

УДК 004.925.8

Вяткин С.И.

Институт автоматики и электрометрии
Сибирского отделения Российской академии наук

Романюк А.Н.

Винницкий национальный технический университет

Романюк О.В.

Винницкий национальный технический университет

Лысенко Е.С.

Винницкий национальный технический университет

ТЕОРЕТИКО-МНОЖЕСТВЕННЫЕ ОПЕРАЦИИ НАД СВОБОДНЫМИ ФОРМАМИ

В работе предложен метод вычисления булевых операций (объединения, пересечения, разности), применяемых к твердым телам свободной формы, ограниченными поверхностями свободных форм. Разработаны алгоритмы генерации поверхности с несколькими разрешениями, ограничивающей объем.

Ключевые слова: теоретико-множественные операции, свободные формы, функции возмущения, объемы.

Постановка проблемы. Булевы операции – естественный способ построения сложного твердого тела объекта из простых примитивов. Такой подход очень распространен в автоматизированном геометрическом дизайне, где много искусственных объектов могут строиться из простых частей, таких как цилиндры, прямоугольные блоки и сферы. Некоторые вычислительные представления твердых тел замкнуты относительно логических операций. Это означает, что результат логической операции не может быть представлен точно в большинстве случаев. Один из путей избежания этих проблем заключается в использовании дерева логических операций в качестве представления объектов и реализации различных алгоритмов непосредственно для такого представления. Такой подход называется конструктивной твердотельной геометрией (CSG) [1]. Однако для многих приложений CSG – не самый эффективный или соответствующий метод. Чаще всего используются граничные представления (В-гер) твердых тел. И булевы операции должны быть реализованы в В-гер базе. Такое применение довольно сложно для высшего порядка В-гер, так как требует пересечений параметрических поверхностей, разделяя их на куски при строительстве новых поверхностей из этих частей. Существующие системы обычно рассматривают В-Rep как набор обрезанных участков сплайна, разделяющего границы. Границы из

отдельных патчей часто совпадают лишь приблизительно. Каждая операция пересечения ведет к более сложным и трудно применимым обрезкам кривых. Ее трудно применить для гладких деформаций к результирующим моделям. С особой тщательностью должны быть приняты меры, чтобы избежать трещин и т. д. Элементарная операция, необходимая для этого представления поверхности, – пересечение двух обрезанных NURBS патчей. Однако это является сложной проблемой. В результате логические операции часто выполняются медленно и не полностью надежны. При этом отличные результаты достигаются с помощью сплошного твердотельного моделирования.

Анализ последних исследований и публикаций. Результаты булевых операций на В-Rep твердых тел показаны в работе [2]. Использование специальной схемы комбинированного деления описано в [3] для представления кривой пересечения. Существует обширная литература по твердотельному моделированию с В-Rep [4; 5].

В этой статье мы представляем новый подход к вычислению результата булевых операций над свободными формами на основе функций возмущения.

Изложение основного материала исследования.

Свободные формы объектов

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

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

где x , y и z – пространственные переменные. Можно записать это неравенство в матричном виде:

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

На базе квадрик строится класс свободных форм с использованием функций возмущения.

Предлагается описывать сложные геометрические объекты, задавая функцию отклонения (второго порядка) от базовой квадрики в виде (1).

Свободная форма есть композиция базовой квадрики и возмущения

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

где функция возмущения $R(x, y, z)$ находится следующим образом:

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

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

При решении описывающей функции в виде неравенства $F(X) \geq 0$ можно визуализировать не только поверхность, но и внутреннюю структуру объекта.

Теоретико-множественные операции

Примитивы можно комбинировать с помощью трехмерных теоретико-множественных операций, таких как объединение, логическая сумма (соединение двух объектов – $A \cup B$ или $(A+B)$ – объединение множеств A и B). Пересечение, логическое произведение (выделение общего подмножества – $A \cap B$ или (AB) – множества A и B имеют общую часть, пересекаются). Разность (взятие всего первого объекта за исключением тех его частей, которые являются общими со вторым объектом – $A-B$).

Для того чтобы создать сложную сцену, необходимо в ней описать какое-то определенное количество примитивов, необходимых для конкретной задачи. Отображаемый объект, с которым алгоритм растеризации осуществляет взаимодействие посредством запросов, представляет собой всю трехмерную сцену, поэтому геометрическая модель должна позволять конструировать объекты и их композиции неограниченной сложности. Достигается это в первую очередь применением булевых операций объединения и пересечения. В описываемой системе визуализации определен объект особого вида, осуществляющий логические операции над объектами любых видов, поэтому вся сцена представляет вид дерева, каждый узел которого является *объектом-конструктором*, осуществляющим логические операции над своими потом-

ками, а вершинами дерева являются примитивы, используемые системой. В момент, когда растровый алгоритм обращается с каким-либо запросом к объекту-конструктору, этот объект обращается к своим потомкам, преобразует полученный результат, и выдает соответствующий ответ на запрос. При этом потомком может являться как примитив, так и другой объект-конструктор. При применении геометрических операций, поворотов, перемещений, масштабирования к объекту-конструктору он производит все эти операции со своими потомками, а в случае инвертирования изменяет еще свою булеву функцию.

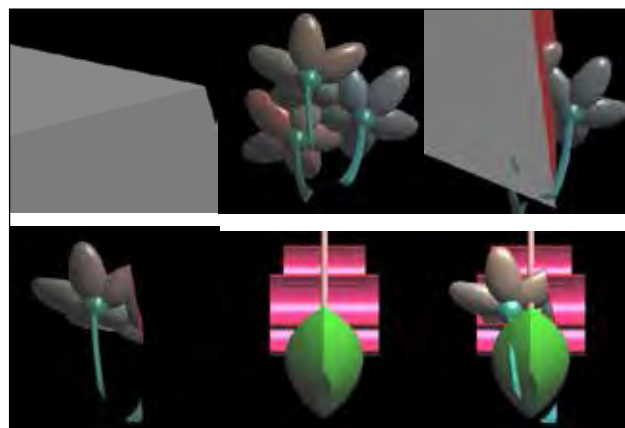


Рис. 1. Теоретико-множественные операции над объектами

Адаптированный алгоритм визуализации

Для простоты понимания будем считать, что сцена находится в единичном трехмерном кубе (рис. 2). Перспектива рассматриваться не будет ввиду того, что она сводится к переходу в другую систему координат. Так как пирамида видимости будет представлять собой куб в новой системе координат, поэтому опустим начальные преобразования и уделим больше внимания основной части алгоритма. Будем считать, что наблюдатель смотрит вдоль оси Z . Необходимо получить проекцию сцены на плоскость XY . Проекция должна представлять собой конечный набор значений, поэтому весь куб будет делиться на «бруски» так, чтобы каждый брусок соответствовал пикселу на изображении. Каждый из брусков будет делиться вдоль оси Z , образуя набор вокселей (рис. 3). Так как размеры бруска в плоскости XY значительно меньше, чем в направлении оси Z , брусок можно рассматривать как луч. Таким образом, получим функцию плотности вдоль луча, которая зависит от одной переменной. Задача будет состоять в нахождении первой точки, в которой функция обращается в ноль. Найдя такую точку для каждого луча, будет известна глубина кадра. Данными

приближениями задача сводится к задаче отслеживания лучей. Далее в каждом пикселе можно вычислить нормаль. А имея данные о глубине и нормали в каждом пикселе, можно использовать модель локального освещения. В итоге получится изображение гладкого объекта с учетом освещения. Главной частью работы является эффективное нахождение первого пересечения луча с поверхностью. Данная задача напоминает задачи визуализации объемных данных, которые часто применяются, например, в томографии. В подобных задачах задана функция плотности. Основным отличием является то, что в подобных задачах мы имеем дело с дискретными данными. А в нашем случае есть аналитически заданная функция плотности. Это позволяет более эффективно осуществлять поиск. Основным отличием адаптированного метода бинарного деления объектного пространства является отсутствие четверичного деления. Куб делится на части в плоскости XY соответственно пикселям на изображении.

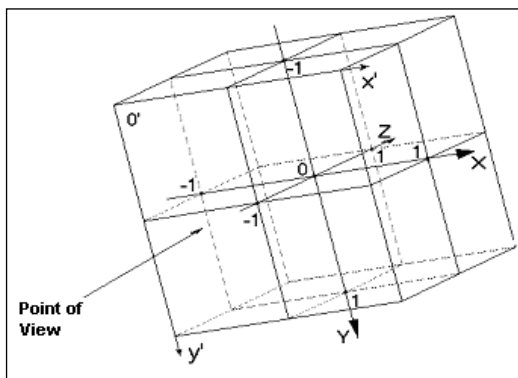


Рис. 2. Единичный куб

Основным отличием адаптированного метода бинарного деления объектного пространства является отсутствие четверичного деления. Куб делится на части в плоскости XY соответственно пикселям на изображении.

Во второй части метод остается прежним. В этом случае уменьшение времени на визуализацию достигается за счет эффективного использования вычислительных ресурсов графического акселератора.

Описание приложения с использованием NVIDIA CUDA

Для реализации была использована Compute Unified Device Architecture (CUDA) от компании NVIDIA. CUDA – это модель параллельного программирования вместе с набором программных средств, которая позволяет реализовывать программы на языке C для исполнения на графическом акселераторе.

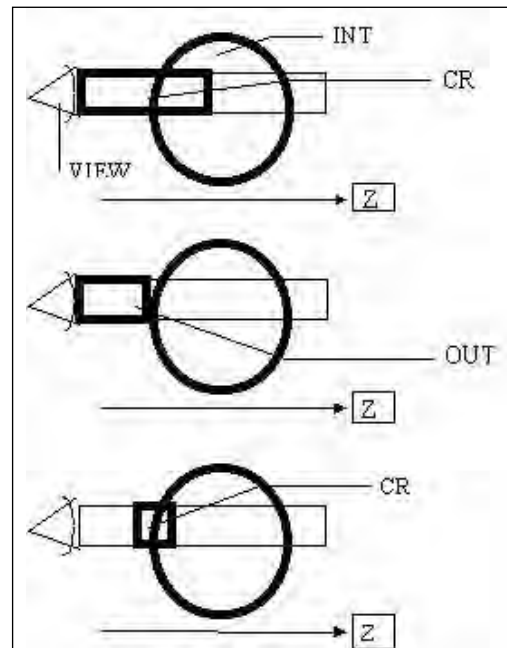


Рис. 3. Бинарное деление луча для вычисления пересечения луча с поверхностью

Достоинством CUDA является и то, что скомпилированная программа будет выполняться на различных графических акселераторах. И результат выполнения будет одинаковый, даже несмотря на то, что у них может быть различное число потоковых мультипроцессоров.

Вернемся к анализу метода. На первых шагах деления куба очень сложно обеспечить параллельную обработку данных, и это потребует применения синхронизации. А в среднем выигрыш в скорости обработки будет меньше, чем накладные расходы, связанные с усложнением модели для дальнейшей обработки, и далеко не все вычислительные ресурсы будут использованы. Теперь рассмотрим деление со стадии, когда куб разбит на достаточное количество частей так, чтобы вычислительные процессоры не простаивали. Первая проблема, с которой мы столкнемся, будет состоять в том, что у разных графических акселераторов разное количество процессоров, и надо будет выбирать во время исполнения оптимальную стадию, с которой начинать работать. Таким образом, можно частично избавиться от проблемы с неиспользуемыми вычислительными ресурсами. Но остается задача балансировки. В этой части вычислений повышение производительности достигается за счет того, что какие-то части будут отброшены на раннем этапе деления куба. Если убрать из рассмотрения какую-нибудь часть пространства, возможно, освободятся вычислительные ресурсы. Но тогда встает задача балансировки. И накладные расходы на ее решение будут превосходить выигрыш в производительности от этой процедуры. Поэтому было принято решение отказаться от четверичного деления кубического

пространства в методе адаптированном для графических ускорителей.

В адаптивном методе используется большое количество вычислительных процессоров, одновременно будет происходить проверка сразу нескольких лучей. А в большей части графических акселераторов, поддерживающих CUDA, не менее ста двадцати восьми скалярных вычислительных ядер. Следовательно, будет отбрасываться достаточно большая часть куба. Стоит заметить, что в последовательном варианте (исходном методе), чтобы отбросить часть куба вблизи с объектом, необходимо поделить рассматриваемую область на более мелкие части. В параллельном варианте это не будет иметь значения, и большая часть будет сразу отброшена.

Как уже упоминалось, во второй стадии метода (двоичное деление) принцип работы последовательного и параллельного вариантов практически не отличается. Отличие лишь в способе задания объектов. В последовательном варианте объекты задавались в виде иерархической модели. То есть в качестве возмущения мог быть другой объект, у которого есть свои возмущения. В адаптивном методе возмущения задаются явно в виде квадрик. Это обусловлено тем, что при написании ядер нельзя использовать рекурсию. Несмотря на это, разнообразие форм, которые можно задать, не меньше чем с использованием иерархического представления. Разница состоит только в удобстве использования. Иерархическое представление является более естественным для такого метода задания объектов.

При реализации учитывалось влияние скорости работы с памятью. Максимально использованы регистры, потому что это самый быстрый вид доступной памяти. Следующая по скорости – совместно используемая память. Во всех остальных случаях используется общая память графического акселератора. Было реализовано приложение, которое визуализирует свободные формы, заданные квадриками с возмущениями на графическом акселераторе. В функции графического акселератора входили расчет глубины кадра, нормалей и освещения. Центральный процессор (ЦПУ или CPU) использовался для геометрических преобразований. Для отображения изображения использовалась DirectX. Тестирование производилось на процессоре Intel Core2 CPU E8400 3.0 GHz, GPU 9800 GT и 470 GTX.

Выводы. Научной областью, к которой относится данная работа, является объемно-ориентированное геометрическое моделирование функционально заданных объектов. Областью практического приложения результатов работы является создание интерактивных графических систем объемной визуализации, САПР, 3D Web-визуализация и т. д. В работе представлены свободные формы на базе аналитических функций возмущения. Свободные формы (квадрики) с функциями возмущения обладают достоинством сплайнового представления поверхностей – высокой степенью гладкости, но главной их отличительной чертой является произвольность формы при небольшом количестве функций возмущения.

Список литературы:

1. Hoffmann C. M. Geometric and solid modeling: an introduction. San Mateo, California: Morgan Kaufmann, 1989.
2. Litke N., Levin A., Schroder P. Trimming for subdivision surfaces. Technical report, Caltech, 2000.
3. Levin A. Combined subdivision schemes for the design of surfaces satisfying boundary conditions. *Computer Aided Geometric Design*, 16(5):345–354, 1999.
4. Agrawal A., Requicha A. A paradigm for the robust design of algorithms for geometric modeling. *Computer Graphics Forum*, 13(3):33–44, 1994.
5. Rossignac J., Requicha A. Solid modeling. In J. Webster, editor, *Encyclopedia of Electrical and Electronics Engineering*. John Wiley and Sons, 1999.

ТЕОРЕТИКО-МНОЖИННІ ОПЕРАЦІЇ НАД ВІЛЬНИМИ ФОРМАМИ

У роботі запропоновано метод обчислення булевих операцій (об'єднання, перетинання, різниці), що застосовуються до твердих тіл вільної форми, обмежених поверхнями вільних форм. Розроблено алгоритми генерації поверхні з кількома дозволами, що обмежує об'єм.

Ключові слова: теоретико-множинні операції, вільні форми, функції збурення, об'єми.

THEORETICAL AND MULTIPLE OPERATIONS OVER FREE FORMS

The paper describes a method for calculating Boolean operations (combinations, crossings, differences) applied to solids of free form, bounded surfaces of free forms. The algorithms of surface generation with several resolutions limiting the volume are presented.

Key words: multiple-theoretical operations, free forms, functions of disturbances, volumes.