РОССИЙСКАЯ АКАДЕМИЯ НАУК

СИБИРСКОЕ ОТДЕЛЕНИЕ

АВТОМЕТРИЯ

2002, том 38, № 4

УДК 681.3.06

С. И. Вяткин, Б. С. Долговесов

(Новосибирск)

СИНТЕЗ ПОВЕРХНОСТЕЙ СВЕРТКИ С РЕКУРСИВНЫМ ДЕЛЕНИЕМ ОБЪЕКТНОГО ПРОСТРАНСТВА

Предлагается эффективный алгоритм быстрой визуализации поверхностей свертки. Рассматриваются поверхности свертки, ограниченные оболочками свободных форм. Предложен рекурсивный алгоритм деления объектного пространства. Приводятся примеры изображений, полученных при моделировании алгоритма.

Введение. Поверхности свертки [1–3] – это интегральное представление неявно заданных поверхностей, известных в компьютерной графике как капельные модели [4, 5] (рис. 1), метасферы [6], мягкие объекты [7]. Данные поверхности сочетают в себе гибкость капельных моделей и компактность скелетных моделей [8] и представляют собой гораздо более мощное средство геометрического моделирования, чем традиционные модели неявных поверхностей [9]. Несмотря на различные названия все эти модели описывают фактически один и тот же объект, а именно изоповерхность S уровня T в скалярном поле $f(\mathbf{p})$:

$$S = \{\mathbf{p} \in R^3 \mid f(\mathbf{p}) - T = 0\}. \qquad (1)$$

Поверхность свертки – это неявная поверхность S с базовой функцией $f(\mathbf{p})$, полученная с помощью свертки:

$$f(\mathbf{p}) = g(\mathbf{p}) * h(\mathbf{p}) = \int_{R^3} g(\mathbf{r}) h(\mathbf{p} - \mathbf{r}) d\mathbf{r}, \qquad (2)$$

где $\mathbf{r}$ – расстояние действия поля.

Геометрическая функция $g(\mathbf{p})$ определяет форму объекта и его положение в трехмерном пространстве. Ядро свертки $h(\mathbf{p})$ определяет распределение потенциала в каждой точке объекта. Свертка двух функций – это скалярная функция $f(\mathbf{p})$, которая является поверхностью свертки. Поверхности свертки – это обычные изоповерхности, с той разницей, что скалярное поле $f(\mathbf{p})$ для поверхности свертки за-

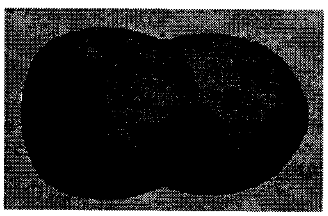
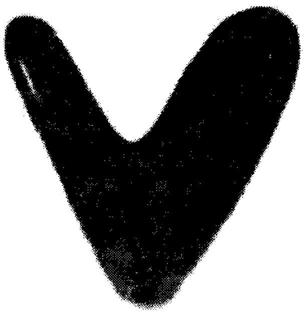


Рис. 1. Капельная модель

Рис. 2. Поверхность свертки на базе скелетона из двух примитивов



дается с помощью различных геометрических объектов.

Скелетоны. Если капельные модели построены из точечных компонентов-капель, то поверхности свертки базируются на более разнообразном выборе: точки, отрезки, дуги, треугольники, плоскости и в принципе любые геометрические примитивы. Скелетон – это совокупность геометрических примитивов, которые вместе определяют общие очертания объекта. В терминах поверхностей свертки скелетон задается суммой геометрических функций $g(\mathbf{p})$. Оператор свертки обладает свойством линейности, что позволяет считать поле от всего скелетона по частям, а также строить сложные примитивы из простых компонентов. Например, из дуг можно легко строить кольца и спирали, из треугольников – многоугольники и т. д. Итак, скелетон есть сумма (объединение) геометрических функций $g = \sum_{i=1}^{N} g_i$ и $f = h * \sum_{i=1}^{N} g_i$, которая также равна $\sum_{i=1}^{N} h * g_i$, потому что свертка – это линейный оператор.

Пример свертки с отрезками показан на рис. 2. В работе [1] описаны функция поля и два параметра: собственный радиус R и параметр округлости B. Первый параметр задает размер сферы – изоповерхности от отдельно взятого точечного примитива, т. е. это расстояние между геометрическим примитивом и неявно заданной поверхностью. Второй параметр определяет степень округлости объекта относительно скелетона.

Параметры R и B, введенные для точечной капельной модели, можно легко перенести на более сложные примитивы, например, отрезки, дуги, треугольники и плоскости. На рис. 3 и 4 показана поверхность с одним и тем же скелетоном и разными значениями R и B. Целесообразно включать собственный радиус и параметр округлости в набор параметров, задающих оптические свойства поверхности, такие, как цвет, коэффициенты преломления, отражения и т. д. Таким образом, можно расширить понятие материала, который помимо чисто фотометрических характеристик теперь еще несет и геометрическую информацию об объекте.

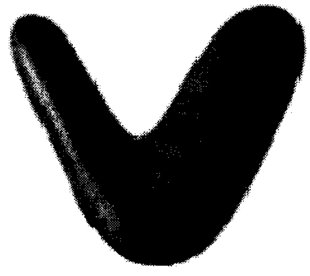


Рис. 3. Поверхность свертки

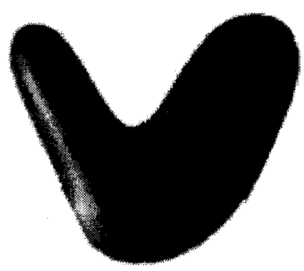


Рис. 4. Поверхность свертки с увеличенными параметрами R и B

Для генерации полутоновых изображений используется алгоритм трассировки лучей. Чтобы найти точки поверхности свертки, необходимо в каждой точке трехмерного пространства шаг за шагом проверять знак функции, когда точка, принадлежащая поверхности и лежащая на луче, найдена. Если поверхность непрозрачная, процесс вычислений для этого луча прекращается. Чтобы ускорить вычисления, в работе [9] был предложен быстрый алгоритм трассировки лучей с применением ограничивающих объемов. Более того, чтобы еще уменьшить количество вычислений, были предложены два вида кластеризации объектов (Volatile and Permanent Clusters).

В данной работе предлагается более эффективный алгоритм быстрой визуализации поверхностей свертки с применением ограничивающих оболочек свободных форм и рекурсивного деления объектного пространства.

Ядра свертки. Рассмотрим известные ядра свертки, отметим их достоинства и недостатки, выберем и обоснуем наиболее оптимальное ядро для моделирования поверхности свертки.

1. Функция Гаусса [1, 4, 8]:

$$h(r)=\exp(-a^2r^2)^2,\qquad r>0. \tag{3}$$

Для таких примитивов, как точки, линии и плоскости, поведение данной функции корректно, но для дуг и треугольников не годится.

2. Обратноквадратичная функция [7]:

$$h(r)=1/r^2,\qquad r>0. \tag{4}$$

Данная функция ведет себя корректно с разными примитивами, однако для плоскостей не сходится. Решение для дуг выражается через эллиптический интеграл, который очень сложен для вычисления.

3. Полином четвертой степени W-формы [9]:

$$h(r)=(1-r^2)^2_,\qquad r<1. \tag{5}$$

С такими примитивами, как точки, линии, плоскости и дуги, данная функция ведет себя корректно, но для треугольников не годится.

4. Функция Коши [9]:

$$h(r)=1/(1+s^2r^2)^2_,\qquad r>0, \tag{6}$$

где r – радиус, а s – коэффициент, с помощью которого контролируется ширина ядра. Функция Коши ведет себя корректно с полным набором геометрических примитивов. С помощью такого ядра можно вычислить аналитически интеграл свертки (2).

Методы визуализации. Рассмотрим известные методы визуализации, отметим их достоинства и недостатки, обоснуем наш выбор в пользу многоуровневого рекурсивного деления объектного пространства, предложенного в данной работе.

Маршировка по лучу. Это метод «грубой силы» пошаговых вычислений вдоль луча, функция $f(t)$ вычисляется на каждом шаге. Первое изменение знака функции $f(t)$ сигнализирует о том, что найдена поверхность $F(r) = f(t) = 0$. Главный недостаток этого метода заключается в том, что он очень медленный и не гарантирует обнаружение поверхности.

Метод LG-поверхности. Разработан метод обнаружения поверхности с применением L- и G-параметров, которые являются константами Липшица для функции f и производной df/dt вдоль луча. Для неалгебраической функции f вычисления L- и G-параметров становятся сложными, даже если они представлены в символической форме.

Трассировка луча с анализом интервала. Данный метод является модификацией метода LG-поверхности.

Быстрая трассировка луча [9]. Данный метод лишен недостатков перечисленных выше методов, однако поиск лучей, пересекающих поверхности, сложен и не достаточно эффективен, поскольку способы кластеризации этого метода не решают данную проблему полностью.

Синтез поверхностей свертки с рекурсивным делением объектного пространства. Этот алгоритм предлагается в данной работе, он лишен всех недостатков вышеперечисленных методов и вобрал в себя все их положительные качества. Кроме того, данный алгоритм максимально адаптирован к аппаратной реализации.

Ограничивающие оболочки. В работе [9] для уменьшения количества вычислений предложены ограничивающие объемы нескольких видов: для точки – это сфера; для отрезка – один цилиндр и две сферы; для дуги – часть тора и две сферы; для треугольника – три цилиндра, три сферы и одна призма; для плоскости – одна бесконечная пластина. В данной работе предлагается ввести ограничивающие оболочки свободных форм относительно не только всех перечисленных примитивов, но и самой поверхности свертки (см. рис. 4).

Квадрика является основой для построения оболочек свободных форм и определяется с помощью вещественной непрерывной описывающей функции трех переменных (x_1, x_2, x_3). Рассматриваются квадрики как замкнутые подмножества эвклидова пространства E_n, определяемые описывающей функцией $F(X) \geq 0$, где F – непрерывная вещественная функция; $X = (x_1, x_2, x_3)$ – точка в E_n, задаваемая координатными переменными. Здесь $F(X) > 0$ задает точки внутри квадрики; $F(X) = 0$ – точки на границе; $F(X) < 0$ – точки, лежащие снаружи и не принадлежащие квадрике. Предлагается описывать сложные геометрические объекты, задавая функцию отклонения (второго порядка) от базовой квадрики в виде

$$F(x,y,z) = A_{11}x^2 + A_{22}y^2 + A_{33}z^2 + A_{12}xy + A_{13}xz + A_{23}yz +$$

$$+ A_{14}x + A_{24}y + A_{34}z + A_{44} \geq 0. \tag{7}$$

На базе квадрики строятся свободные формы [10]. Свободная форма есть композиция базовой квадрики и возмущения $F'(x,y,z) = F(x,y,z) + R(x,y,z)$, где функция возмущения $R(x,y,z)$ находится следующим образом:

$$R(x,y,z) = \begin{cases} Q^2(x,y,z) & \text{при} \quad Q(x,y,z) > 0; \\ 0 & \text{при} \quad Q(x,y,z) \leq 0, \end{cases} \tag{8}$$

$Q(x, y, z)$ – возмущающая квадрика.

В качестве Q также может быть возмущенная квадрика (свободная форма). Другими словами, композиция базовой квадрики и функции отклонения являются новой функцией возмущения, т. е. производной для другой базовой квадрики. Вследствие того что $\max[Q+R] \le \max[Q]+\max[R]$, для оценки максимума Q на некотором интервале необходимо вычислить максимум функции возмущения на этом же интервале.

Алгоритм растрирования. Благодаря ограничивающим оболочкам можно сканировать не все трехмерное пространство, а лишь ту его часть, которая ими ограничена, и чем плотнее оболочки будут подогнаны к поверхностям, тем меньший объем надо просканировать, пока не будет выделена поверхность свертки. В данной работе использовался алгоритм многоуровневого рекурсивного деления объектного пространства [11, 12], осуществляющий эффективный поиск лучей, пересекающих оболочки, и точек пересечения луча с поверхностью оболочки. Предлагаемый алгоритм совмещает в себе такие свойства, как универсальность, надежность и эффективность. Он удовлетворяет следующим двум необходимым условиям.

Условие 1: а) функция $f(t)$ является C^1-непрерывной для всех t; б) функция $f(t)$ и ее производная $df(t)/dt$ имеют ненулевые значения внутри неперекрывающихся интервалов $[t_{1i}, t_{2i}]$, где $i=1,\dots,k$ – номера интервалов.

Условие 2 гарантирует, что каждый объект, заданный функцией f, может быть заключен в ограничивающую оболочку.

Алгоритм включает два основных этапа: первый – это поиск пересечений луча с оболочкой, второй – поиск точек поверхности свертки. На первом шаге рекурсии исходная пирамида видимости разбивается на четыре меньшие подпирамиды в экранной плоскости. На этапе деления пространства по четверичному дереву рекурсивное преобразование коэффициентов квадрики-оболочки выглядит так:

$$\begin{gathered}
A'_{11}=A_{11}/4; \quad A'_{22}=A_{22}/4; \quad A'_{33}=A_{33};\\
A'_{12}=A_{12}/4; \quad A'_{13}=A_{13}/2; \quad A'_{23}=A_{23}/2;\\
A'_{14}=A_{14}/2+iA_{11}/2+jA_{12}/4; \quad A'_{24}=A_{24}/2+iA_{12}/4+jA_{22}/2;\\
A'_{34}=A_{34}+iA_{13}/2+jA_{23}/2; \quad A'_{44}=A_{44}+iA_{14}/4+jA_{24}/4;\\
A''_{44}=A'_{44}+iA'_{14}/2+jA'_{24}/2,
\end{gathered} \tag{9}$$

где коэффициенты без штриха берутся с предыдущего шага рекурсии; переменные $i, j=\pm 1$ определяются в зависимости от направления погружения в рекурсию (i погружается по оси x, j – по оси y). Полученные коэффициенты используются в тесте на пересечение (пересечение луча с оболочкой). Для каждой новой подпирамиды выполняется тест на пересечение. Если в уравнении квадрики-оболочки $Q(x, y, z)=0$ значения переменных x, y, z меняются в пределах отрезка [–1, 1], то

$$\begin{gathered}
\max[|Q(x,y,z)-A_{44}|] \le \max F =\\
=|A_{11}|+|A_{22}|+|A_{33}|+|A_{12}|+|A_{13}|+|A_{23}|+|A_{14}|+|A_{24}|+|A_{34}|.
\end{gathered} \tag{10}$$

Рис. 5. Выделенная поверхность – берцовая кость человека. Ограничивающая оболочка показана светлым фоном

Теперь заметим, что если $|A_{44}| \le \max[|Q(x,y,z) - A_{44}|] \le \max F$, то, возможно, существует точка $M_0 = (x_0, y_0, z_0)(-1 < x_0, y_0, z_0 < 1)$ такая, что $Q(x_0, y_0, z_0) = 0$.

мид, которые лежат внутри квадрики-оболочки целиком или частично, а внешние подпирамиды заведомо исключаются из обработки. Тест на пересечение подпирамид с оболочками свободных форм несколько отличается. Для базовой квадрики-оболочки тест на пересечение выглядит следующим образом. Если

$$((A_{44} + R) < 0) \;\&\; (|A_{11}| + |A_{22}| + |A_{33}| + |A_{12}| + |A_{13}| + |A_{23}| + |A_{14}| + |A_{24}| + |A_{34}| < -(A_{44} + R)), \tag{11}$$

тогда подпирамида находится снаружи. Здесь R – максимум функции возмущения на текущем интервале, а A_{ij} – коэффициенты квадратичной функции. Для функции возмущения проводится тест. Если

$$(|A_{11}| + |A_{22}| + |A_{33}| + |A_{12}| + |A_{13}| + |A_{23}| + |A_{14}| + |A_{24}| + |A_{34}| < |A_{44}|), \tag{12}$$

тогда подпирамида находится снаружи области определения возмущения, где A_{ij} – коэффициенты квадратичной функции возмущения. Дополнительно вычисляется значение R, которое служит добавкой к базовой функции. Если пересечение имеет место, то подпирамида подвергается следующему уровню рекурсии. Подпирамиды, не пересекающиеся с оболочкой, в дальнейшем не обрабатываются. Это исключает из рассмотрения квадратные области экрана, на которые данная подпирамида (следовательно, и поверхность оболочки) не отображается (рис. 5). Деление пирамиды видимости ведется до тех пор, пока не достигается максимально установленный уровень рекурсии. Преимущество этой методики в том, что она позволяет на ранней стадии «отбросить» большие части пустого пространства. В процессе поиска точек, содержащих в себе участки поверхности оболочки (рис. 6), осуществляется обход пирамидального пространства по четверичному дереву, листья которого являются корнями двоичных деревьев. Алгоритм многоуровневого рекур-

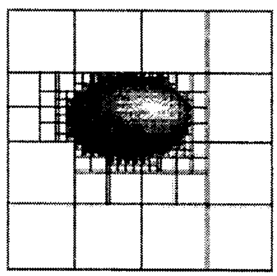


Рис. 6. Проекции подпирамид разных уровней пирамиды видимости на экранную плоскость

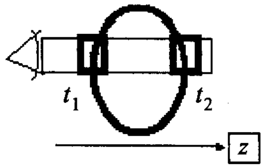


Рис. 7. Пересечения луча с поверхностью ограничивающей оболочки

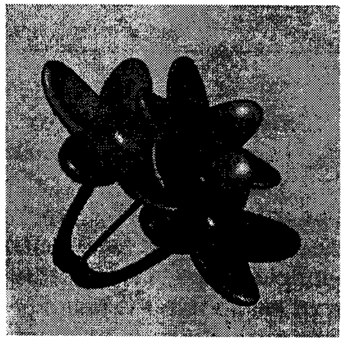


Рис. 8. Поверхность свертки на базе скелетонов нескольких видов

сивного деления объектного пространства позволяет эффективно и быстро определить принадлежность лучей разных уровней (пирамид) поверхностям оболочек и отбраковать области пространства вне оболочек.

В соответствии с теоремой Вейерштрасса любая непрерывная функция f может быть аппроксимирована на замкнутом интервале с помощью полинома p так точно, как это необходимо. В нашем случае интервал $[t_1, t_2]$(рис. 7) вычисляется после нахождения пересечений луча с оболочкой. Далее вычисляется средняя точка t_{mid} этого интервала. Все уравнения для выделения поверхности свертки необходимо решать для подинтервалов $[t_1, t_{\text{mid}}]$ и $[t_{\text{mid}}, t_2]$, чтобы получить кусочно-непрерывное представление функции f на интервале $[t_1, t_2]$. Это обеспечит низкий порядок функции, C^1-непрерывность и вычислительную эффективность. Результирующие интервалы t_1 и t_2 определяют геометрическое положение вдоль луча, где поле ненулевое. Затем для каждого интервала t_i соответствующая функция поля f_i интерполируется полиномом p_i. На последнем этапе вычислений все интервалы сортируются вдоль луча в последовательности t_1, t_2, t_3 и т. д. Таким образом, для каждой функции f_i вычисляются пересечения луча с соответствующей оболочкой. Уравнения изоповерхности решаются для корней t на всех интервалах. В общем случае число корней в каждом интервале зависит от степени всех интерполяционных полиномов $p_i(t)$, определенных на этом интервале. Эти корни могут оказаться вне интервала. Поэтому в алгоритме есть проверка на принадлежность всех корней каждому интервалу. Пример поверхности свертки, иллюстрирующий результат работы данного алгоритма, показан на рис. 8.

Заключение. В работе [9] для поиска лучей, пересекающих оболочки, применяется кластеризация двух видов, однако их эффективность значительно уступает предложенному в данной работе алгоритму. Это особенно заметно для равномерно распределенных по пространству небольших объектов, которые плохо объединяются в кластеры. К основным положительным особенностям предлагаемого алгоритма следует отнести:

1. Эффективность метода растрирования, сочетающего простоту вычисления с быстрым поиском и отбраковкой областей, не занятых объектами сцены.

2. Уменьшение количества вычислений за счет применения оболочек свободных форм.

3. Адаптация к аппаратной реализации.

Время вычислений при введении ограничивающих оболочек для разных сцен в среднем уменьшается на порядок. Еще на порядок уменьшается время вычислений при рекурсивном делении объектного пространства, поскольку обрабатываются только те лучи, которые пересекают поверхности оболочек. В методе [9] лучи, проходящие мимо оболочек, отбраковываются только после соответствующих вычислений, в то время как в предлагаемом алгоритме области, не занятые объектами сцены, отбраковываются большими частями на более ранних этапах. При аппаратной реализации данного алгоритма все вычисления легко распараллеливаются и могут быть организованы в виде вычислительного конвейера.

СПИСОК ЛИТЕРАТУРЫ

1. **Bloomenthal J., Shoemake K.** Convolution surfaces // Computer Graphics. 1991. **25**, N 4. P. 251.
2. **Bloomenthal J.** Bulge elimination in convolution surfaces // Computer Graphics Forum. 1997. **16**, N 1. P. 1.
3. **McCormack J., Sherstyuk A.** Creating and rendering convolution surfaces // Computer Graphics Forum. 1998. **17**, N 2. P. 113.
4. **Blinn J. F.** A generation of algebraic surface drawing // ACM Transactions on Graphics. 1982. **1(3)**. P. 235.
5. **Muraki S.** Volumetric shape description of range data using "blobby model" // Computer Graphics. 1991. **25**, N 4. P. 227.
6. **Nishimura H., Hirai M., Kawai T. et al.** Object modelling by distribution function and a method of image generation // The Transactions of the Institute of Electronics and Communication Engineers of Japan. 1985. **J68-D(4)**. P. 718.
7. **Wyvill G., McPheeters C., Wyvill B.** Data structure for soft objects // The Visual Computer. 1986. **2(4)**. P. 227.
8. **Brandt J., Algazi R. V.** Continuous skeleton computation by Voronoi diagram // Computer Vision, Graphics, and Image Processing: Image Understanding. 1992. **55**, N 3. P. 329.
9. **Sherstuyk A.** Fast ray tracing of implicit surfaces // Computer Graphics Forum. 1999. **18**, N 2. P. 145.
10. **Вяткин С. И., Долговесов Б. С., Есин А. В. и др.** Геометрическое моделирование и визуализация функционально заданных объектов // Автометрия. 1999. № 6. С. 84.
11. **Vyatkin S. I., Dolgovesov B. S, Yesin A. V. et al.** Voxel-Volumes volume-oriented visualization system: Intern. Conf. on Shape Modelling and Applications (March 1–4, 1999, Aizu-Wakamatsu, Japan) // IEEE Comput. Soc. Los Alamitos, California, 1999. P. 234.
12. **Vyatkin S. I., Dolgovesov B. S., Guimaoutdinov O. Y.** Synthesis of virtual environment using perturbation functions: World Multiconference on Systemics, Cybernetics and Informatics Proceedings (Orlando, Florida, USA, July 22–25, 2001) // Emergent Computing and Virtual Engineering. 2001. V. III. P. 350.

Институт автоматики и электрометрии СО РАН,
E-mail: sivser@mail.ru

Поступила в редакцию
31 января 2002 г.