

С.І. ВЯТКИН

Інститут Автоматики і Електрометрії, СО РАН

С.А. РОМАНЮК

Вінницький національний технічний університет

СОЗДАНИЕ ОБЪЕКТОВ НА ОСНОВЕ ФУНКЦИЙ ВОЗМУЩЕНИЯ С ПОМОЩЬЮ ОБРАТНОГО ОБХОДА ВОСЬМЕРИЧНОГО ДЕРЕВА ДЕЛЕНИЯ ОБЪЕКТНОГО ПРОСТРАНСТВА

Поверхности на основе функций используются для множества задач в компьютерной графике, включая моделирование мягких или органических объектов, трехмерного морфинга, обнаружения столкновений и конструктивной твердой геометрии. Предлагается метод создания функциональных поверхностей на основе функций возмущения из полигональных моделей. Функциональные объекты строятся из свободных форм с аналитическими функциями возмущения, благодаря чему достигается высокий коэффициент геометрического сжатия трехмерных моделей.

Ключевые слова: полигональные модели, функционально заданные объекты, функции возмущения.

S.I VYATKIN

Mechanics and Institute, Russian Academy of Sciences

S.A. ROMANYUK

Vinnytsia National Technical University

BUILDING ON THE BASIS OF FUNCTIONS DISTURBANCES BY THE BACKWARD WALK OCTREE FISSION OBJECT SPACE

Surface-based functions are used for a variety of problems in computer graphics, including modeling soft or organic objects, three-dimensional morphing, collision detection, and constructive solid geometry. We propose a method of creating functional surfaces on the basis of perturbation functions of polygonal models. Functional objects are constructed from free-form analytic functions with indignation, thus achieving high geometrical compression ratio of three-dimensional models.

Keywords: polygonal models, are functionally defined objects, perturbation function

Введение

В настоящее время для генерации изображений в реальном времени используются алгоритмы растрования на плоскости и воксельно-базируемые алгоритмы. Основным достоинством первого подхода является низкая вычислительная сложность алгоритмов рендеринга. Для генерации изображения не требуется высоких вычислительных мощностей. К недостаткам можно отнести, в первую очередь, большое количество примитивов (треугольников). Чтобы получить полноценный трехмерный объект со сложной поверхностью, необходимо иметь огромное число треугольников, что по числу данных приближается к воксельным моделям. С целью геометрического сжатия и качественного отображения таких сложных объектов были разработаны патчи свободных форм, а также свободные формы на базе квадрик с аналитическими функциями возмущения [1, 2]. В работе [3] была решена задача автоматического преобразования полигональных моделей в функциональные модели с помощью объемных патчей свободных форм. Известна работа [4] в которой решалась задача конверсии для вариационных неявных поверхностей, в которой применялся итерационный метод. В этом методе на преобразование сложных моделей с разрешением объема 170x170x373 требовались часы работы компьютера (R10000 MIPS процессор, 195MHz, SGI Origin). Подобная задача решалась в работе [5]. Известен также метод [6] преобразования полигональных моделей в функциональное описание на основе радиальных функций. В этом методе полигональная модель аппроксимировалась функциями второго порядка (патчами). Патч – это квадрика, которая аппроксимирует поверхность около выбранной точки.

Данная работа посвящена автоматическому преобразованию полигональных моделей в функциональное описание на базе свободных форм с аналитическими функциями возмущения. В данном методе создание объекта сводится к задаче деформации базовой поверхности (квадрики) нужным образом (в соответствии с полигональной моделью), а не к аппроксимации ее примитивами (патчами). В предлагаемом методе время преобразования моделей составляет от секунды до нескольких секунд на ПК (Intel Pentium 4 Processor, 2800 MHz).

Функционально заданные объекты

В работе [2] предлагалось описывать сложные геометрические объекты, задавая функцию отклонения (функция второго порядка в неявном виде) от базовой квадрики. Алгебраическим неравенством второй степени (с тремя неизвестными x, y, z) называется всякое неравенство вида:

$$F(x, y, z) = Ax^2 + By^2 + Cz^2 + Dxy + Exz + Fyz + Gx + Hy + Iz + K \leq 0, \quad (1)$$

где x, y и z - пространственные переменные.

Квадрика определяется с помощью вещественной непрерывной описывающей функции трех

переменных (x_1, x_2, x_3) в виде $F(X) \geq 0$. Рассматриваются квадрики как замкнутые подмножества Эвклидова пространства E^n , определяемые описывающей функцией $F(X) \geq 0$. Где F - непрерывная вещественная функция и $X = (x_1, x_2, x_3)$ – задаваемая координатными переменными точка в E^n . Здесь $F(X) > 0$ задает точки внутри квадрики, $F(X) = 0$ – точки на границе и $F(X) < 0$ – точки, лежащие снаружи и не принадлежащие квадрике. Квадрика с возмущением (свободная форма) это композиция базовой квадрики и возмущения

$$R_i(x, y, z) = \begin{cases} Q_i^3(x, y, z), & \text{if } Q_i(x, y, z) \geq 0 \\ 0, & \text{if } Q_i(x, y, z) < 0 \end{cases} \quad (2)$$

где $R(x, y, z)$ – возмущение. Полученная поверхность будет гладкой (рис. 1), и потребуется небольшое количество функций возмущения для создания сложных форм поверхностей. Предложенный способ описания объектов трехмерных сцен базовыми поверхностями и функциями возмущения имеет компактное описание, что позволяет существенно уменьшить объем хранимых и передаваемых данных от 10 до 1000 раз, а также улучшить качество изображений.

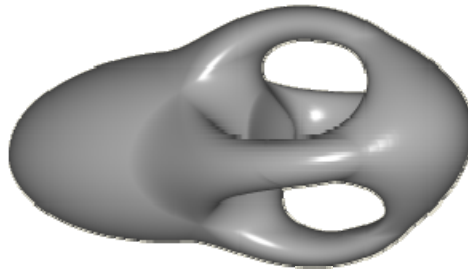


Рис. 1. Свободная форма на основе одной базовой квадрики и четырех функций возмущения второй степени

Таким образом, задача конструирования объекта сводится к задаче деформации базовой поверхности нужным образом, а не к аппроксимации ее примитивами.

Вычисление коэффициентов уравнения базовой квадрики

С целью корректного преобразования и максимально компактного функционального описания, сложная полигональная модель, состоящая из объединения нескольких простых геометрических моделей, должна быть сегментирована.

Чтобы отличить базовую квадратичку от функций возмущения, коэффициенты ее уравнения находим заранее. Для этого необходимо квадратичку вписать в исходную полигональную модель, точнее найти коэффициенты базовой квадрики по девяти габаритным точкам (вершинам полигональной сетки). Коэффициенты уравнений функций возмущения в дальнейшем будут обозначаться буквами А, В, С, D, E, F, G, H, I, K, а коэффициенты уравнений базовых квадратичек - буквой q.

При следующем задании квадрики

$$Q = \begin{pmatrix} q_{xx} & \frac{q_{xy}}{2} & \frac{q_{xz}}{2} & \frac{q_x}{2} \\ \frac{q_{xy}}{2} & q_{yy} & \frac{q_{yz}}{2} & \frac{q_y}{2} \\ \frac{q_{xz}}{2} & \frac{q_{yz}}{2} & q_{zz} & \frac{q_z}{2} \\ \frac{q_x}{2} & \frac{q_y}{2} & \frac{q_z}{2} & q \end{pmatrix} \quad (3)$$

Значение функции заданной (3) в произвольной точке $P[x, y, z]$ будет:

$$Q(P[x, y, z]) = q_{xx}x^2 + q_{yy}y^2 + q_{zz}z^2 + q_{xy}xy + q_{xz}xz + q_{yz}yz + q_x x + q_y y + q_z z + q \quad (4)$$

Т.к. значение Q на поверхности равно нулю, то для нахождения коэффициентов квадрики по девяти $P_1[x_1, y_1, z_1] \div P_9[x_9, y_9, z_9]$ точкам получаем систему линейных уравнений.

$$\begin{cases} Q(P_1) = 0 \\ Q(P_2) = 0 \\ \dots \\ Q(P_i) = 0 \\ \dots \\ Q(P_9) = 0 \end{cases} \quad (5)$$

Подставляя (4), получаем:

$$\left. \begin{aligned} & q_{xx}x_1^2 + q_{yy}y_1^2 + q_{zz}z_1^2 + q_{xy}x_1y_1 + q_{xz}x_1z_1 \\ & + q_{yz}y_1z_1 + q_x x_1 + q_y y_1 + q_z z_1 + q = 0 \\ & q_{xx}x_2^2 + q_{yy}y_2^2 + q_{zz}z_2^2 + q_{xy}x_2y_2 + q_{xz}x_2z_2 \\ & + q_{yz}y_2z_2 + q_x x_2 + q_y y_2 + q_z z_2 + q = 0 \\ & \dots \\ & q_{xx}x_i^2 + q_{yy}y_i^2 + q_{zz}z_i^2 + q_{xy}x_iy_i + q_{xz}x_iz_i \\ & + q_{yz}y_iz_i + q_x x_i + q_y y_i + q_z z_i + q = 0 \\ & \dots \\ & q_{xx}x_9^2 + q_{yy}y_9^2 + q_{zz}z_9^2 + q_{xy}x_9y_9 + q_{xz}x_9z_9 \\ & + q_{yz}y_9z_9 + q_x x_9 + q_y y_9 + q_z z_9 + q = 0 \end{aligned} \right\} \quad (6)$$

Решением системы уравнений (7) со свободным членом q (положим его равным -K) для заданных $P_1[x_1, y_1, z_1] \div P_9[x_9, y_9, z_9]$ будут девять искомым коэффициентов $q_{xx}, q_{yy}, q_{zz}, q_{xy}, q_{xz}, q_{yz}, q_x, q_y, q_z$.

Далее коэффициенты этой квадрати будут использоваться при преобразовании в функциональную модель. Коэффициенты квадрати вычисляются с помощью метода Крамера.

$$\begin{bmatrix} x_1^2 & y_1^2 & z_1^2 & x_1y_1 & x_1z_1 & y_1z_1 & x_1 & y_1 & z_1 \\ x_2^2 & y_2^2 & z_2^2 & x_2y_2 & x_2z_2 & y_2z_2 & x_2 & y_2 & z_2 \\ x_3^2 & y_3^2 & z_3^2 & x_3y_3 & x_3z_3 & y_3z_3 & x_3 & y_3 & z_3 \\ x_4^2 & y_4^2 & z_4^2 & x_4y_4 & x_4z_4 & y_4z_4 & x_4 & y_4 & z_4 \\ x_5^2 & y_5^2 & z_5^2 & x_5y_5 & x_5z_5 & y_5z_5 & x_5 & y_5 & z_5 \\ x_6^2 & y_6^2 & z_6^2 & x_6y_6 & x_6z_6 & y_6z_6 & x_6 & y_6 & z_6 \\ x_7^2 & y_7^2 & z_7^2 & x_7y_7 & x_7z_7 & y_7z_7 & x_7 & y_7 & z_7 \\ x_8^2 & y_8^2 & z_8^2 & x_8y_8 & x_8z_8 & y_8z_8 & x_8 & y_8 & z_8 \\ x_9^2 & y_9^2 & z_9^2 & x_9y_9 & x_9z_9 & y_9z_9 & x_9 & y_9 & z_9 \end{bmatrix} \begin{bmatrix} q_{xx} \\ q_{yy} \\ q_{zz} \\ q_{xy} \\ q_{xz} \\ q_{yz} \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} K \\ K \\ K \\ K \\ K \\ K \\ K \\ K \\ K \end{bmatrix} \quad (7)$$

Преобразование полигональных моделей в функциональное описание

Преобразуемая модель (рис. 2) задается в кубе с центром (0, 0, 0) и координатами от -1 до 1 по x, y, z, в системе координат объекта. При восьмеричном делении мы делим коэффициенты уравнения A, B, C, D, E, F на 4, коэффициенты X Y Z на два (этим мы уменьшаем размеры куба (8)) и после этого делаем сдвиг на вектор (+/- 0.5; +/- 0.5; +/- 0.5), т.е. в один из восьми подкубов.

$$\begin{aligned} A' &= A/4, B' = B/4, C' = C/4, D' = D/4, \\ E' &= E/4, F' = F/4, \\ G' &= G/2 + i*A/2 + j*D/4 + k*E/4, \\ H' &= H/2 + i*D/4 + j*B/2 + k*F/4, \\ I' &= I/2 + i*E/4 + j*F/4 + k*C/2, \\ K' &= K + i*G/4 + j*N/4 + k*I/4, \end{aligned} \quad (8)$$

где коэффициенты без штриха берутся с предыдущего шага рекурсии.

Прежде чем преобразовывать полигональные сегментированные модели в функциональные, необходимо найти все точки полигональной поверхности, подобно тому, как это было описано в работе [7]. Далее отмечаются все ветви и листья дерева, где имело место пересечение модели с подкубами разных уровней восьмеричного дерева деления пространства объекта. Это просто сделать, поскольку известны все пересеченные листья дерева.

$$\begin{aligned} N_x &= G' / \sqrt{G'^2 + H'^2 + I'^2} \\ N_y &= H' / \sqrt{G'^2 + H'^2 + I'^2} \\ N_z &= I' / \sqrt{G'^2 + H'^2 + I'^2} \end{aligned} \quad (9)$$

Главная идея заключается в следующем. Если с помощью рекурсивного деления объектного

пространства [2] можно визуализировать функционально заданные объекты (вычислять точки поверхности, нормали в этих точках, освещенность и т.д.), то можно решить и обратную задачу - по заданным точкам и

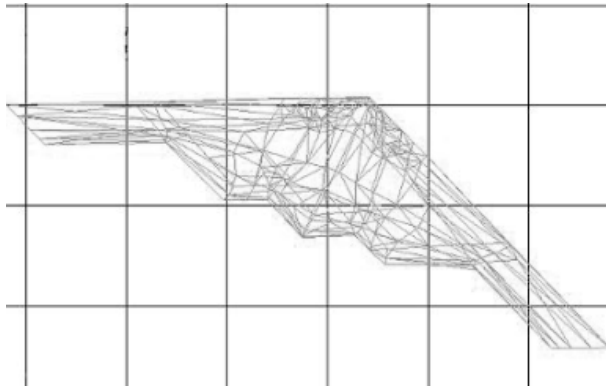


Рис. 2. Полигональная модель (4160 вершин, 8192 полигонов, общий объем данных-162 кб)

определяются одинаковые коэффициенты уравнений



Рис. 3. Функционально заданный объект, 36 функций возмущения

(рис. 3).

нормалам (9), найти функции, описывающие данный объект. Для этого необходимо решить систему уравнений (8). Данное преобразование состоит из следующих основных задач: 1) определение коэффициентов уравнений для самого низкого уровня деления пространства объекта (листьев восьмеричного дерева), 2) обратный обход восьмеричного дерева деления объектного пространства, 3) поиск коэффициентов функций, 4) минимизация функций. Таким образом, стоит задача деформации базовых квадрик в соответствии с полигональными сетками (рис. 2).

Далее решаются системы линейных уравнений на каждом уровне рекурсии, определяются одинаковые коэффициенты уравнений (с учетом установленного порога точности приближения) и минимизируется количество функций на каждом уровне, только после обработки всех уравнений данного уровня происходит всплытие на следующий верхний уровень, и так процесс повторяется до самого корня восьмеричного дерева. В результате получается необходимый минимум функций, описывающих данный объект в формате описания функционально заданных объектов на базе квадрик с аналитическими функциями возмущения

Заключение

В данной работе представлен метод автоматического преобразования полигональных моделей в функциональные на базе квадрик с аналитическими функциями возмущения (свободные формы). В методе на базе свободных форм преобразование происходит с помощью обратного обхода дерева деления объектного пространства, что является обратной задачей визуализации. Таким образом, прямое рекурсивное деление объектного пространства используется для визуализации функциональных объектов, а обратное - для поиска функций, описывающих объект. Для анализа отклонений (отклонение вершин и нормалей) был разработан анализатор критериев отклонений. Анализ степени приближения полученной функционально заданной поверхности к полигональной сетке осуществляется по двум метрикам. Это отклонение вершин треугольной сетки от поверхности функционального объекта. Отклонение нормалей к вершинам треугольной сетки от значений нормалей в точках функциональной поверхности. С помощью данного метода возможно преобразование полигональных моделей в функциональные без визуальных и процентных потерь.

Литература

1. Вяткин С. И., Долговесов Б. С. Поверхности и патчи свободных форм на основе скалярных и аналитических функций возмущения // Труды 12-й Междунар. конф. "Графикон-2002". Нижний Новгород, 2002. С. 147-152.
2. Вяткин С.И., Долговесов Б.С., Есин А.В. и др. Геометрическое моделирование и визуализация функционально заданных объектов // Автотметрия. 1999. №6. С. 84-92.
3. Вяткин С.И., Долговесов Б.С., Рухлинский А.В. Создание сложных поверхностей на базе функций возмущения из полигональных сеток // Труды 16-й Междунар. конф. по компьютерной графике и ее приложениям "Графикон-2006", Новосибирск, 2006. С. 192-197.
4. Gary Yngve, Grey Turk. "Robust Creation of Implicit Surfaces from Polygonal Meshes" // IEEE Transactions on Visualization and Computer Graphics. Vol.8., No.4., October-December 2002. P. 346-359.
5. Shen, C. O'Brien, J. F. Shewchuk, J. R. Interpolating and Approximating Implicit Surfaces From Polygon Soup // ACM TRANSACTIONS ON GRAPHICS 2004, VOL 23; PART 3, P. 893-901.
6. Ohtake Y., Belyaev A., Alexa M., Seidel H.-P. "Multi-level partition of unity implicits", ACM Transactions on Graphics (SIGGRAPH 03 Proceedings), vol. 22, No. 3, 2003, P.463-470.
7. Вяткин С.И., Долговесов Б.С. Создание баз данных на основе скалярных функций возмущения // Труды 16-й Междунар. конф. по компьютерной графике и ее приложениям "Графикон-2006". (Новосибирск, ИВММГ СО РАН, 2006). С. 477-480.

References

1. Vyatkin S.I., Dolgovesov B.S. Surfaces and patches of free forms on the basis of scalar and analytic perturbation functions // Proceedings of 12th Int. Conf. " GraphiCon -2002 ." Nizhny Novgorod , 2002 . S. 147-152 .
2. Vyatkin S.I., Dolgovesov B.S., Esin A. And other geometric modeling and visualization of functionally defined objects // Optoelectronics . 1999 . N6. S. 84-92
3. Vyatkin S.I., Dolgovesov B.S., Ruhlinsky A.V. The creation of complex surfaces based on perturbation functions of the polygonal mesh // Proceedings of the 16th Intern. Conf. on Computer Graphics and Applications " GraphiCon -2006" , Novosibirsk, 2006 . S. 192-197 .
4. Gary Yngve, Grey Turk. "Robust Creation of Implicit Surfaces from Polygonal Meshes" // IEEE Transactions on Visualization and Computer Graphics. Vol.8., No.4., October-December 2002 . P. 346-359 .
5. Shen, C. O'Brien, J. F. Shewchuk, J. R. Interpolating and Approximating Implicit Surfaces From Polygon Soup // ACM TRANSACTIONS ON GRAPHICS 2004 , VOL 23 ; PART 3 , P. 893-901 .
6. Ohtake Y., Belyaev A., Alexa M., Seidel H.-P. "Multi-level partition of unity implicits", ACM Transactions on Graphics (SIGGRAPH 03 Proceedings), vol. 22 , No. 3, 2003 , P.463- 470.
7. Vyatkin S.I., Dolgovesov B.S. Creating a database on the basis of scalar perturbation functions // Proceedings of the 16th Intern. Conf. on Computer Graphics and Applications " GraphiCon -2006 ." (Novosibirsk, IVMMG Russian Academy of Sciences , 2006). С. 477-480 .

Рецензія/Peer review : 2.7.2013 р. Надрукована/Printed :24.10.2013 р.

Рецензент:

УДК 004.925

М.Д. ОБІДНИК, С.О. РОМАНЮК

Вінницький національний технічний університет

МЕТОД РОЗПАРАЛЕЛЕННЯ РЕНДЕРИНГУ ТА ЗАСОБИ ДЛЯ ЙОГО РЕАЛІЗАЦІЇ

Запропоновано метод розпаралелення рендерингу, який дозволяє незалежно розраховувати інтенсивність кольору в складових точках трикутника і підвищити продуктивність зафарбовування. Отримано формули для незалежного розрахунку векторів нормалей у рядку растеризації трикутника. Запропоновано структуру спеціалізованого процесора для зафарбовування полігонів.

Ключові слова: комп'ютерна графіка, зафарбовування, розпаралелення, рендеринг, текстура освітлення, вектори нормалі.

M.D.OBIDNYK, S.O.ROMANYUK
Vinnitsia National Technical University

PARALLEL RENDERING METHOD AND MEANS FOR REALIZATION

The method of parallelization rendering, which allows independent calculate the intensity of the color components in the triangle points and increase productivity shading was proposed. Formulas for calculating independent normal vectors in the row triangle rasterization was obtained. The structure of a specialized processor for shading polygons was proposed.

Keywords: computer graphics, shading, parallelization, rendering, lighting texture, normal vectors.

Вступ

У процесі формування зображення тривимірної сцени використовується велика кількість однотипних, незалежних одна від одної операцій, що дає можливість розпаралелити розрахунки. Тому в сучасних графічних системах збільшують кількість процесорів, що можуть виконувати обчислення одночасно. Передові професійні відеокарти містять 2048 – 3072 потокових процесорів [1, 2]. Це забезпечує виконання паралельної обробки на рівні групи примітивів, коли полігони обробляються одночасно та незалежно один від одного. При цьому інтерполяційні методи векторів та інтенсивності кольору використовують покрокові алгоритми і тому відповідні розрахунки не можуть виконуватись одночасно. Для того, щоб забезпечити виконання паралельної обробки на рівні елемента зображення, необхідно знайти співвідношення для розрахунку освітлення у кожній точці полігону, що не будуть залежати від проміжних значень.

Теоретична основа та практична реалізація методу розпаралелення рендерингу

За основу для виведення формул паралельного визначення векторів нормалей доцільно використаємо метод сферично-кутової інтерполяції, запропонований у роботі [3]. Формула для сферично-кутової інтерполяції векторів має вигляд:

$$\vec{N}_t = \vec{N}_A \cdot \cos((t-1) \cdot \psi) + \vec{N}_K \sin((t-1) \cdot \psi), \quad (1)$$

де \vec{N}_t – поточний вектор нормалі, що розраховується; t – позиція пікселя у PPT (рядок растеризації трикутника), $t \in [0, 1]$; \vec{N}_A і \vec{N}_B – нормалі у початковій та кінцевій точках PPT; \vec{N}_K – допоміжний вектор, який обчислюється за формулою $\vec{N}_K = \vec{N}_B - \vec{N}_A (\vec{N}_B \cdot \vec{N}_A) / \sqrt{1 - (\vec{N}_B \cdot \vec{N}_A)^2}$ [33]; ψ – приріст кута між векторами у PPT, що розраховується за формулою $\psi = \phi / (w-1)$, де $\phi = \arccos(\vec{N}_A \cdot \vec{N}_B)$ – кут між