УДК 681.5.18

МЕТОД СЖАТИЯ ГЕОМЕТРИЧЕСКИХ ДАННЫХ С ПРИМЕНЕНИЕМ ФУНКЦИЙ ВОЗМУЩЕНИЯ*

С. И. Вяткин, Б. С. Долговесов

Институт автоматики и электрометрии СО РАН,
630090, г. Новосибирск, просп. Академика Коптюга, 1
E-mail: sivser@mail.ru

Рассмотрены проблемы компактного полигонального описания объектов. Предложен метод сжатия без потерь геометрических данных на основе функций возмущения. Показаны преимущества такого подхода перед известными алгоритмами преобразования трёхмерных моделей для быстрой передачи информации и компактного её хранения.

Ключевые слова: функции возмущения, теоретико-множественные операции, сжатие геометрических данных.

DOI: 10.15372/AUT20180403

Введение. С развитием компьютерных технологий, сети Интернет и расширением сфер их применения становится актуальным компактное описание сложных компьютерных трёхмерных моделей. Базы данных таких объектов могут занимать большой объём памяти. Загрузка файлов (особенно из сети Интернет) может происходить продолжительное время. Для уменьшения объёма памяти, необходимого для хранения полигонального описания, разработаны различные алгоритмы сжатия геометрических данных.

В полигональных моделях геометрическая структура, именуемая геометрией, состоит из множества точек, а также топологии, характеризующей взаимосвязи между смежными вершинами. Описание завершается списком атрибутов (нормали, цвета, текстуры). Большинство методов сжатия геометрических данных базируется на кодировании связей полигонального представления объектов и описании вершин и граней охватывающего дерева путём переупорядочивания вершин [1, 2] или использования дополнительной информации, например степени вершин, задающей способ подключения вершины к предыдущей последовательности [3].

Таким образом, список вершин состоит из связей и геометрии, в которых применяется дифференциальное кодирование или прогнозирование позиций, а не абсолютные координаты вершин. Методы одиночного разрешения описаны в работах [1–5]. Методы так называемого прогрессивного геометрического сжатия являются расширениями методов одиночного разрешения [6–10]. Естественно, если модели остаются в рамках полигонального описания, высокой степени сжатия ожидать не приходится. Так, сложный файл VRML (Virtual Reality Modeling Language) можно уменьшить на 2,33 % от первоначального размера для 12-битовой квантизации и только на 1,67 % для 10-битовой квантизации.

Сделать более компактным описание трёхмерных объектов можно с помощью функций. Поверхности на основе функций используются для множества задач в компьютерной графике, включая моделирование мягких или органических объектов, трёхмерного морфинга, обнаружения столкновений и конструктивной твёрдой геометрии. Хотя операции для функциональных объектов просты, создание форм является большой проблемой. В

*Работа выполнена при финансовой поддержке Федерального агентства научных организаций (государственная регистрация № АААА-А17-117062110016-4).

работе [11] решалась задача конверсии для вариационных неявных поверхностей с применением итерационного метода. Однако на преобразование сложных моделей с разрешением $170 \times 170 \times 373$ вокселей требовались часы работы компьютера (R10000, MIPS-процессор, 195 МГц, SGI Origin). Итерационный подход описан также в [12]. В [13] представлен метод преобразования полигональных моделей в трёхмерные на основе радиальных функций. В этом методе полигональная модель аппроксимировалась функциями второго порядка, где искомая функция есть сумма патчей, помноженная на функцию контейнера (патч — квадрика, аппроксимирующая поверхность около выбранной точки). Функция контейнера является множителем, который гарантирует отсутствие влияния патча на другие патчи за границей заданной области. В этом подходе максимальный коэффициент сжатия равен четырём.

Цель предлагаемой работы — создание быстрого метода с высоким коэффициентом сжатия без потерь геометрических данных на основе функций возмущения.

Функции возмущения. Для описания сложных геометрических объектов используются функции отклонения (второго порядка) от базовой квадрики [14]. Функционально заданные поверхности строятся из поверхностей второго порядка (квадрик) с аналитическими функциями возмущения, благодаря чему достигается высокий коэффициент геометрического сжатия реалистичных трёхмерных объектов. Рассматриваются поверхности как замкнутые подмножества евклидова пространства E^3, определяемые описывающей функцией $F(x, y, z) \geq 0$, где F — непрерывная вещественная функция; x, y, z — задаваемая координатными переменными точка в E^3. Функция $F(x, y, z) > 0$ описывает точки внутри поверхности, $F(x, y, z) = 0$ — точки на границе, $F(x, y, z) < 0$ — точки, лежащие снаружи и не принадлежащие поверхности.

Алгебраическим неравенством второй степени (с тремя неизвестными x, y, z) называется всякое неравенство

$$F(x, y, z) = A_{11}x^2 + A_{22}y^2 + A_{33}z^2 + A_{12}xy + A_{13}xz + A_{23}yz + A_{14}x + A_{24}y + A_{34}z + A_{44} \geq 0. \quad (1)$$

Можно записать это неравенство в матричном виде:

$$\begin{pmatrix} x & y & z & 1 \end{pmatrix} \begin{pmatrix} A_{11} & A_{12/2} & A_{13/2} & A_{14/2} \\ A_{12/2} & A_{22} & A_{23/2} & A_{24/2} \\ A_{13/2} & A_{23/2} & A_{33} & A_{34/2} \\ A_{14/2} & A_{24/2} & A_{34/2} & A_{44} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \geq 0, \quad (2)$$

или общее неравенство второй степени относительно пространственных переменных x, y, z:

$$(A_{11}x + A_{12}y + A_{13}z + A_{14})x + (A_{21}x + A_{22}y + A_{23}z + A_{24})y +$$

$$+ (A_{31}x + A_{32}y + A_{33}z + A_{34})z + A_{41}x + A_{42}y + A_{43}z + A_{44} \geq 0, \quad (3)$$

где $A_{ik} = A_{ki}$, $i, k = 1, 2, 3, 4$.

Свободные формы строятся с помощью квадрик и представляются композицией базовой квадрики и возмущений:

$$F'(x, y, z) = F(x, y, z) + \sum_{i=1}^{N} f_i R_i(x, y, z). \quad (4)$$

Здесь f_i — формфактор, $R(x, y, z)$ — возмущение:

$$R_i(x,y,z) = \begin{cases} Q_i^3(x,y,z), & \text{если } Q_i(x,y,z) \geq 0, \\ 0, & \text{если } Q_i(x,y,z) < 0, \end{cases} \tag{5}$$

где $Q(x, y, z)$ — возмущающая квадрика.

Геометрическая модель создаёт условия для конструирования объектов и их композиций различной сложности. Для этого применяется множество геометрических операций, определяемое математически следующим образом [15]:

$$M^1 + M^2 + \ldots + M^n \to M. \tag{6}$$

Для формирования моделей сложных объектов на базе функций возмущения используются теоретико-множественные объединения и пересечения, осуществляемые с применением булевых операций. Бинарная операция объектов G_1 и G_2 означает операцию $G_3 = \Phi_j(G_1, G_2)$ определения

$$f_3 = \psi(f_1(x,y,z), f_2(x,y,z)) \geq 0, \tag{7}$$

где ψ — непрерывная вещественная функция двух переменных.

Для решения данной задачи представляют интерес, прежде всего, операции объединения и пересечения.

Полная теория геометрических преобразований неявных поверхностей изложена в [16].

Метод сжатия геометрических данных. В целях корректного преобразования и максимально компактного функционального описания исходная полигональная модель (рис. 1), состоящая из объединения нескольких простых геометрических моделей, должна быть сегментирована (рис. 2). Чтобы отличить базовую квадрику от функций возмущения, коэффициенты её уравнения находим заранее. Для этого необходимо квадрику вписать в простую полигональную модель (рис. 3), точнее, найти коэффициенты базовой квадрики по девяти габаритным точкам (вершинам полигональной сетки). Коэффициенты уравнений функций возмущения (1): A_{11}, A_{22}, A_{33}, A_{12}, A_{13}, A_{23}, A_{14}, A_{24}, A_{34}, A_{44} далее будут обозначаться буквами $A, B, C, D, E, F, G, H, I, K$, а коэффициенты уравнений базовых квадрик — буквой q. При следующем задании квадрики

$$Q = \begin{pmatrix} q_{xx} & q_{xy}/2 & q_{xz}/2 & q_x/2 \\ q_{xy}/2 & q_{yy} & q_{yz}/2 & q_y/2 \\ q_{xz}/2 & q_{yz}/2 & q_{zz} & q_z/2 \\ q_x/2 & q_y/2 & q_z/2 & q \end{pmatrix}. \tag{8}$$

Рис. 1. Исходная полигональная модель

Рис. 2. Простые полигональные объекты

Рис. 3. Простой полигональный (слева) и функционально базируемый объекты

Значение функции, заданной (8) в произвольной точке $P[x, y, z]$, будет иметь вид

$$Q(P[x,y,z]) = q_{xx}x^2 + q_{yy}y^2 + q_{zz}z^2 + q_{xy}xy + q_{xz}xz + q_{yz}yz + q_x x + q_y y + q_z z + q. \tag{9}$$

Поскольку значение Q на поверхности равно нулю, для нахождения коэффициентов квадрики по девяти точкам ($P_1[x_1, y_1, z_1]$– $P_9[x_9, y_9, z_9]$) получаем систему линейных уравнений

$$\left\{ \begin{array}{c} Q(P_1) = 0 \\ Q(P_2) = 0 \\ \dots \\ Q(P_i) = 0 \\ \dots \\ Q(P_9) = 0 \end{array} \right\}. \tag{10}$$

Подставляя (9) в (10), имеем

$$\left\{ \begin{array}{c} q_{xx}x_1^2 + q_{yy}y_1^2 + q_{zz}z_1^2 + q_{xy}x_1y_1 + q_{xz}x_1z_1 + q_{yz}y_1z_1 + q_x x_1 + q_y y_1 + q_z z_1 + q = 0 \\ q_{xx}x_2^2 + q_{yy}y_2^2 + q_{zz}z_2^2 + q_{xy}x_2y_2 + q_{xz}x_2z_2 + q_{yz}y_2z_2 + q_x x_2 + q_y y_2 + q_z z_2 + q = 0 \\ \dots \\ q_{xx}x_i^2 + q_{yy}y_i^2 + q_{zz}z_i^2 + q_{xy}x_iy_i + q_{xz}x_iz_i + q_{yz}y_iz_i + q_x x_i + q_y y_i + q_z z_i + q = 0 \\ \dots \\ q_{xx}x_9^2 + q_{yy}y_9^2 + q_{zz}z_9^2 + q_{xy}x_9y_9 + q_{xz}x_9z_9 + q_{yz}y_9z_9 + q_x x_9 + q_y y_9 + q_z z_9 + q = 0 \end{array} \right\}. \tag{11}$$

Коэффициенты квадрики вычисляются с помощью метода Крамера:

$$\begin{bmatrix} x_1^2 & y_1^2 & z_1^2 & x_1y_1 & x_1z_1 & y_1z_1 & x_1 & y_1 & z_1 \\ x_2^2 & y_2^2 & z_2^2 & x_2y_2 & x_2z_2 & y_2z_2 & x_2 & y_2 & z_2 \\ x_3^2 & y_3^2 & z_3^2 & x_3y_3 & x_3z_3 & y_3z_3 & x_3 & y_3 & z_3 \\ x_4^2 & y_4^2 & z_4^2 & x_4y_4 & x_4z_4 & y_4z_4 & x_4 & y_4 & z_4 \\ x_5^2 & y_5^2 & z_5^2 & x_5y_5 & x_5z_5 & y_5z_5 & x_5 & y_5 & z_5 \\ x_6^2 & y_6^2 & z_6^2 & x_6y_6 & x_6z_6 & y_6z_6 & x_6 & y_6 & z_6 \\ x_7^2 & y_7^2 & z_7^2 & x_7y_7 & x_7z_7 & y_7z_7 & x_7 & y_7 & z_7 \\ x_8^2 & y_8^2 & z_8^2 & x_8y_8 & x_8z_8 & y_8z_8 & x_8 & y_8 & z_8 \\ x_9^2 & y_9^2 & z_9^2 & x_9y_9 & x_9z_9 & y_9z_9 & x_9 & y_9 & z_9 \end{bmatrix} \begin{bmatrix} q_{xx} \\ q_{yy} \\ q_{zz} \\ q_{xy} \\ q_{xz} \\ q_{yz} \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} K \\ K \\ K \\ K \\ K \\ K \\ K \\ K \\ K \end{bmatrix}. \tag{12}$$

Решая системы уравнений (12) со свободным членом q (положим его равным K) при заданных $P_1[x_1, y_1, z_1]$–$P_9[x_9, y_9, z_9]$, находим девять искомых коэффициентов: q_{xx}, q_{yy}, q_{zz}, q_{xy}, q_{xz}, q_{yz}, q_x, q_y, q_z. Далее коэффициенты этой квадрики будут использоваться при преобразовании в функциональную модель.

Функционально базируемая модель при растеризации задаётся в кубе с центром (0, 0, 0) и координатами от -1 до 1 по x, y, z в системе координат объекта. При обходе восьмеричного дерева

$$\begin{gathered} A' = A/4; \quad B' = B/4; \quad C' = C/4; \quad D' = D/4; \quad E' = E/4; \quad F' = F/4; \\ G' = G/2 + iA/2 + jD/4 + kE/4; \quad H' = H/2 + iD/4 + jB/2 + kF/4; \\ I' = I/2 + iE/4 + jF/4 + kC/2; \quad K' = K/2 + iG/4 + jH/4 + kI/4; \\ K'' = K'/2 + iG'/2 + jH'/2 + kI'/2 \end{gathered} \tag{13}$$

делим коэффициенты уравнения A, B, C, D, E, F на 4, координаты x, y, z — на 2 (уменьшаем размеры куба) и после этого делаем сдвиг на вектор (±0,5; ±0,5; ±0,5), т. е. в один из восьми подкубов (13), где коэффициенты без штриха берутся из предыдущего шага рекурсии.

Прежде чем преобразовывать полигональные сегментированные модели в функциональные, необходимо найти все точки полигональной поверхности. Далее отмечаются все ветви и листья дерева, где имело место пересечение модели с подкубами разных уровней восьмеричного дерева деления пространства объекта. Это сделать просто, поскольку известны все пересечённые листья дерева:

$$N_x = A'_{14}/\sqrt{A'^2_{14} + A'^2_{24} + A'^2_{34}}, \tag{14}$$

$$N_y = A'_{24}/\sqrt{A'^2_{14} + A'^2_{24} + A'^2_{34}}, \tag{15}$$

$$N_z = A'_{34}/\sqrt{A'^2_{14} + A'^2_{24} + A'^2_{34}}. \tag{16}$$

Рис. 4. Функционально базируемая модель

Суть преобразования заключается в следующем. Если с помощью рекурсивного деления объектного пространства можно визуализировать функционально заданные объекты (вычислять точки поверхности, нормали в этих точках, освещённость и т. д.), то можно решить и обратную задачу: по заданным точкам и нормалям (14)–(16) найти функции, описывающие данный объект. Для этого необходимо решить систему уравнений (13), т. е. вычислить коэффициенты уравнений самого низкого уровня деления пространства объекта (листьев восьмеричного дерева), сделать обратный обход восьмеричного дерева деления пространства объекта, вычислить коэффициенты функций и минимизировать эти функции. Решаются системы линейных уравнений на каждом уровне рекурсии, определяются одинаковые коэффициенты уравнений (с учётом установленного порога точности приближения) и минимизируется количество функций на каждом уровне. Только после обработки всех уравнений уровня происходит переход на следующий верхний уровень, и так процесс повторяется до самого корня восьмеричного дерева. В результате получается необходимый минимум функций, представляющих данный объект в формате описания функционально заданных объектов на базе квадрик с аналитическими функциями возмущения. Далее остаётся только объединить простые объекты в сложный с помощью теоретико-множественной операции [15] (рис. 4).

Анализ метода и результаты работы. Исходная модель (см. рис. 1) состояла из 136306 треугольников и 68418 вершин и была преобразована в 83 функции (см. рис. 4). При описании поверхности, имеющей $n \times n$ вершин, её представление на базе полигонов требует $3n^2$ вещественных чисел, необходимых для хранения вершин, и $6(n-1)^2$ целых чисел, необходимых для описания треугольников, в то время как для одной функции достаточно десяти коэффициентов в общем случае. Рассматриваемая модель состоит в основном из эллипсоидов, а они задаются ещё меньшим количеством коэффициентов. Коэффициент сжатия составляет более 200 единиц. Проведённые экспериментальные исследования показали, что коэффициенты сжатия в зависимости от тестовых моделей различного уровня детализации варьировались от 10 до 100 и выше, время конверсии составляло от нескольких миллисекунд до 1 секунды. Метод был протестирован для нескольких марок автомобилей, трёх типов самолётов, многих деталей и нескольких сборочных единиц полигональных моделей. С помощью данного метода возможно преобразование полигональных моделей в функциональные объекты без потерь. Более того, для гладких криволинейных поверхностей качество изображения улучшится. При необходимости можно конвертировать модель обратно в формат полигонального описания, как это было в [17], и визуализировать стандартным образом, принятым для геометрических акселераторов, или в функционально заданном виде [18]. Тестирование производилось на процессоре Intel Core2 CPU E8400 3.0 GHz. В известных подходах конверсии полигональных моделей в неявные поверхности применяются итерационные методы, которые являются приближёнными и медленными. В предлагаемом методе исходная полигональная сетка сложного объекта преобразуется в функционально базируемую модель посредством строгого математического вычисления без итераций.

Для определения отклонений (вершин и нормалей) разработан анализатор критериев отклонений. Анализ степени приближения полученной функционально заданной поверхности к исходной сетке треугольников осуществляется по двум метрикам: отклонение вершин треугольной сетки от поверхности функционального объекта и отклонение нормалей в вершинах треугольной сетки от значений нормалей функциональной поверхности. Для этого сначала вычисляются данные буфера глубины полигональной и функциональной моделей в виде двумерного массива. Каждый элемент такого массива представляет собой расстояние от камеры до точки на поверхности модели. Затем сравниваются все точки буферов для нахождения средней разницы и вычисляется среднее отклонение для соответствующей модели. Если полигональная и функциональная модели полностью совпадают, то отклонение будет равно нулю. Эти результаты определялись в пространственном кубе с разрешением $1{,}0 \times 1{,}0 \times 1{,}0$, и на их основании установлено, что сжатие происходит без потерь.

Заключение. В представленной работе предлагается эффективный метод сжатия геометрических (полигональных) данных без потерь, суть которого сводится к преобразованию объектов в функциональное описание на основе функций возмущения. Преобразование происходит путём строгого математического вычисления, а не итерационно или другими методами приближения, которые приводят к частичной потере информации. Предложенные способ задания трёхмерных объектов и метод сжатия геометрических данных имеют преимущества перед известными подходами. К их основным достоинствам следует отнести простоту преобразования полигональных объектов в функциональное описание с быстрым поиском описывающих функций и существенное уменьшение количества поверхностей для задания криволинейных объектов.

СПИСОК ЛИТЕРАТУРЫ

1. **Rossignac J., Szymczak A.** Wrap&zip: Linear decoding of planar triangle graphs // Computat. Geometry: Theory and Appl. 1999. **14**, N 1–3. P. 119–135.
2. **Rossignac J.** Edgebreaker: Connectivity compression for triangle meshes // IEEE Trans. Vis. Comput. Graph. 1999. **5**, Is. 1. P. 47–61. DOI: 10.1109/2945.764870.
3. **Touma C., Gotsman C.** Triangle mesh compression // Proc. of the Conf. on Graphics Interface'98. Vancouver, Canada, 18–20 June, 1998. P. 26–34. DOI: 10.20380/GI1998.04.
4. **Deering M.** Geometry compression // Proc. of the Conf. SIGGRAPH'95. Los Angeles, USA, 6–11 Aug., 1995. P. 13–20. DOI: 10.1145/218380.218391.
5. **Isenburg M., Snoeyink J.** Mesh collapse compression // Proc. of the 15th Annual Symp. Computational Geometry (SCG'99). Miami Beach, USA, 13–16 June, 1999. P. 419–420. DOI: 10.1145/304893.305000.
6. **Pajarola R., Rossignac J., Szymczak A.** Implant sprays: Compression of progressive tetrahedral mesh connectivity // Proc. of the IEEE Conf. on Visualization'99. San Francisco, USA, 24–29 Oct., 1999. P. 299–306.
7. **Pajarola R., Rossignac J.** Compressed progressive meshes // IEEE Trans. Vis. Comput. Graph. 2000. **6**, Is. 1. P. 79–93. DOI: 10.1109/2945.841122.
8. **Pajarola R., Rossignac J.** Squeeze: Fast and progressive decompression of triangle meshes // Proc. of the Computer Graphics Intern. Conf. (CGI'2000). Geneva, Switzerland, 19–24 June, 2000. P. 173–182.
9. **Derzapf E., Guthe M.** Dependency-free parallel progressive meshes // Comput. Graph. Forum. 2012. **31**, N 8. P. 2288–2302.
10. **Caillaud F., Vidal V., Dupont F., Lavoué G.** Progressive compression of arbitrary textured meshes // Comput. Graph. Forum. 2016. **35**, N 7. P. 475–484.
11. **Yngve G., Turk G.** Robust creation of implicit surfaces from polygonal meshes // IEEE Trans. Vis. Comput. Graph. 2002. **8**, N 4. P. 346–359.

12. **Shen C., O'Brien J. F., Shewchuk J. R.** Interpolating and approximating implicit surfaces from polygon soup // Proc. of the 31th Intern. Conf. ACM SIGGRAPH. Los Angeles, USA, 8–12 Aug., 2004. P. 896–904.
13. **Ohtake Y., Belyaev A., Alexa M., Seidel H.-P.** Multi-level partition of unity implicits // Proc. ACM Trans. Graph. 2003. **22**, N 3. P. 463–470.
14. **Вяткин С. И.** Моделирование сложных поверхностей с применением функций возмущения // Автометрия. 2007. **43**, № 3. С. 40–47.
15. **Вяткин С. И.** Преобразования функционально заданных форм // Программные системы и вычислительные методы. 2014. **9**, № 4. С. 484–499.
16. **Bloomenthal J., Bajaj C., Blinn J. et al.** Introduction to Implicit Surfaces. San Francisco, USA.: Morgan Kaufmann Publishers, 1997. 332 p.
17. **Vyatkin S. I.** Polygonization method for functionally defined objects // Intern. Journ. Automat., Control and Intell. Syst. 2015. **1**, N 1. P. 1–8.
18. **Вяткин С. И.** Метод рекурсивного поиска элементов изображения функционально заданных поверхностей // Автометрия. 2017. **53**, № 3. С. 53–57.

Поступила в редакцию 12 марта 2018 г.