі.

1. J. D. J. Ingle and S. R. Crouch, *Spectrochemical Analysis*, Prentice Hall, New Jersey (1988).
2. Houghton, J.T. *The Physics of Atmospheres 2nd ed. Chapter 2*.
3. G. Bisob, D. Weirner, *Fast Phong Shading*, SIGGRAPH 86, *Computer Graphics*, v20, n4, сt527-536, 1986.
4. Закон Ламберта. Модель

отражения Фонга. Модель отражения Блинна–Фонга ( [www.comppgraphics.info](http://www.comppgraphics.info) ). 5. Программирование шейдеров на HLSL. Модели освещения. ( [www.gamedev.ru](http://www.gamedev.ru) ) 6. Кубическая текстура. ( [www.ru.wikipedia.org](http://www.ru.wikipedia.org) ). 7. Blinn K. Models of Light Reflection for Computer Sythesized Pictures, SIGGRAPH 77, Computer Graphics, v11, n4, cm273-536, 1977. 8. HDRCubeMap Direct3D Sample ( [www.msdn.microsoft.com](http://www.msdn.microsoft.com) ).

УДК 004.272.2, 004.272.23

А.О. Мельник, А.М. Сало, В.А. Клименко, Л.О. Цигилик, А.В. Юрчук  
Національний університет “Львівська політехніка”,  
кафедра електронних обчислювальних машин

## РЕАЛІЗАЦІЯ ПРОГРАМНИХ СПЕЦІАЛІЗОВАНИХ ПРОЦЕСОРІВ У РЕКОНФІГУРОВНИХ ПРИСКОРЮВАЧАХ УНІВЕРСАЛЬНИХ КОМП'ЮТЕРІВ

© Мельник А.О., Сало А.М., Клименко В.А., Цигилик Л.О., Юрчук А.В., 2009

**Розглянуто історію появи прискорювачів універсальних комп'ютерів. Подано вимоги до реконфігурованого прискорювача щодо приймання, опрацювання та передавання даних. Представлено модель спеціалізованого процесора та показано методи оптимізації архітектури цієї моделі під об'єм ПЛІС. Визначено основні вимоги щодо оптимізації архітектури апаратних ресурсів ПЛІС. Розглянуто етапи проектування прискорювачів універсальних комп'ютерів за допомогою САПР ХАМЕЛЕОН.**

**In this article the requirements to the reconfigurable accelerator in relation to data reception, processing and data transfer are given. The model of the specialized processor is presented and the architecture optimization methods of this model in accordance to the FPGA capacity are described. The basic requirements concerning architecture optimization of FPGA hardware resources are defined.**

**Вступ.** Потреба вирішення ресурсомістких задач та задач з інтенсивним використанням даних, таких як обробка мультимедійних даних в реальному часі, сучасне математичне моделювання і обробка вмісту спричинили формування нового типу програмних та апаратних засобів, які отримали назву реконфігурованих прискорювачів. Реконфігуровані прискорювачі все частіше використовуються для виконання складних задач обробки даних з метою зменшення навантаження на універсальні процесори та підвищення продуктивності комп'ютерних систем. При цьому можливість реконфігурування, тобто заміни вмісту реконфігурованого прискорювача новим спеціалізованим процесором, відкриває перспективи надання принципово нових властивостей та досягнення високих технічних характеристик комп'ютерної системи, до складу якої входить реконфігурований прискорювач [6].

**1. Поява прискорювачів універсальних комп'ютерів.** Використання в комп'ютерних системах прискорювачів не є новим підходом. Перші прискорювачі універсальних комп'ютерів з'явилися в середині 60-х років минулого століття, коли було набуто певного досвіду використання комп'ютерів у науковій та виробничій сферах, покращилась їх архітектура, але швидкодія була недостатньою для вирішення складних наукових проблем, насамперед завдань цифрової обробки сигналів. Для таких потреб почали створювати спеціалізовані процесори, які апаратно реалізували найскладніші алгоритми і під'єднувались до універсальних комп'ютерів як прискорювачі при виконанні цих алгоритмів. Зокрема, такі спеціалізовані процесори створювались для виконання