

# Манипулирование Свободными Формами с Функциями Возмущения

С.И. Вяткин, Б.С. Долговесов, М.А. Гордилов  
Институт автоматизации и электрометрии, СО РАН,  
Новосибирск, Россия

[sivser@mail.ru](mailto:sivser@mail.ru), [bsd@iae.nsk.su](mailto:bsd@iae.nsk.su), [gorodilovm@gmail.com](mailto:gorodilovm@gmail.com)

## Аннотация

В данной статье рассматриваются свободные формы, синтезированные с помощью функций возмущения. Свободные формы имеют высокую степень гладкости и компактное описание. Для создания формы предлагается набор алгоритмов и программного обеспечения на основе функций возмущения. Интерактивная модификация модели и быстрая визуализация позволяют обеспечить манипулирование объектом с необходимой степенью детализации, для синтеза изображений реалистичного качества.

*Ключевые слова:* Функции возмущения, Геометрические объекты и операции, Интерактивное моделирование.

## 1. ВВЕДЕНИЕ

При формировании трехмерных сцен наиболее часто используется полигональное задание моделей объектов, которое на современном уровне развития компьютерной графики имеет ряд ограничений. Каркасные модели трехмерных объектов являются приближенными. Для достижения фотореализма необходимо использование свыше миллиона полигонов в сцене, причем наблюдается тенденция к дальнейшему увеличению детализации. Уже сегодня во многих приложениях количество треугольников сцены сопоставимо или превышает число пикселей, занимаемых на экране, что нивелирует преимущества полигонального подхода. При возрастании сложности сцены эффективность полигонального метода экспоненциально падает. Структура полигональных сеток линейна и они не обеспечивают поддержки многомасштабности, поэтому работа с большими сетками затруднена и требует вычислительно сложных методов упрощения. Полигональная модель принципиально не позволяет получить многих визуальных эффектов, необходимых для реалистичного отображения сцены. С помощью скелетной анимации нельзя сделать качественную анимацию гибких материалов, а также выполнить сложный морфинг объектов.

От этих недостатков можно избавиться, применяя аналитическое задание объектов и растеризацию их при помощи алгоритмов трассировки лучей. Аналитическое задание геометрических объектов не требует большого объема памяти. Существуют работы по визуализации функционально заданных поверхностей, таких как F-гер [1], поверхностей свертки [2], неявно заданных поверхностей, известных в компьютерной графике как капельные модели [3], метасфер [4], мягких объектов [5] и т. д. Однако их применение ограничено довольно узким классом моделируемых поверхностей и медленной визуализацией. Используемые алгоритмы сложно оптимизировать для визуализации в реальном времени, что также накладывает ограничения на их

практическое применение. Один из главных недостатков известных методов визуализации - сложность вычисления точек поверхности. Так, метод маршировки по лучу не гарантирует обнаружение поверхности, кроме того, он медленный [6]. В работе [7] описан метод вычисления пересечения луча с поверхностью, заданной в неявном виде, однако вычисления  $L$ - и  $G$ - параметров очень сложные. В методе сферы-трэйсинга нахождение наибольшего радиуса, чтобы ни одна точка объема не лежала внутри сферы - нетривиальная задача [8]. Такой метод можно эффективно применять только тогда, когда объекты статичны. То есть, когда форма и масштаб самих объектов не изменяются. В трассировке луча с анализом интервала для сложных функций требуется делать много вычислений, поскольку они необходимы индивидуально для каждого луча и для каждого интервала вдоль этого луча [9]. В быстрой трассировке луча поиск лучей, пересекающих поверхности сложен и не достаточно эффективен, поскольку способы кластеризации этого метода не решают данную проблему полностью [10]. В функциональных заданиях поверхностей вычисление определяющей функции в заданной точке часто оказывается весьма трудоемкой задачей, особенно в случае использования R-функций с квадратными корнями [1].

В работе [11] описан метод отслеживания лучей для визуализации поверхностей, заданных алгебраически полиномами высокой степени. Чтобы с помощью обычного полинома задать сложную поверхность необходимо увеличивать его степень. В случае с кривыми Безье гарантируется высокая точность нахождения пересечения с кривой, но не гарантируется то, насколько точно начальная функция будет приближена этой кривой. Ещё одним недостатком этого метода является то, что перевод объекта в другую систему координат – не простая задача. Поэтому создание динамических сцен становится проблематичным.

Целью данной работы является интерактивное моделирование и визуализация функционально заданных объектов на основе функций возмущения.

В последнее время компьютерная графика, связанная с интерактивным моделированием и редактированием трехмерных объектов развивается быстрыми темпами. Известны такие коммерческие системы как SoftImage, MAYA, 3DStudioMax и др. для интерактивного редактирования полигональных моделей. Существуют интерактивные системы на базе полиномов, неявных поверхностей [12], треугольных сеток [13], изображений [14], объемов [15], функционально заданных моделей [16, 17].

Конструирование объекта в нашем случае сводится к задаче деформации базовой поверхности нужным образом, а не к аппроксимации ее примитивами, данный процесс напоминает лепку модели из пластилина с применением геометрических операций, представленных ниже и деформации. Основное положительное отличие от известных интерактивных систем

на базе функционально заданных моделей это то, что нет необходимости в полигонизации функциональной модели перед деформацией.

## 2. ГЕОМЕТРИЧЕСКИЕ ОБЪЕКТЫ

Для описания сложных геометрических объектов используются функции отклонения (второго порядка) от базовой квадрики [18]. На базе квадрик строятся свободные формы. Свободная форма представляется композицией базовой квадрики и возмущений

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

где  $R_i(x, y, z)$  – возмущение,  $f_i$  – форм-фактор.

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

где  $Q_i(x, y, z)$  – возмущающая квадрика.

Для того чтобы поверхность была гладкой, степень должна быть больше двух (2). Это условие гарантирует непрерывность функции и её производной.

## 3. ГЕОМЕТРИЧЕСКИЕ ОПЕРАЦИИ

Геометрическая модель должна обеспечивать конструирование объектов и их композиций различной сложности. Для этого используется множество геометрических операций  $\Phi_j$ , определяемое математически следующим образом [1]:

$$\Phi_j: M^1 + M^2 + \dots + M^n \rightarrow M, \quad (3)$$

Для формирования моделей сложных объектов на базе функций возмущения используются такие операции, как проекции, офсеттинг, метаморфозис, кручение и заметание движущимся твердым телом [19], осуществляемые с применением булевых операций объединения и пересечения. Бинарная операция ( $n=2$ ) (3) объектов  $G_1$  и  $G_2$  означает операцию  $G_3 = \Phi_j(G_1, G_2)$  с определением:

$$f_3 = \psi(f_1(x, y, z), f_2(x, y, z)) \geq 0, \quad (4)$$

где  $\psi$  – непрерывная вещественная функция двух переменных. С помощью операции смещения (offsetting) можно создавать увеличенную или уменьшенную копию исходного объекта, то есть делать положительный или отрицательный офсеттинг соответственно. При метаморфозисе или морфинге осуществляется плавный переход одного объекта в другой. При функциональном задании объектов с применением функций возмущения можно осуществлять трехмерный морфинг негомеоморфных объектов (например, тора и сферы) и морфинг с ограничениями без разрыва поверхности и последующего “склеивания” последней. Два тела гомеоморфны, если из исходного тела путем взаимно однозначного непрерывного преобразования описывающей его функции можно получить второе. То есть для негомеоморфных объектов мы не можем простым приближением квадрик вычислить неразрывный морфинг. В процессе морфинга могут участвовать не два исходных объекта, а четыре. В этом случае схема интерполяции будет билинейной, а геометрическая операция – кватернарной ( $n=4$ ) (3). Кручение – это деформация тела, являющаяся частным

случаем биективного отображения. Биективное отображение служит для определения деформаций исходных объектов. Заметание – это проекция движущегося тела из  $4D(x, y, z, t)$  в  $3D(x, y, z)$  пространство. Эти операции не изменяют степень функций заданных моделей. Одним из примеров отношений может служить определение столкновений между объектами [20]. Бинарное отношение есть множество множества

$M^2 = M \times M$ . Оно может быть определено как:

$$Sj: M \times M \rightarrow I \quad (5)$$

С помощью особого теста на пересечение и бинарного поиска можно за постоянное число шагов (определяется заданной точностью) определить точку столкновения объектов, если таковое происходит. В целях расчета времени обнаружения столкновений тестировались объекты, различавшиеся как по степени сложности (форме), так и по виду столкновения (имеется в виду столкновение различными сторонами и частями объектов).

Более подробное описание трехмерных объектов на базе функций возмущения в монографии [21].

## 4. ВИЗУАЛИЗАЦИЯ

Главным на этом этапе является эффективное нахождение первого пересечения луча с поверхностью. Данная задача сходна с визуализацией объёмных томографических данных, где функция плотности задается в виде дискретных данных. А в нашем случае используется аналитически заданная функция, что позволяет более эффективно осуществлять поиск точек поверхности.

Для вычисления пересечения лучей с поверхностями трехмерных объектов предлагается метод, в котором отсутствуют вышеперечисленные недостатки, характерные для рассмотренных ранее известных методов аналитического задания поверхностей.

Для простоты понимания будем считать, что сцена находится в единичном трёхмерном кубе. Перспектива рассматриваться не будет, ввиду того, что она сводится к переходу в другую систему координат. Поэтому опустим начальные преобразования и уделим больше внимания основной части метода. Будем считать, что наблюдатель смотрит вдоль оси  $Z$ . Необходимо получить проекцию сцены на плоскость  $XY$ . Проекция должна представлять собой конечный набор значений. Через плоскость  $XY$  куба проходят лучи, и каждому лучу соответствует пиксел на изображении. Лучи ограничены передней и задней гранями куба. В процессе поиска точки пересечения луча и поверхности, каждый луч делится вдоль оси  $Z$ , образуя набор вокселей. Найдя ближайшую к наблюдателю точку для каждого луча, можно вычислить координату  $Z$ . Далее в каждом пикселе вычисляется нормаль. Имея все координаты и нормали в каждом пикселе, можно использовать модель локального освещения. В итоге получится изображение гладкого объекта с учётом освещения.

Уменьшение времени на визуализацию достигается за счёт эффективного использования вычислительных ресурсов графического акселератора с архитектурой CUDA от компании NVIDIA. Большое количество вычислительных процессоров позволяет параллельно проверять пересечение с объектом одновременно нескольких лучей. Следовательно, будет параллельно вычисляться достаточно большая часть куба [22].

## 5. ИНТЕРАКТИВНОЕ МОДЕЛИРОВАНИЕ

До недавнего времени удовлетворительные методы создания функциональных объектов отсутствовали. Объекты описывались вручную в текстовом виде. Эта операция чрезвычайно трудоемка. Поэтому задача создания инструментальной среды для редактирования объектов с применением функций возмущения стала очень актуальной. Были предложены, разработаны и реализованы структурные и алгоритмические решения по интерактивному объемно-ориентированному геометрическому моделированию функционально заданных объектов на основе аналитических функций возмущения, позволяющие пользователям легко и быстро создавать, моделировать и редактировать трехмерные объекты в интерактивном режиме. Для достижения поставленной задачи было сделано следующее: сформулированы требования к разрабатываемому программному обеспечению; разработан формат хранения данных в памяти; разработаны и реализованы алгоритмы работы с файлами и алгоритмы построения сцены из памяти. А также, разработан и реализован алгоритм добавления объекта в сцену; разработан и реализован алгоритм нахождения объектов в сцене; создан интерфейс к программе.

В результате была разработана интерактивная инструментальная среда, которая значительно упрощает процесс создания и редактирования функционально заданных объектов. Для визуализации был использован оптимизированный алгоритм многоуровневого отслеживания лучей, который осуществляет эффективный поиск точек поверхностей, участвующих в формировании изображения. Были разработаны и реализованы необходимые алгоритмы и C++ классы для интерактивного объемно-ориентированного геометрического моделирования: алгоритм рекурсивного многоуровневого отслеживания лучей (оптимизированный алгоритм рендеринга) сцен, содержащие функционально заданные объекты (включая OpenGL буферную совместимость - цвет/глубина); C++ классы функционально-заданных объектов; C++ классы рендеринга функционально заданных объектов; C++ классы интерфейса системы интерактивного объемно-ориентированного геометрического моделирования с возможностью простого механизма расширения новых алгоритмов и характеристик.

Эти иерархические классы составляют ядро пакета, который может быть расширен для добавления функциональности или изменения характеристик.

Другие задачи, включающие преобразование полигональных объектов в функциональные модели и обратное преобразование (триангуляцию), разработаны с использованием этих классов. В данной работе они не рассматриваются.

Для интерактивного моделирования необходимо было решить следующие задачи.

1) Реакцию программы на события от мыши (и, соответственно, выбор и модифицирование объектов). Необходимо уметь выделять объект (или несколько объектов) в сцене и предоставлять пользователю возможность проделать с ним (с ними) некоторые операции.

2) Выбор операции – аффинные преобразования MRS (move, rotate, scale), геометрические операции или деформация. Деформация состоит в возможности добавлять в любую точку на поверхности возмущение с параметрами

задаваемыми *инструментом*. *Инструмент* задает область действия и вид возмущения. Для этого необходимо графически предоставить пользователю некоторую информацию (в том числе выделять объект, например, цветом, или с помощью bounding box, рисовать оси и т.д.). Всю отрисовку «поддержки» работы с операциями лучше производить с помощью OpenGL – для скорости и функциональности.

3) Возможность работы со списком инструментов.

4) Запись в файл и загрузка дерева сцены.

5) Возможность создания новых тэгов и их распознавания.

Ниже показан псевдокод алгоритма добавления возмущения.

```
Setup temporary rendering context – camera matrix to restrict drawing to a small region of the viewport around the center of a picking region; clearing array of objects selected;
```

```
Render scene with RENDER_NAMES flag filling CVxManipulator::m_setObj set by calling VxDepthRende::PerformRender;
```

```
Sort objects in the CVxManipulator::m_setObj set
```

```
if the current design (manipulation) mode is “deformation” or Ctrl is pressed and there is any perturbation in the CVxManipulator::m_setObj set
```

```
remove all non-perturbation objects to ensure that will change only existing perturbations;
```

```
else remove all perturbation objects from CVxManipulator::m_setObj to ensure that will change only existing ‘base’ objects;
```

```
remove any descendant objects (since they would be altered automatically since their parents change).
```

Алгоритм 1: Алгоритм добавления возмущения.

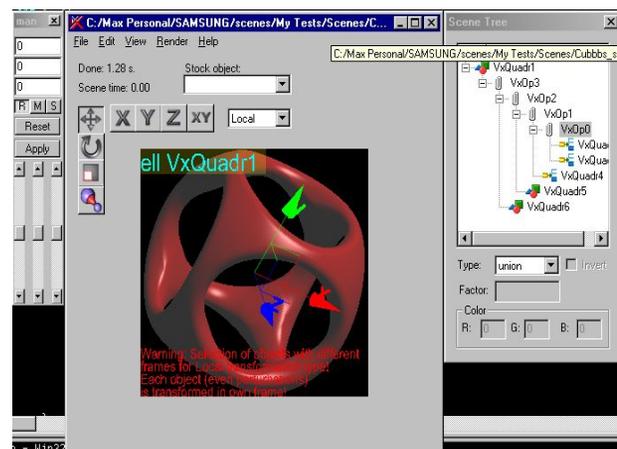


Рис 1: Главное окно программы.

## 6. ЗАКЛЮЧЕНИЕ

К основным достоинствам предлагаемого способа задания объектов и метода их визуализации следует отнести простоту вычисления точек поверхности. Задание объектов

поверхностями свободных форм сокращает в 100 и более раз описание баз данных по сравнению с заданием их полигонами.

Время визуализации зависит от сложности объекта. Для задания более сложных объектов требуется больше возмущений. Производительность сильно зависит от скорости работы памяти. Поэтому наибольшей производительности можно достичь при использовании только регистров графического акселератора.

В настоящее время в связи с развитием вычислительной техники численное решение задач трехмерной геометрии, требующих больших вычислительных ресурсов, находит все большее применение в реальных приложениях. Это моделирование динамики твердых и деформируемых тел, решение задач в области вычислительной биологии, молекулярного моделирования и т.д. Компьютерное трехмерное геометрическое моделирование позволяет получить и исследовать геометрические параметры объекта, проанализировать механические свойства, динамику поведения и взаимодействия объектов. Многообразие задач геометрического моделирования требует разработки эффективных методов и алгоритмов формирования компьютерных трехмерных геометрических моделей, способных обеспечить высококачественную и информативную визуализацию в реальном времени, используя стандартные современные программно-аппаратные средства. Функциональное задание объектов особенно актуально в ряде задач компьютерной графики, включая моделирование мягких или органических объектов, трехмерного морфинга, определения столкновений объектов и конструктивной блочной геометрии. Области применения функционально заданных объектов: молекулярная биология, интерактивные графические системы визуализации, CAD-системы, системы 3D-моделирования, 3D веб-визуализация, системы прототипирования и т. д.

## 7. ССЫЛКИ

- [1] Pasko A., Adzhiev V., Sourin, A., et al. Function representation in geometric modeling: concepts, implementation and applications //The Visual Computer, 11, 6, 1995, pp. 429-446.
- [2] McCormack J., Sherstyuk A. Creating and rendering convolution surfaces. Computing Graphics Forum. - 1998.-Vol. 17. - No.2. – pp. 113-120.
- [3] S. Muraki. Volumetric shape description of range data using “blobby model”. Computer Graphics. - July 1991. - 25(4) - pp. 227-235.
- [4] H. Nishimura, M. Hirai, T. Kawai, T. Kawata, I. Shirakawa, and K. Omura. Object modeling 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) - pp. 718-725.
- [5] G. Wyvill, C. McPheeters, and B. Wyvill. Data structure for soft objects. The Visual Computer. - 1986. - 2(4) - pp. 227-234.
- [6] K. Perlin, E. M. Hoffert. Hypertexture. Proceedings of the 1989 ACM SIGGRAPH conference, Volume 23, Issue 3 (July 1989), pp. 253 – 262.
- [7] D. Karla and A.H. Barr. Guaranteed ray intersections with implicit surfaces. *Computer Graphics*, 23:297-306, November 1989.
- [8] J. C. Hart. Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12:527-545, 1994.
- [9] D. Mitchel. Robust ray intersection with interval arithmetic. In *Proceedings on Graphics Interface 1990*, pp. 68-74. 1990.
- [10] A. Sherstyuk. Fast ray tracing of implicit surfaces. *Computer Graphics Forum*, Volume 18, Number 2, 1999, pp. 139-147.
- [11] M. Reimers, J. Seland. Ray Casting Algebraic Surfaces using the Frustum Form. *Eurographics*, Vol. 27 (2008), No. 2, pp. 361-370.
- [12] R.N. Perry and S.F. Frisken. Kizami: A System for Sculpting Digital Characters, in *SIGGRAPH'01*- 2001. – pp. 47-56.
- [13] M. Agrawala, A.C. Beers and M. Levoy, M. 1995. 3D Painting on Scanned Surfaces. - ISBN 0-89791-736-7. 1995- pp. 145-150.
- [14] B.M. OH, M. Chen, J. Dorsey and F. Durand. Image-based modeling and photoediting, in *SIGGRAPH'01*. - 2001. - pp. 433-442.
- [15] E. Ferley, M.-P. Cani and J.-D. Gascuel. Virtual Sculpture, *Visual Computer* 16:8. - 2000. - pp. 469-480.
- [16] Sourin. Functionally based virtual embossing. *The Visual Computer*.-7-2001. – pp. 258-271.
- [17] K.Levinski and A.Sourin, Interactive Function-Based Shape Modeling for Cyberworlds, 2004 International Conference on Cyberworlds, Tokyo, 18-20 November, 2004.-pp.54-61.
- [18] S. I. Vyatkin, “Complex Surface Modeling Using Perturbation Functions,” *Optoelectronics, Instrumentation and Data Processing*, Vol. 43, No. 3, 2007. pp. 40-47.
- [19] S. I. Vyatkin, B. S. Dolgovesov and A. T. Valetov, “Geometric Operations for Functionally Defined Objects Using Perturbation Functions,” *Optoelectronics, Instrumentation and Data Processing*, Vol. 40, No. 1, 2004, pp. 65-73.
- [20] S. I. Vyatkin, B. S. Dolgovesov and A. S. Korsun, “Collision Detection of Functionally Defined Objects in Computer Graphics Tasks,” *Optoelectronics, Instrumentation and Data Processing*, Vol. 39, No. 6. 2003, pp. 119-126.
- [21] Вяткин С.И. Трехмерные объекты на базе функций возмущения - LAP LAMBERT Academic Publishing GmbH & Co. KG, Saarbrücken Germany (ID: 45569) ISBN: 978-3-8484-2398-9 – 2012. (128 стр.)
- [22] Vyatkin S.I., Gorodilov M., Dolgovesov B.S. GPU-based Binary Adaptive Ray Casting for Freeform Objects with Perturbation Functions// Proceedings of the IASTED International Conferences on Automation, Control, and Information Technology (ACIT 2010), vol. “Information and Communication Technology”, “Optical Information Technology”. Acta Press, Calgary, 2010. pp. 223–228.