

ВИЗУАЛИЗАЦИЯ СВОБОДНЫХ ФОРМ И ВОКСЕЛЬНЫХ ОБЪЕМОВ В РЕАЛЬНОМ ВРЕМЕНИ

Вяткин С.И., Долговесов Б.С., Есин А.В., Жигач А.А., Чижик С.Е., Щербаков Р.А.
Институт автоматики и электрометрии
Новосибирск, Россия

Аннотация

Обсуждается проблема синтеза изображений фотореалистичного качества в реальном времени. Рассматриваются новые способы задания поверхностей свободных форм без аппроксимации их полигонами или патчами. Предложен рекурсивный алгоритм деления объектного пространства с многоуровневым маскированием и с учетом перспективы и коррекции дисторсии. Предлагаются новые способы генерации трехмерных текстур, ограниченных произвольными поверхностями. Показана возможность визуализации массивов трехмерных скалярных данных. Приводятся примеры изображений, полученных при моделировании работы алгоритма с разными способами задания объектов.

Ключевые слова: Свободные формы, текстура формы, воксельные объемы.

1. Введение

С появлением систем виртуальной реальности, в которых наблюдатель погружается в мир модели, возникает потребность визуализации виртуальной среды, максимально приближенной к тому, что человек наблюдает в естественных условиях. Для этого требуются мощные системы, такие как Oplex2 Reality Monster фирмы "Silicon Graphics" стоимостью около 100 тыс. долларов. Кроме этого, существуют специфические задачи, такие как, например, задача динамической коррекции дисторсии, возникающей при проецировании изображения на криволинейные экраны с учетом подвижности наблюдателя и проектора, являющейся неотъемлемой частью дорогих систем визуализации реального времени (системы семейства ESIG и HARMONY фирмы "Evans & Sutherland").

Все более широкое распространение получают сегодня методы визуализации внутренних структур объектов. Традиционная 3D графика основывается на представлении объектов поверхностями, как правило – полигональными. В качестве альтернативы поверхностям существует представление объектов в виде объемов. Объем – это трехмерный массив кубических элементов

(вокселей), представляющих единицы 3D пространства по аналогии с двумерным массивом пикселей, каждый из которых представляет единицу площади. Несмотря на недостатки большого количества вычислений и данных, объем имеет очень важные достоинства: он может представлять внутренность объекта, а не только внешний слой, в общем виде, рендеринг и обработка зависят не от сложности объектов, а только от разрешения объема. Эти методы компьютерной графики называются визуализацией объема. Существуют два класса методов визуализации объема с точки зрения примитивов. В методах, основанных на представлении объектов поверхностями, сначала создается промежуточная модель с выделением поверхности объекта. Далее выполняется рендеринг поверхностей. В последнем проекте под названием Volumizer компания "Silicon Graphics" расширила API-стандарт введением объемных элементов изображения – вокселей, ограниченных четырьмя треугольными гранями – тетраэдрами. Другие методы, основанные на воксельном представлении объемов, создают трехмерные изображения объекта непосредственно из объемных данных. Впервые в мире компанией "Mitsubishi Electric" разработан заказной чип vg500 системы объемной визуализации VolumePro производительностью 500 миллионов вокселей в секунду, что позволяет визуализировать объем с разрешением 256x256x256 в реальном времени (30 Гц) [1]. Оба эти метода имеют свои достоинства и решение о том, какой из них следует использовать для конкретного приложения, зависит от целей визуализации. В компании "Hewlett-Packard" разработан метод визуализации объема через Фурье-преобразование. Этот метод использует не пространственное описание объектов, а их частотные спектры [2]. Это быстрый метод, однако его применимость ограничена простыми интегральными проекциями. Очень перспективный метод использует вейвлетные преобразования (wavelet transforms). Оба эти метода относятся к подходу на основе преобразований в котором создается мультимасштабное представление объектов, что позволяет очень сильно сократить количество данных и время вычислений.

Существуют четыре способа сканирования при визуализации объема: растризация полигонов в пространстве экрана (визуальный образ объемных данных строится путем рендеринга полигонов, произведя рендеринг полигонов в порядке от заднего плана к переднему и смешивая их в буфере кадра, получают изображение объема); рендеринг в объектном пространстве или рендеринг прямого хода (к этому способу относятся алгоритмы множественных преобразований [3,4], алгоритмы сплатинга [5-7], сдвиговые алгоритмы [8,9]); рендеринг обратного хода (к этому способу относятся алгоритмы трассировки [10] и отслеживания лучей [11-14]); гибридные способы, например, обратное инкрементальное преобразование в объектное пространство реализовано на IBM Power Visualization System (PVS) [15].

В данной работе предлагается метод, сочетающий возможности визуализации функционально-заданных поверхностей свободных форм, с использованием вещественных и скалярных функций и неоднородных объемных областей трехмерных массивов вокселей.

2. Поверхности свободных форм

Характерной особенностью предлагаемых способов задания поверхностей, во-первых, является то, что основными примитивами выбраны поверхности второго порядка - квадратики. На этой базе строится первый класс свободных форм, с использованием вещественных функций. Свободная форма – это произвольная поверхность, обладающая свойствами гладкости, непрерывности и неразрывности. Поверхности такого типа выгодно применять при синтезе искусственных (рукотворных) объектов. Во-вторых предлагается второй класс свободных форм, с использованием скалярных функций возмущения относительно базовой поверхности, например, для генерации рельефа местности или скульптурных моделей на базе вокселей. И - третьих, предлагается использовать совместно с первыми двумя классами – третий, о котором уже кратко упоминалось выше, т.е. представление объектов в виде объемов.

2.1. Функции возмущения в неявном виде

Реализована возможность описывать сложные геометрические объекты, задавая функцию отклонения (второго порядка) от базовой поверхности тоже второго порядка. В простейшем случае $F(x,y,z)$ задает поверхность второго порядка:

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

На базе квадратиков строятся поверхности свободной формы. Поверхность свободной формы есть композиция базовой поверхности квадрата и возмущения $F'(x,y,z) = F(x,y,z) + R(x,y,z)$, где функция возмущения $R(x,y,z)$ находится следующим образом:

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

$Q(x,y,z)$ – возмущающий квадратик.

В качестве Q так же может быть возмущенный квадратик (поверхность свободной формы). Другими словами, композиция базовой поверхности и функции отклонения является новой функцией возмущения, т.е., производной для другой базовой поверхности. Получающаяся поверхность будет гладкой (рис. 1), и потребуются небольшое количество функций возмущения для создания сложных форм поверхностей. Таким образом, задача конструирования объекта сводится к задаче деформации поверхности квадрата нужным образом, а не к аппроксимации его примитивами (полигонами или патчами, представленными B-сплайнными поверхностями).

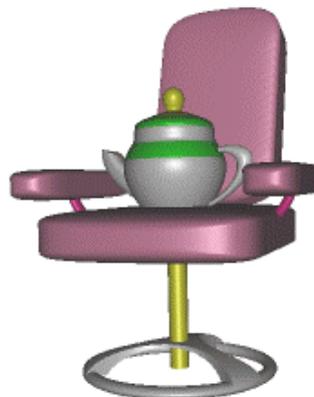


Рис. 1. Сцена, описанная сорока пятью функциями 2-го порядка.

2.2 Функции возмущения в скалярном виде

Предлагается описывать сложные геометрические объекты, задавая функцию отклонения (в скалярном виде) от базовой поверхности второго порядка (Рис. 2) или, в простейшем виде, от базовой плоскости (Рис. 3). Рельеф местности является частным случаем таких объектов и задается с помощью базовой плоскости и функции возмущения, которая определена в бесконечно длинном параллелепипеде. Значения функции возмущения задаются в поперечном сечении

параллелепипеда двумерной таблицей высот. В

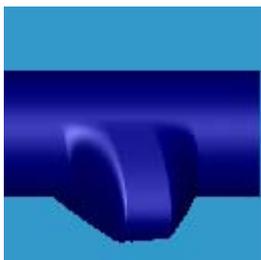


Рис. 2.

качестве базовой поверхности можно использовать плоскость, тогда направление нормали несущей плоскости должно совпадать с продольным направлением параллелепипеда - области определения функции возмущения.

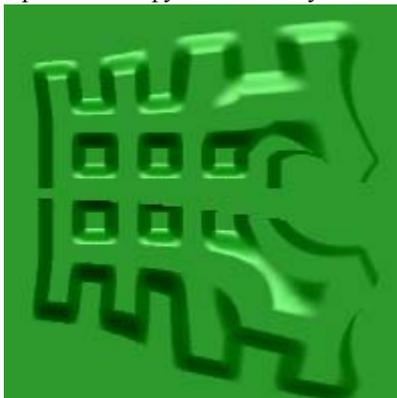


Рис. 3.

Поскольку во время растризации необходимо оценивать максимум функции на трехмерном или одномерном интервале, то для эффективности вычислений предварительно

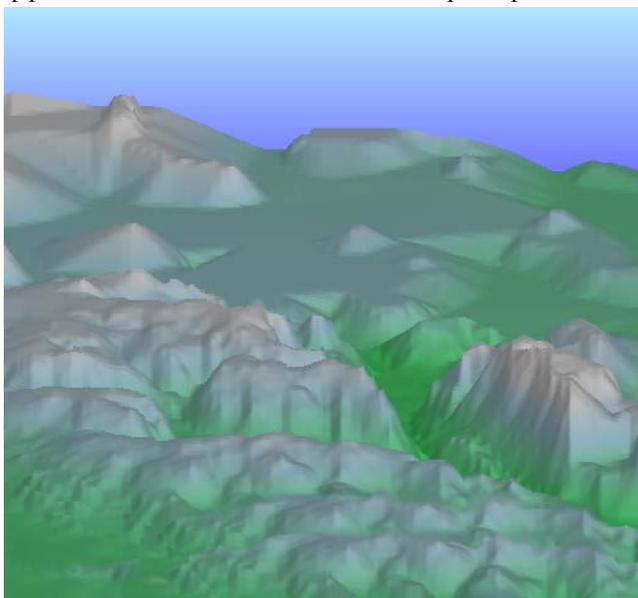


Рис. 4. Горный ландшафт на основе одной плоскости и сетки высот разрешением 64x64.

составляются карты *уровней детальности*. Исходные данные составляют уровень n , если размерность сетки равна $2^n \times 2^n$. Данные для уровня $n-1$ получаются выбором максимума из четырех смежных значений уровня n , остальные три далее не учитываются, т.е. получаем размерность сетки $2^{n-1} \times$

2^{n-1} . Уровень 0 состоит из единственного значения - максимума по всей карте высот.

При определении максимума возмущения вычисляется характерный размер проекции текущего интервала, исходя из которого выбирается уровень детальности. Для большего интервала соответственно выбирается более грубое приближение исходной функции. Если требуется более точное представление, чем имеется в наличии, то производится билинейная или бикубическая интерполяция значений высот, составляющих последний уровень детальности. Таким образом, рельеф местности можно задать картой (сеткой) высот и без предварительной триангуляции его отобразить, получая при этом высокую реалистичность (рис. 4).

3. Трехмерная текстура

В случае, когда объекты являются полупрозрачными, поверхности свободных форм используются как ограничивающие оболочки для трехмерной текстуры (рис. 5). При визуализации объемных данных (значения плотности заданные на трехмерной сетке), существует проблема заключения данных в ограничивающую оболочку (как правило, в качестве ее берут выпуклый многогранник, а сканирующий луч проверяют на пересечение с полигонами многогранника). Эффективность алгоритма сканирования будет определяться плотностью подгонки оболочки к объемным данным. Предлагаемый способ задания, как нельзя лучше, подходит для решения данной проблемы. Поверхности свободных форм, способы задания которых были описаны выше, используются как ограничивающие оболочки для трехмерной текстуры.

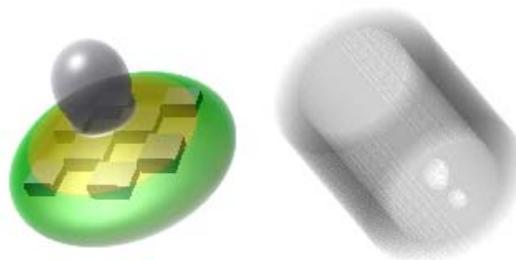


Рис. 5. Полупрозрачная поверхность, описанная четырьмя функциями второго порядка, заполненная трехмерной текстурой и цилиндр, синтезированный непосредственно из трехмерной карты плотностей.

Вообще, в данном случае, объемную текстуру можно рассматривать как отдельный объект, во всем пространстве возвращающий положительный ответ на запрос о пересечении. Нанесение текстуры на объект можно интерпретировать как пересечение данного объекта с объектом-текстурой, при этом цвет и непрозрачность выбираются не из атрибутов

объекта, а из текстурной карты. Возможны два подхода.

1.) Изображение объема генерируется в процессе сканирования из трехмерной карты плотности (рис. 5, цилиндр). При этом прозрачность элементов изображения выбирается в зависимости от значения плотности, а нормаль к изоповерхности (которая нужна для модели освещения) вычисляется как градиент плотности [16].

2.) Трехмерная текстура генерируется аналогичным образом, но до процесса визуализации. Преимущество этого подхода в том, что резко сокращаются вычислительные затраты. Для того чтобы вычислить вектор нормальный к поверхности из дискретной сетки плотностей требуется несколько операций сложения и умножения, и, что самое главное, необходимо полученный вектор нормализовать, для этого требуется операция вычисления квадратного корня и деления. Если нормаль вычислена заранее, то в процессе визуализации нужно только перевести нормаль из текстурной системы координат в объектную, что делается соответствующей матрицей поворота. К недостаткам данного подхода следует отнести увеличение объема памяти для хранения текстуры.

4. Способ растризации

Метод растривания включает алгоритм многоуровневого отслеживания лучей (multi-level ray casting) [17], осуществляющий эффективный поиск элементов объема - вокселей, участвующих в формировании изображения. Отличием алгоритма отслеживания лучей (ray casting) от трассировки лучей (ray tracing) является то, что в первом из них в целях увеличения скорости вычислений не отслеживаются вторичные лучи. Поиск производится в пространстве внутри куба, от -1 до +1 по каждой координате, так что центр куба соответствует началу координат. На первом шаге рекурсии исходный объем разбивается на четыре меньших объема в экранной плоскости. Для каждого нового объема выполняется тест на пересечение с объектом. Если пересечение имеет место, то объем подвергается следующему уровню рекурсии. Объемы, не пересекающиеся с объектом, дальнейшему погружению в рекурсию не подлежат, что соответствует исключению из рассмотрения квадратных областей экрана, на которые данный объем (следовательно, и поверхность объекта) не отображается. Деление объема ведется до тех пор, пока его размер не станет равным размеру вокселя (т.е. достигается максимально возможный уровень рекурсии). Преимущество этой методики в том, что она позволяет на ранней стадии отбросить большие части пустого пространства. В процессе поиска вокселей, содержащих в себе участки поверхности объекта, формирующие изображение, алгоритм осуществляет обход кубического пространства по четверичному дереву, листья которого являются корнями двоичных деревьев. В процессе обхода дерева используется механизм многоуровневого

маскирования в случае непрозрачных объектов. Метод многоуровневого отслеживания лучей позволяет эффективно и быстро определить принадлежность лучей разных уровней (пирамид) поверхностям квадриков и отбраковать области пространства вне объектов. Наиболее простое деление пространства осуществляется в кубе или параллелепипеде. Однако, при переходе к пирамидальному пространству (с учетом перспективы), можно столкнуться с проблемами, либо увеличения порядка поверхностей, а, следовательно, сложности и количества вычислений, либо неоднородности пространственного описания элементов изображения (вокселей, воксельных объемов). Эти проблемы еще более усложняются при решении задачи компенсации дисторсии (каждый пространственный элемент изображения становится уникальным, что также затрудняет качественную фильтрацию). Решена задача простого (как в кубе) деления перспективного пространства (при этом пирамидальное пространство может быть достаточно произвольной формы - что очень важно для компенсации дисторсии) без увеличения порядка поверхностей за счет трансформации пространства в пространство. Применение *проективного преобразования* обобщает приведенный алгоритм на пирамидальные объемы, что позволяет синтезировать изображения с перспективой. В трехмерном пространстве точке с декартовыми координатами (x, y, z) соответствует бесконечное множество однородных координат (x', y', z', a) таких, что $x=x'/a$, $y=y'/a$, $z=z'/a$, т.е. однородные координаты определяются с точностью до общего ненулевого множителя. Представляет интерес матрица преобразования, действующая на однородные координаты следующим образом [18]:

$$(C)(M) = (P),$$

где (C) - матрица преобразования, (M) - однородные координаты точки пространства M, (P) - соответствующие по отображению координаты в P.

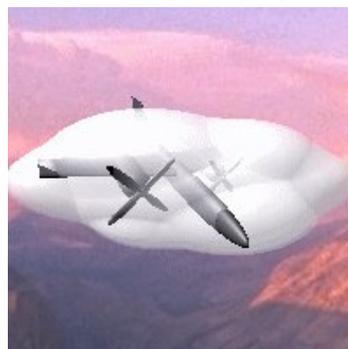


Рис. 6. Сцена, иллюстрирующая комбинацию поверхностей и воксельных объемов.

Вышеописанные проективные преобразования дают возможность реализовать динамическую коррекцию искажений проекционной системы. Данными для вычисления координат

опорных точек искривленного объектного пространства являются единичные векторы (храняемые в таблице и пересчитываемые в каждом кадре при динамической коррекции искажений). Орбиты вокруг наблюдателя (на 360 град. во всех направлениях) пересекаются трансформированными пирамидами видимости, которые образуют искаженное сферическое объектное пространство. Для того чтобы не было разрывов поверхностей при больших искажениях, вводится апертура, охватывающая воксели разных уровней пространства. Апертура зависит от степени искажений, при больших искажениях апертуру требуется увеличивать. За счет апертуры и субвоксельного разрешения устраняются дефекты зыбкой границы объектов, так как нет потери отдельных вокселей, а границы хорошо фильтруются. При пересечении вокселя последнего уровня с поверхностью мы имеем уравнение касательной плоскости к поверхности в данной точке, а, следовательно, и нормали. Интерполяция производится по Фонгу в объектном пространстве с учетом источника освещения и искажений. Текстурируются поверхности с учетом дисторсии; уровни детальности зависят от искажений, которые известны в процессе сканирования пространства, то же самое относится к вычислению функции тумана. Таким образом, мы комплексно решаем все части данной задачи в отличие от традиционного полигонального задания примитивов, где вначале корректируется форма объектов в процессе разложения многоугольников в телевизионный растр (геометрия ограничивающих многоугольник ребер, при этом многоугольник остается плоским), а затем его цвет, текстура и т.д. с учетом дисторсии.

При визуализации поверхностей проверяется тест на принадлежность им лишь пересеченных вокселей, внешние и внутренние воксели отбраковываются. Если делить и внутреннее пространство, можно выделять объемы. Внутренние воксели, также как и пересеченные могут быть различных уровней. Для того чтобы повысить реалистичность изображения и расширить класс отображаемых объектов (полупрозрачные структуры с внутренним распределением плотности, 3D текстуры) необходимо отображать внутреннюю полупрозрачную структуру объекта. Для этого в формировании изображения должны участвовать не только воксели (элементы объема), которые лежат на поверхности, а так же и те, которые находятся внутри объекта. Следовательно, при делении объема внутренние части объекта не отбрасываются, для них проводится дальнейшая рекурсия алгоритма. При сканировании объема по X и Y координатам алгоритм работает аналогично алгоритму при рендеринге поверхности, - при отсутствии пересечения бруска с объектом данная область пространства отбрасывается, а при их пересечении происходит погружение на следующий уровень рекурсии, пока не достигается максимально возможный уровень, устанавливаемый пользователем. При сканировании сцены по Z

координате, что соответствует сканированию объема в глубину, алгоритм не прерывается при встрече с поверхностью, а работает далее, пока полностью не просканируется объем или не накопится определенное значение прозрачности большее некоторого порогового значения. Для уменьшения времени вычислений алгоритм адаптирован к быстрому прохождению однородных областей объектов, для которых совсем необязательно полностью сканировать объем, доходя до последнего уровня рекурсии, а следует 'проскакать' пустые или однородные участки по Z координате, и сразу же вычислить цвет и общую прозрачность. Так как прохождение лучом пустого пространства не вносит вклад в конечное изображение, то скачок через пустое пространство способен обеспечить существенное ускорение обработки и не влияет на качество изображения. Вследствие того, что объемные области могут ограничиваться произвольными поверхностями можно получить дополнительное ускорение обработки объема. В методе перескока можно эффективно использовать ограничивающие оболочки. Объект, находящийся в объеме, окружается плотно подогнанной произвольной поверхностью – оболочкой, с которой легко вычислить пересечение луча, как уже было описано выше. После нахождения пересечения луча с ограничивающим объектом - обход существенной части объема начинается с этой точки пересечения, а не от границы объема. С целью уменьшения вычислений можно выделять лишь поверхность, ограничивающую объем, если в объем попадает непрозрачный объект, то выделяется его поверхность в объеме взвешиванием интенсивностей с учетом функции затухания (Рис. 6).

5. Вычисление цвета

Вычисление всех составляющих цвета пиксела производится аналогично по следующим формулам:

$$C = (Q_{\text{ambi}}C_{\text{ambi}} + Q_{\text{diff}}C_{\text{diff}} + Q_{\text{spec}}C_{\text{spec}}) / (Q_{\text{ambi}} + Q_{\text{diff}} + Q_{\text{spec}});$$

где индекс "ambi" относится к характеристикам *рассеянного* излучения, а "diff" и "spec" - к *диффузной* и *зеркальной* частям отраженного света соответственно; C - компоненты цвета; Q - весовые коэффициенты. Вычисления компонент цвета производятся на основе векторной модели освещения. В расчетах участвуют четыре вектора: нормаль к поверхности (n), вектор на источник освещения (l), направление отраженного света (r) и вектор на наблюдателя (v):

$$C_{\text{diff}} = (n, l) C_{\text{lite}} C_{\text{surf}}; \text{ где } C_{\text{lite}} - \text{цвет источника, } C_{\text{surf}} - \text{цвет поверхности.}$$

$$C_{\text{spec}} = (r, v)^p C_{\text{lite}}; \text{ где } p - \text{коэффициент шероховатости поверхности.}$$

При моделировании прохождения света через полупрозрачные среды в целях уменьшения количества вычислений можно не учитывать преломление и затухание вторичных лучей. В случае, когда остается только отражение и затухание света на пути следования от объекта к глазу наблюдателя, формулу, по которой вычисляется цвет пиксела можно выразить следующим образом [16]:

$$P_{\lambda} = \sum_{n=0}^N I_{\lambda n} \Omega_n \prod_{m=0}^{n-1} (1 - \Omega_m)$$

Где P_{λ} конечный цвет пиксела, а λ может быть r, g или b (т.е. красный, зеленый или синий, соответственно), $I_{\lambda n}$ интенсивность в n-ом вокселе, вычисленная по модели освещения Фонга, Ω_n непрозрачность n-го вокселя. $I_{\lambda 0}$ отраженный свет от первой точки на луче сканирования, $I_{\lambda N}$ цвет фона и $\Omega_N = 1$. Преодоление порога можно отследить следующим образом, - если на k-ом шаге общая прозрачность $(1 - \Omega_0)(1 - \Omega_1) \dots (1 - \Omega_{k-1})$ становится меньше некоторого ε , то это значит, что вклад всех следующих за k-ым вокселей будет малым и, поэтому, сканирование можно прекратить.

Чтобы учесть перспективу, следует внести поправку в алгоритм накопления цвета в пикселе. Дело в том, что размеры вокселя, в результате трансформации геометрических примитивов, становятся зависимыми от координаты Z, поэтому при пересчете цвета в пикселе прозрачность вокселя следует также пересчитывать с поправкой на изменение его длины.

6. Фильтрация изображений

Фильтрация осуществляется следующим образом: 3D субвоксельные маски от всех квадратов внутривокселя объединяются по "&" и результирующая маска проецируется на экран. По количеству субпикселов в этой маске, как обычно, определяется площадь фрагмента внутри пиксела. Другим немаловажным качеством является фильтрация для удаления артефактов, возникающих в перспективной проекции удаленных от наблюдателя объемных объектов. Дискретизация в глубину приводит к более серьезным и специфическим проблемам: разрыв дискретизации, корректность вычисления оптической глубины, перспективная дисторсия. Перечисленные проблемы разрешаются при помощи следующего метода. Данные трехмерной текстуры хранятся в виде последовательности уровней детальности (трехмерный вариант MIP-Map). В каждой точке

пространства мы можем определить куб, включающий эту точку, ребра которого параллельны осям текстурных координат U, V, W, при этом размер грани куба равен сечению (в данной точке) пирамиды, определяемой точкой наблюдения и рассматриваемым пикселом. При этом координаты U, V, W данной точки пространства дадут нам текстурный адрес, а размер куба позволит определить уровень детальности. Для устранения резких краев объемных блоков и плавной смены уровней детальности применяется квадратичная интерполяция.

7. Архитектура, оборудование

Предполагается, что объемно-ориентированная компьютерная система визуализации будет базироваться на заказных СБИС специального назначения (ASIC) с интеграцией порядка 3М транзисторов и по 300 выводов в микросхеме, в первом типе которых главным вычислительным ядром является конвейер однотипных клеточных процессоров генератора вокселей, другой тип служит профильным назначениям в блоках вычислителя цвета, тумана, текстуры и фильтрации генератора пикселов. Система будет характеризоваться высоким параллелизмом, однородностью и векторизованностью вычислений. Реализация нашего подхода позволяет создать широкий спектр систем для разных применений, начиная с простых (графический акселератор объемной визуализации - одна плата PC стандарта с двумя или тремя кристаллами с ускоренным графическим портом - AGP) - до четырех плат мультимикропроцессорной системы с распределенной памятью. Предполагается возможным использование сигнальных процессоров [19]. Генератор вокселей представляет собой конвейер однотипных процессоров воксельных объемов. Каждый процессор элементарного уровня соединен с памятью списков поверхностей и объемов, принадлежащих данному вокселю текущего уровня. Такой подход позволяет загрузить работой каждый уровень конвейера наиболее полно и выполнить задачу Z-буфера по удалению невидимых поверхностей. Кроме того, каждый уровень конвейера выполняет функцию клипирования, поэтому традиционного клипера как отдельного функционального устройства нет, что привело к однородности структуры и упрощению вычислений.

8. Области применения

Если говорить о перспективах применения систем визуализации, то имеются несколько основных областей.

Тренажеры (космические, авиационные и т.д.). Требуют эффективной работы с большой базой данных, описывающей моделируемую обстановку в реальное время. Алгоритмы корректно решающие задачи данного применения, можно определить как алгоритмы визуализации незамкнутой базы данных. Под незамкнутостью базы данных понимается то, что база данных не входит целиком в поле зрения. Перспективные системы вооружений и военной техники, предъявляют принципиально новые требования к компьютерным системам визуализации (КСВ), используемым в тренажерах для генерации изображения окружающей обстановки [20,21]. К наиболее сложной можно отнести такую задачу, как полет на малых высотах с огибанием горного рельефа местности.

Медицинские системы. Основное требование - достоверность и качество визуализации. Время обновления изображения уже не так критично, база данных замкнута практически всегда (исключая микронавигацию и микрохирургию). Ранее в распоряжении врачей были лишь рентгеновские снимки, которые дают некоторое представление об исследуемых органах в виде наложения теней на изображения и отличаются плохой контрастностью и отсутствием какой-либо информации о глубине объектов. Использование компьютеров дало возможность развиваться новым направлениям томографической интроскопии, таким как компьютерная томография, магнитная резонансная томография, позитронная эмиссионная томография, и др. С помощью томографической аппаратуры можно получить снимки множества сечений тела пациента, которые характеризуют особенности его анатомии, причем с большой четкостью и без наложения изображений органов друг на друга, это важно для таких клинических задач, подобных хирургическому планированию, где необходимо понимать 3D структуру во всей ее сложности и видеть дефекты. Ряд проблем препятствует более широкому использованию методов визуализации объема в медицине и одна из них состоит в огромном количестве вычислений (сотни Тфлопс для рендеринга объема с высоким разрешением).

Научная визуализация. Во многом подобна медицинской, поэтому характер требований почти такой же. Особое значение приобретает вопрос классификации и разделения значений (какими цветами и типами материалов лучше визуализировать, например, напряженность электрического поля? - это вопросы составления цветовой палитры для научной визуализации на основе понятия фундаментального цветового пространства), а также, предобработка данных, как и в задаче компьютерной томографии - восстановления функции поглощения по совокупности ее

линейных интегралов. Ценность любой теории проверяется тем, насколько она приложима к решению различных классов практически важных задач. Визуализация превращает данные численного моделирования в легко интерпретируемые зрительные образы и позволяет быстро просмотреть и оценить в целом результаты моделирования.

9. Заключение

Проведенные исследования по объемно-ориентированной технологии визуализации создают предпосылки для нового класса компьютерных систем визуализации широкого применения. Перечислим основные достоинства описанного метода.

- 1. Существенно уменьшается количество поверхностей для отображения криволинейных объектов. Задание объектов поверхностями произвольных форм сокращает в 100 и более раз описание баз данных по сравнению с заданием их полигонами.
- 2. Существенно облегчается коррекция дисторсии. Отметим, что в системе синтеза с дисторсией изображения, неплоских поверхностей большинство, так как в результате искажения экраном плоских поверхностей они оказываются кривыми, в то время как кривая поверхность уже вряд ли станет плоской.
- 3. Существенно облегчается работа геометрического процессора и уменьшается поток данных из него в видеопроцессор.
- 4. Решаются практически все проблемы с отображением фотореалистичного рельефа местности по карте высот [22-26] без первичной триангуляции.

Недостатком традиционного способа представления рельефа гранями является то, что требуется большое количество граней, из-за чего возникают проблемы высокой глубинной сложности, отбрасывания задних граней, проблема приоритетов, определения и смены уровней детальности, клипирования многоугольников пирамидой видимости. В предлагаемом методе этих проблем нет. В геометрическом процессоре обрабатывается всего одна подстилающая плоскость. Правильный приоритетный порядок обеспечивается соответствующим обходом дерева и системой масок. Обратная поверхность рельефа отбрасывается автоматически. Клипирование рельефа пирамидой видимости становится ненужным, так как в процессе растривания автоматически обеспечивается выборка высот из карты высот, попавших в пирамиду видимости. Для смены уровней детальности используется тот же механизм, что и для обычной текстуры [27].

- 5. Время вычисления при отображении рельефа местности слабо зависит от разрешения карты высот.
- 6. Простота анимации и деформации объектов и поверхностей.
- 7. Возможность визуализации объемов.
- 8. Широкий спектр возможных областей применения.

Данный метод визуализации объемов имеет преимущества от упорядочения в пространстве изображения (быстрая остановка луча) и от упорядочения в объектном пространстве (регулярность обхода объема). Кроме того, в нем элегантно сочетаются возможности визуализации произвольных поверхностей и неоднородных объемных областей.

ЛИТЕРАТУРА

1. **Pfister H.** A real-time volume rendering architecture using an adaptive resampling scheme for parallel and perspective projections // Proc. of 1998 IEEE Symposium on Volume Visualization. Research Triangle Park, North Carolina, 1998. P. 31.
2. **Malzbender T.** Fourier Volume Rendering // ACM Trans. on Gr. 1993. **12**, N3.
3. **Vežina G., Fletcher P., Robertson P. K.** Volume Rendering on the MasPar MP-1 // Proc. of 1992 Workshop on Volume Visualization. Boston, MA, 1992.
4. **Schroeder P., Salem J.B.** Fast Rotation of Volume Data on Data Parallel Architecture // Proc. of Visualization'91, San Diego, 1991. P. 50.
5. **Westover L.** Footprint Evaluation for Volume Rendering // Computer Graphics 1990. **24(4)** P. 37.
6. **Fuchs H., Poulton J., Eyles J.** et al. Pixel-Planes 5: A heterogeneous multiprocessor graphics system using processor-enhanced memories // Computer Graphics 1989. **23**. N. 3, P. 79.
7. **Fuchs H.** Direct visualisation of Volume Data. Volume-based Anatomical Application // IEEE CG & Applications. 1992. **12(4)**. P. 63.
8. **Lacroute P.** Real-Time Volume Rendering on Shared Memory Multiprocessors Using the Shear-Warp Factorization // Proc. 1995 Parallel Rendering Symposium, Atlanta, Georgia, 1995. P. 15.
9. **Amin A.B., Grama A., Singh V.** Fast Volume Rendering Using an Efficient, Scalable Parallel Formulation of the Shear-Warp Algorithm // Proc. 1995 Parallel Rendering Symposium, Atlanta, Georgia, 1995, P. 7.
10. **Neih J., Levoy. M.** Volume Rendering on Scalable Shared Memory Architecture // Proc. of 1992 Workshop on Volume Visualization, Boston, MA, 1992. P. 17.

11. **Montani C., Perego R., Scopingo R.** Parallel Volume Visualization on a Hypercube Architecture // Proc. of 1992 Workshop on Volume Visualization, MA, 1992. P. 9.
12. **Corrie B., Mackerras P.** Parallel Volume Rendering and Data Coherence on the Fujitsu AP1000 // Technical Report TR-CS-92-11, Department of Computer Science, Australian National University, Canberra, ACT, Australia, 1992.
13. **Law A., Yagel R.** Exploiting Spatial, Ray, and Frame Coherency for Efficient Parallel Volume Rendering // GraphiCon'96, St.Petersburg, 1996.
14. **Law A., Yagel R.** CellFlow: A Parallel Rendering Scheme for Distributed Memory Architectures // Proc. of International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'95), 1995, P.1.
15. **Yagel R., Machiraju R.** Data Parallel Volume Rendering Algorithms // The Visual Computer. 1995. **11(6)**. P. 319.
16. **Knittel G.** Voxel Engine for Real-time Visualization and Examination // Eurographics'93. **12**.
17. **Вяткин С.И., Долговесов Б.С. Чижик С.Е.** Синтез виртуальной среды с рекурсивным делением объектного пространства // Графикон '98 М. 1998.
18. **Ефимов Н. В.** Высшая Геометрия. М.: Наука, 1978, С. 324-338.
19. **Savchenko V.V., Pasko A.A. Vyatkin S.I. et al.** New approach in geometric modeling: distributed and hardware implementation perspectives.// International Conference on Computers and Devices for Communication CODEC-98. Calcutta, India, 1998. P. 285.
20. **Cosman M.A., Mathisen A.E., Robinson J.A.** A new visual system to support advanced requirements // IMAGE V Conference Phoenix, USA, 1990.
21. **Cosman M.A.** Mission Rehearsal Modeling // IMAGE VI Conference Scottsdale, Arizona 14-17 1992.
22. **Cohen D., Shaked A.** Photo-realistic Imaging of Digital Terrains // Proc. of Eurographics'93, UK, 1993. P. 363.
23. **Paglieroni D.W., Petersen S.M.** Parametric Heights Fields Ray Tracing // Proc. of Graphics Interface'92. 1992. P. 192.
24. **Vežina G., Robertson P.K.** Terrain Perspectives on a Massively Parallel SIMD Computer // Proc. of CG International'91. 1991. P. 163
25. **Pitot P., Duthen Y., Caubet R.** A parallel architecture for ray-casting // Computer Graphics '89. 1989. P. 463.
26. **Agranov G., Gotsman C.** Algorithms for Rendering Realistic Terrain Image Sequences and Their Parallel Implementation // Proc. of Graphicon'95. 1995. P. 153.

27. **Vyatkin S.I., Dolgoesov B.S., Ovechkin V.V. et al.** Photorealistic imaging of digital terrains, freeforms and thematic textures in realtime visualization system Voxel-Volumes // GraphiCon'97. Moscow.

Авторы:

Сергей Иванович Вяткин вед. инж.
Борис Степанович Долговесов к. т. н., зав. лаб.
Александр Владимирович Есин инж.
Александр Александрович Жигач инж.
Сергей Евгеньевич Чижик вед. инж.
Руслан Алексеевич Щербаков инж.

Адрес: 630090, Новосибирск, пр. Каптюга, 1
Телефон: (3832) 333 630
Email: bsd@iae.nsk.su

FREE FORMS AND VOXEL VOLUMES REAL-TIME VISUALIZATION

Sergei I. Vyatkin, Boris. S. Dolgoesov,
Alexander V. Yesin, Alexander A. Zigach,
Sergei E. Chizhik, Ruslan A. Scherbakov
Institute of Automation and Electrometry,
Siberian Division of Russian Academy of
Sciences
Computer-based Visualization Systems
Laboratory*

Virtual reality systems, where viewer merges into the model world, require high quality visualization, for example, on the level provided by Onyx2 Reality Monster by Silicon Graphics, priced around 100 thousand dollars. In addition there is a number of application area specific problems, for example, dynamic distortion correction, which is required for image projection to non-planar screens taking into account viewer and projection device movements. These problems are solved in high-end real-time visualization systems ESIG and HARMONY of Evans & Sutherland).

More recently, voxel-based methods have been developed which create a 3D view directly from the volume data. For the first time on PC-class computers, the vg500 of Mitsubishi Electric allows high quality, real time volume rendering (30 frames/sec.). The vg500 computes 500 million voxels per second with associated 3D gradient estimations.

The surface-based methods first create on intermediate surface representation of the object to be shown. Both surface-and voxel-based methods have their merits; the decision which one should be used for a particular application depends on the visualization goals. We are present a new project on volume modeling and rendering that combines volume representations by voxel data and by real continuous functions.

Image generators traditionally use polygons as database primitives. It is difficult to use polygons to

make convincing models of clouds, smoke, trees and anything else for which no simple surface representation exists. Research system that uses voxels as the database primitive has the complementary behavior. An image generator architecture that could process both surfaces and voxels in an elegant way would have the advantages of both approaches. We suggest expanding the notion of primitives and making it possible to process them by easy and effective method without approximation by polygons.

The analysis of possible directions of evolution of a real-time visualization systems shows that the easiest way to improve picture quality, i.e. to increase number of polygons rendered per frame, is not the most effective one. Going this way, the qualitative changes can be hardly achieved. For instance, introduction of texture was such a change. Even displaying much more polygons, an image without textures will be poorer than an image with fewer polygons but with color textures. Moreover, since Phong shading renders the higher quality with a much smaller number of polygons, one needs much less computing power for geometry processing; consequently, the overall cost of the Phong-based system is significantly less than that of a comparable Gouraud-based system. Thus, changing polygon "quality" holds more promise for evolution of visualization systems rather than increasing polygon quantity.

Nevertheless imaging realistic mountains and terrain in general requires big number of polygons. The same is with sculptural surfaces. Even if NURBS (non-uniform rational B-splines) or Bezier patches are applied, a comparatively big number of them is required. All this proves need for processing a lot of primitives in visualization systems. Existing algorithms work either with polygonal or with polynomial representations of geometry. Surfaces of high order are approximated by polygons or patches and these approximations again require lots of primitives. On the other side using polynoms of high order is not convenient due to complexity of computations and accuracy limitations.

In this article a technique for rendering free form surfaces and volumes is proposed, which provides quality impossible with polygonal representation even with very big number of polygons. From a practical point of view an versatile system providing solution of wide range of tasks and minimal replacement of modules for application area specific tasks is the most efficient.