УДК 681.3

Один із підходів до апаратної реалізації бамп-мепінгу

Войтко В.В., Романюк О.В.
Вінницький національний технічний університет
oksana_gelika@mail.ru

Abstract

Voitko V., Romaniuk O. One approach to bump-mapping hardware realization. *This article presents the bump-mapping method that requires minimal hardware in comparison with hardware needed for classical bump-mapping realization. The method expels costly per-pixel transforming of normal vectors into tangent space and perturbing the interpolated normal vectors, which allows to save hardware and increase rendering speed.*

Вступ

Процес візуалізації реалістичних тривимірних зображень є досить трудомістким і характеризується застосуванням складних моделей освітлення та зафарбовування. Для кожної точки поверхні необхідно визначити інтенсивність кольору та екранні координати, при чому слід враховувати розташування джерела світла та спостерігача, оптичні властивості матеріалу, спектральні характеристики джерела світла та кривизну поверхні. У зв'язку із цим кінцева візуалізація супроводжується значними обчислювальними витратами.

Відомо, що для забезпечення більшої реалістичності зображення спочатку проводиться теселяція геометричних моделей, тобто розбиття на низькорівневі полігони, та здійснюються відповідні перетворення у вершинах цих полігонів (вершинні обчислення), а потім у всіх точках всередині цих полігонів (піксельні обчислення).

Основною стратегією апаратного прискорення формування графічних об'єктів є поділ обчислювального процесу на вершинні та піксельні компоненти [1]. Як правило, вершинні обчислення відбуваються у вершинних процесорах, в той час як піксельні обчислення проводяться у піксельних процесорах. За такого поділу вартість апаратних засобів повинна бути виправдана якістю та швидкістю візуалізації зображень.

Через те, що вершинні обчислення відбуваються у вершинному процесорі, вартість реалізації нової функції буде визначатися вартістю додаткових апаратних засобів піксельної обробки. Якщо ця функція має дуже специфічне застосування, то додаткове апаратне забезпечення важко виправдати, оскільки воно неефективно застосовуватиметься в програмних додатках, які не використовують дану функцію. В ігрових системах, де ефективно використовується кожен транзистор, додаткове апаратне забезпечення растеризації є особливо дорогим. Тому альтернативою надлишковим апаратним засобам є багаторазове використання наявних апаратних ресурсів.

Метою даного дослідження є підвищення ефективності використання апаратних ресурсів реалізації бамп-мепінгу.

Апаратна реалізація бамп-мепінгу

Модель освітлення Фонга [2,3,4] є досить популярною та широко використовується у графічних системах, не зважаючи на наявність більш точних моделей освітлення. У даній моделі вершинні вектори нормалі $\vec{N}$ інтерполюються і нормалізуються в кожному пікселі, а потім використовуються для обчислення інтенсивності піксела. При цьому досягається краща локальна апроксимація кривизни поверхні і, як наслідок, отримується більш реалістичне зображення. Вектор світла $\vec{L}$ та вектор півшляху $\vec{H} = \vec{L} + \vec{V}$, де $\vec{V}$ - вектор спостереження, обчислюються безпосередньо у площині зору або інтерполюються. На рис. 1 зображено апаратну реалізацію моделі освітлення Фонга.

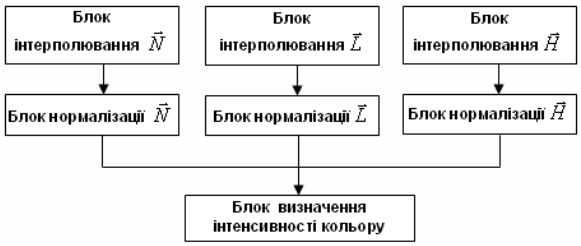

Рисунок 1 - Апаратна реалізація моделі Фонга.

Така реалізація значно підвищує вартість апаратного забезпечення растеризації. Однак у трьохмірних графічних додатках завжди прагнули досягнути вищої якості освітлення, а вдосконалення напівпровідникової технології робить модель освітлення Фонга більш практичною.

Візьмемо модель Фонга і текстурне апаратне забезпечення як основу для апаратної реалізації бамп-меппінгу.

Бамп-мепінг (bump-mapping) – це техніка імітації нерівностей поверхні (або моделювання мікрорельєфу) на пласкій поверхні без зміни геометрії [5,6]. Для кожного піксела поверхні виконується обчислення освітлення, беручи значення зі спеціальної карти висот. Як правило, це 8-бітна чорно-біла текстура і значення кольору текстури не накладаються як звичайні текстури, а використовуються для опису шорсткості поверхні. Колір кожного піксела визначає висоту відповідної точки рельєфу: більші значення означають більшу висоту над вихідною поверхнею, а менші, відповідно, меншу висоту.

Бамп-мепінг дозволяє деталізувати такі природні поверхні, як стіни з цеглин, бруківка, галька чи пори на шкірі, що мають дрібні деталі поверхні, які надто мілкі, щоб відображати їх за допомогою геометричних мозаїчних представлень поверхні. Bump Mapping найбільш доцільно використовувати, коли джерело світла рухається відносно поверхні.

При застосуванні бамп-мепінгу вектор нормалі $\vec{N}$ поверхні модифікується в точці P вектором збурення, який розрахований з наперед заданої карти висот. Якщо $P(u,v)$ відображає позицію на параметричній поверхні, то вектор нормалі поверхні в цій точці може бути отримано за формулою

$$\vec{N} = \vec{P}_u \times \vec{P}_v \,, \qquad (1)$$

де $\vec{P}_u$ і $\vec{P}_v$ часткові похідні P по параметричним напрямкам u і v відповідно.

Для того, щоб отримати вектор нормалі збуреної поверхні, поверхня модифікується шляхом додавання функції $F(u,v)$ карти висот. Разом P і F визначають нове зміщення поверхні $P'(u,v)$:

$$P'(u,v) = P(u,v) + F(u,v)\vec{N} \qquad (2)$$

Вектор нормалі $\vec{N}'$ збуреної поверхні визначається за формулою

$$\vec{N}' = \vec{P}'_u \times \vec{P}'_v \,. \qquad (3)$$

Оскільки значення функції $F(u,v)$ мале, то нормаль збуреної поверхні може бути апроксимована:

$$\vec{N}' = \vec{N} + F_u(\vec{N} \times \vec{P}_v) + F_v(\vec{P}_u \times \vec{N}) \qquad (4)$$

Кінцевий крок – нормалізація вектора $\vec{N}'$, який використовується для обчислення освітлення.

На рис.2 запропонована апаратна реалізація техніки бамп-мепінг.

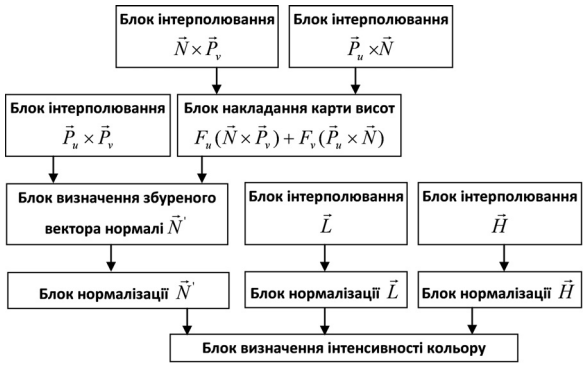

Рисунок 2 – Апаратна реалізація бамп-мепінгу

Оскільки вектори $\vec{P}_u$ і $\vec{P}_v$ необмежені, то апаратні засоби для інтерполювання, додавання, множення і нормалізації векторів повинні мати значно більшу точність, ніж та, яка вимагається для обмежених векторів.

У роботі [7] була запропонована апроксимація для такого впровадження, де $\vec{P}_v \times \vec{N}$ і $\vec{N} \times \vec{P}_u$ залишаються постійними вздовж полігона. Якщо уникати інтерполяції, то ця апроксимація може призводити до появи артефактів [7].

У даній роботі розглядається впровадження бамп-мепінгу, яке максимально використовує апаратні засоби реалізації моделі освітлення Фонга і яке не потребує значних інвестицій в спеціальне апаратне забезпечення. Головна ідея полягає в тому, щоб перенести обчислення бамп-мепінгу в іншу систему відліку. Оскільки модель освітлення є функцією векторних операцій (таких як скалярний добуток) між збуреним вектором нормалі та іншими векторами, то вони можуть бути обчислені відносно будь-якої системи відліку. Тому можна перекласти частину обчислень бамп-мепінгу на вершинний процесор. В результаті до ланцюга обчислення освітлення Фонга додаються мінімальні апаратні засоби.

Новий алгоритм бамп-мепінгу та його апаратна реалізація

У роботі [5] класична апроксимація бамп-мепінгу передбачає, що поверхня є локально гладкою в кожній точці. Тому збурення є функцією лише локальної дотичної площини. Дотичний простір є локальною системою координат, ортонормальний базис якої включає нормалізований незбурений вектор нормалі $\vec{N}$, вектор дотичної $\vec{T} = \dfrac{\vec{P}_u}{\left|\vec{P}_u\right|}$ та вектор бінормалі $\vec{B} = \left(\vec{N} \times \vec{T}\right)$. Вектори $\vec{T}, \vec{B}$ і $\vec{N}$ формують ортонормальну систему координат, в якій буде

виконуватись бамп-меппінг. $\vec{P}_v$ знаходиться у площині дотичної та бінормалі і визначається як:

$$\vec{P}_v = (\vec{T}\cdot\vec{P}_v)\vec{T} + (\vec{B}\cdot\vec{P}_v)\vec{B} \quad (5)$$

Тому

$$\vec{P}_v \times \vec{N} = (\vec{B}\cdot\vec{P}_v)\vec{T} - (\vec{T}\cdot\vec{P}_v)\vec{B} \quad (6)$$

Вектор збурення нормалі $\vec{D}$ матиме вигляд:

$$\begin{aligned}\vec{D} &= -F_u(\vec{P}_v\times\vec{N}) - F_v(\vec{N}\times\vec{P}_u) = \\ &= -F_u(\vec{P}_v\times\vec{N}) - F_v\left|\vec{P}_u\right|\vec{B} = \\ &= -F_u(\vec{B}\cdot\vec{P}_v)\vec{T} - (F_v\left|\vec{P}_u\right| - F_u(\vec{T}\cdot\vec{P}_v))\vec{B}\end{aligned} \quad (7)$$

Оскільки $\vec{D}$ виражений у дотичному просторі, то можна призначити векторам $\vec{T}, \vec{B}$ і $\vec{N}$ зручну орієнтацію, таку як $\vec{T}=[1,0,0], \vec{B}=[0,1,0]$ і $\vec{N}=[0,0,1]$. Тоді визначити збурений вектор нормалі у дотичному просторі стає дуже просто

$$\vec{N}'_{TBN} = \vec{N} + \vec{D} = [0,0,c] + \vec{D} = [a,b,c] \quad (8)$$

де

$$\begin{aligned} a &= -F_u(\vec{B}\cdot\vec{P}_v) \\ b &= F_u(\vec{T}\cdot\vec{P}_v) - F_v\left|\vec{P}_u\right| \\ c &= \left|\vec{P}_u\times\vec{P}_v\right| \end{aligned} \quad (9)$$

Коефіцієнти a, b і c є функцією самої поверхні (через $\vec{P}_u$ і $\vec{P}_v$) і карти висот (через F_u і F_v). Таким чином бамп-карта є фіксованою до поверхні, коефіцієнти можуть бути попередньо обчислені для цієї поверхні в кожній точці карти висот і збережені як текстурна карта. Компоненти текселя лежать в межах (-1;1).

Текстурна карта, яка містить збурені вектори нормалі, фільтрується як проста текстура, використовуючи, наприклад, трилінійну або анізотропну фільтрацію міп-текстурування [8,9]. Міп-текстурування полягає в тому, що для близько розташованих об'єктів використовується більша та детальніша текстура, а для віддалених – її спрощені копії. Щоб не було помітно розривів між міп-рівнями, на переходах прискорювач розраховує лінійну комбінацію кольорів, обчислених за «ближчими» і «віддаленими» текстурами, що й називається трилінійною фільтрацією. Однак її недоліком є некоректний розрахунок кольорів для похилих поверхонь, тому в такому випадку краще використовувати анізотропну фільтрацію, яка покликана усунути цей недолік. Але такий метод вимагає великих обчислювальних витрат і висуває непомірні вимоги до пропускної здатності пам'яті. Природно, що у чистому вигляді, для розрахунку тривимірної графіки у реальному часі, такий метод не використовується, а застосовується його апроксимація. Для покращення результату такої апроксимації, перш за все, слід збільшити кількість текселів, які визначають колір пікселя, а також врахувати форму проекції, залежно від кута нахилу текстури. Варто також зазначити, що після інтерполяції міп-текстурування текстура не буде нормалізованою, тому необхідно її нормалізувати перед обчисленням освітленням.

Для обчислення освітлення слід перетворити вектор світла $\vec{L}$ та нормалізований вектор півшляху $\vec{H} = \dfrac{\vec{L}+\vec{V}}{\left|\vec{L}+\vec{V}\right|}$ у дотичний простір через матрицю 3х3, стовпцями якої є $\vec{T}, \vec{B}$ і $\vec{N}$. Тепер дифузна складова у моделі освітлення може бути обчислена за допомогою збуреного вектора нормалі, взятого з текстурної карти і трансформованого вектор світла: $\vec{N}'_{TBN}\cdot\vec{L}_{TBN}$. Аналогічно розраховуються інші складові моделі освітлення.

Перетворення світлового вектора та вектора півшляху повинно бути проведено у кожному пікселі; однак, якщо зміна локального дотичного простору вздовж полігона незначна, то ефективнішої апроксимації можна досягти шляхом перетворення векторів лише у вершинах полігона. Потім вони інтерполюються і нормалізуються у точках всередині полігона. Зазвичай таке припущення ефективно працює, оскільки дотичний простір змінюється швидко на ділянках з високою кривизною поверхні, і графічний додаток потребуватиме кращої теселяції поверхні на цих ділянках для зменшення геометричного фасетування [10].

Апаратні засоби растеризації, які необхідні для реалізації даного алгоритму бамп-меппінгу зображено на рис.3. Додавши мультиплексор до апаратних засобів, що зображені на рис.1, можна забезпечити реалізацію як освітлення Фонга, так і бамп-мепінгу. Хоча даний алгоритм потребує перетворення світлового вектора та вектора півшляху у дотичний простір у кожній вершині, зберігання трикомпонентної карти текстури замість двокомпонентної карти і наявності окремих карт для кожної поверхні, однак йому необхідний лише один мультиплексор, який додається до апаратних засобів реалізації освітлення Фонга, він уникає інтерполяції $\vec{P}_v\times\vec{N}$ і $\vec{N}\times\vec{P}_u$, збурення вектора нормалі у кожній точці, і не потребує додаткової точності, яка необхідна для виконання арифметичних операцій над необмеженими векторами. Якщо графічний додаток обмежується растеризатором, то він буде виконувати бамп-мепінг з тією ж швидкістю, що й освітлення Фонга.

Якщо текстурною картою задано функцію параметрів поверхні, то модель освітлення може бути обчислена у просторі об'єкта, а не у дотичному просторі. Тоді, текстура зберігає збурені вектори нормалей у просторі об'єкта, а світловий вектор $\vec{L}$ і вектор півшляху $\vec{H}$ перетворюються у простір об'єкта у вершинах полігона та інтерполюються. Таким чином, для всіх вершин досить провести одне матричне перетворення світлового вектора і вектора півшляху. Дане впровадження ефективно використовує апаратні засоби растеризації, зображені на рис.3, значно мінімізуючи витрати геометричного процесора.

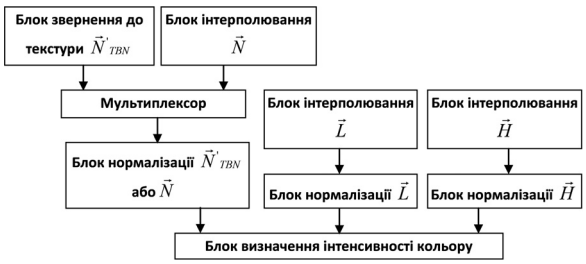

Рисунок 3 – Апаратна реалізація запропонованого алгоритму бамп-мепінгу

Основним недоліком даного методу є наявність залежності текстурної карти від параметрів поверхні. Оскільки текстурна карта є функцією поверхні, то одну і ту ж карту висот не можна використовувати для поверхонь з різними параметрами. Ця проблема особливо загострюється за умов обмеження текстурної пам'яті, наприклад в ігрових системах або під інтерактивного накладання бамп-карти на нову поверхню.

Подолати цю залежність можна, якщо припустити, що, локально, параметри поверхні є такими ж як у квадратної ділянки. Тоді $\vec{P}_u$ і $\vec{P}_v$ є ортогональними і однаковими за розміром. Щоб уникнути бамп-залежності від масштабу поверхні обираємо $\left|\vec{P}_u\right| = \left|\vec{P}_v\right| = k$, де k – постійна величина, яка задає відносну висоту нерівностей. Це, разом з умовою ортогональності, спрощує рівняння 8-9 до

$$\vec{N}'_{TBN} = \vec{N} + \vec{D} = [0,0,c] + \vec{D} = [a,b,c] \quad (10)$$

де

$$a = -\frac{F_u}{k}$$

$$b = -\frac{F_v}{k} \quad (11)$$

$$c = 1$$

Текстурна карта стає функцією лише карти висот, а не геометрії поверхні, тому вона може бути попередньо обчислена і використовуватись для будь-якої поверхні.

Припущення про квадратну ділянку є прийнятним для багатьох поширених поверхонь, таких як сфери, тори, поверхні обертання і пласких прямокутників. Якщо поверхня вже параметризована або може бути параметризована, апроксимація, представлена формулами 10-11, є прийнятною.

Висновки

У даній статті розглядається алгоритм бамп-мепінгу, який забезпечує мінімальні витрати апаратних засобів для його реалізації в порівнянні з апаратною реалізацією традиційного бамп-мепінгу. Така економія апаратних засобів досягається за рахунок виключення трудомістких піксельних кроків перетворення векторів у дотичний простір та збурення векторів нормалі. Натомість проводиться інтерполювання векторів, які були перетворені у дотичний простір у вершинах полігона, а попередньо обчислена збурена карта нормалі зберігається як текстура.

Література

1. Kurt Akeley, Reality Engine graphics, Proceedings of the 20th annual conference on Computer graphics and interactive techniques, p.109-116, September 1993.
2. Bui Tuong Phong, "Illumination for Computer Generated Pictures," Comm. ACM, Vol 18(6):311-317, June 1975.
3. CLAUSSEN, U. Real time phong shading. In Fifth Eurographics Workshop on Graphics Hardware (1989), D. Grimsdale and A. Kaufman, Eds.
4. CLAUSSEN, U. Onreducing the phong shading method. Computers and Graphics 14, 1 (1990), 73-81.
5. Kilgard J. A practical and robust bump-mapping technique for today's // In GDC 2000: Advanced OpenGL Game Development. - July 2000.
6. BLINN, J. F. 1978. Simulation of wrinkled surfaces. *Computer Graphics (SIGGRAPH '78 Proceedings) 12*, 3, 286.292.
7. ERNST, I., JACKEL, D., RUSSELER, H., AND WITTIG, O. Hardware supported bump mapping: A step towards higher quality real-time rendering. In l Oth Eurographics Workshop on Graphics Hardware (1995), pp. 63-70.
8. Heckbert, P., "Texture Mapping Polygons in Perspective," NYIT Tech. Memo No. 13, 1983.
9. Jérôme Maillot , Hussein Yahia , Anne Verroust, Interactive texture mapping, Proceedings of the 20th annual conference on Computer graphics and interactive techniques, p.27-34, September 1993
10. A. A. Kuijk , E. H. Blake, Faster phong shading via angular interpolation, Computer Graphics Forum, v.8 n.4, p.315-324, Dec. 1989 .

Надійшла до редколегії 03.03.2009