УДК 004.92

С.И. ВЯТКИН,
Институт автоматики и электрометрии СО РАН
А.Н. РОМАНЮК, О.Н. РЕЙДА, О.В. РОМАНЮК
Винницкий национальный технический университет

МЕТОД РЕНДЕРИНГА СЛОЖНЫХ ПОЛИГОНАЛЬНЫХ СЦЕН С ПРИМЕНЕНИЕМ ФУНКЦИОНАЛЬНО ЗАДАННЫХ ОБЪЕКТОВ

В последнее время возможности аппаратной визуализации существенно возросли. Однако, обработка сложных сцен по-прежнему является одной из самых фундаментальных проблем в компьютерной графике. При этом важно генерировать сложные графические объекты с приемлемой производительностью, достаточной для динамического и интерактивного режимов.

Представлен метод рендеринга сложных сцен. Основная идея метода заключается в генерации изображения сцены путем ее реконструкции из динамически выбранного набора случайных точек выборки поверхности. Визуализируется изображение сложной трехмерной сцены, состоящей из примитивов – треугольников, с помощью реконструкции из динамически выбранного массива точек поверхностей. Опорные точки представляют сложную геометрию сцены, поэтому не каждый треугольник должен обрабатываться отдельно во время рендеринга. На первом этапе выбираются точки случайной выборки таким образом, чтобы они покрывали проекции объектов в плоскости изображения приблизительно равномерно. На втором этапе, метод реконструирует видимость между выбранными опорными точками.

Метод не зависит от связности сетки и топологии. В результате время рендеринга растет только логарифмически от количества треугольников в сцене. Автоматическое определение сцены с низкой детализацией обеспечивает высокую скорость рендеринга. Предварительно вычисленные структуры данных позволяют интерактивно делать динамическое обновление сцены.

Для описания поверхностей используются функции отклонения (второго порядка) от базовой квадрики. Для формирования моделей сложных объектов на базе функций возмущения используются теоретико-множественные операции объединения и пересечения, осуществляемые с применением булевых функций.

Требования к памяти растут линейно с количеством треугольников. Схема инсталляции на основе графа сцены создает условия для дальнейшего снижения объема памяти. Для увеличения сложности сцен добавлены функционально заданные объекты.

Предложенный метод может быть эффективно использован в системах высоко реалистической компьютерной графики.

Ключевые слова: рендеринг; полигоны; уровни детализации; функционально заданные объекты; графические процессоры.

С.И. ВЯТКИН,
Інститут автоматики та електрометрії СО РАН
О.Н. РОМАНЮК, О.М. РЕЙДА, О.В. РОМАНЮК
Вінницький національній технічний університет

МЕТОД ВІЗУАЛІЗАЦІЇ СКЛАДНИХ ПОЛІГОНАЛЬНИЙ СЦЕН З ВИКОРИСТАННЯМ ФУНКЦІОНАЛЬНО ЗАДАНИХ ОБ'ЄКТІВ

Останнім часом можливості апаратної візуалізації істотно зросли. Однак, обробка складних сцен є однією з найбільш фундаментальних проблем в комп'ютерній

https://doi.org/10.32782/KNTU2618-0340/2020.3.2-2.5

графіці. При цьому важливо генерувати складні графічні об'єкти з прийнятною продуктивністю, достатньої для динамічного та інтерактивного режимів.

Представлено метод візуалізації складних сцен. Основна ідея методу полягає в генерації зображення сцени шляхом її реконструкції з динамічно обраного набору випадкових точок вибірки поверхні.

Візуалізується зображення складної тривимірної сцени, що складається з примітивів – трикутників, за допомогою реконструкції з динамічно обраного масиву точок поверхонь. Опорні точки представляють складну геометрію сцени, тому не кожен трикутник повинен оброблятися окремо під час рендеринга. На першому етапі вибираються точки випадкової вибірки таким чином, щоб вони покривали проекції об'єктів в площині зображення приблизно рівномірно. На другому етапі, метод реконструює видимість між обраними опорними точками.

Метод не залежить від зв'язності мережі та топології. У результаті час рендерингу зростає тільки логарифмічно від кількості трикутників у сцені. Автоматичне визначення сцени з низькою деталізацією забезпечує високу швидкість рендерингу. Попередньо обчислені структури даних дозволяють інтерактивно робити динамічне оновлення сцени.

Для опису поверхонь використовуються функції відхилення (другого порядку) від базової квадрік. Для формування моделі складних об'єктів на базі функцій збурення використовуються теоретико-множинні операції об'єднання і перетину, реалізовані з застосуванням булевих функцій.

Вимоги до пам'яті ростуть лінійно з кількістю трикутників. Схема інсталяції на основі графа сцени створює умови для подальшого зниження об'єму пам'яті. Для збільшення складності сцен додані функціонально задані об'єкти. Запропонований метод може бути ефективно використаний у системах високо реалістичної комп'ютерної графіки.

Ключові слова: рендеринг; полігони; рівні деталізації; функціонально задані об'єкти; графічні процесори.

S.I. VYATKIN,
Institute of Automation and Electrometry of SB ASR
O.N. ROMANYUK, O.V. ROMANYUK
Vinnitsa National Technical University

METHOD OF RENDERING COMPLEX POLYGONAL SCENES WITH APPLICATION OF FUNCTIONALLY SPECIFIED OBJECTS

Recently, the capabilities of hardware visualization have increased significantly. However, handling complex scenes is still one of the most fundamental problems in computer graphics. At the same time, it is important to generate complex graphic objects with acceptable performance sufficient for dynamic and interactive modes.

A method for rendering complex scenes is presented. The main idea of the method is to generate a scene image by reconstructing it from a dynamically selected set of random surface sampling points.

The image of a complex three-dimensional scene consisting of primitives - triangles is visualized using reconstruction from a dynamically selected array of surface points.

Anchor points represent the complex geometry of the scene, so not every triangle needs to be processed separately during rendering. At the first stage, random sampling points are selected so that they cover projections of objects in the image plane approximately uniformly. In the second step, the method reconstructs the visibility between the selected anchor points.

https://doi.org/10.32782/KNTU2618-0340/2020.3.2-2.5

The method is independent of grid connectivity and topology. As a result, the rendering time grows only logarithmically from the number of triangles in the scene. Automatically detecting scenes with low detail provides high-speed rendering. Precomputed data structures enable interactive dynamic scene updating.

To describe the surfaces, the deviation functions (second order) from the base quadric are used. To form models of complex objects based on perturbation functions, set-theoretic union and intersection operations using Boolean functions are used.

Memory requirements grow linearly with the number of triangles. The installation diagram based on the scene graph creates the conditions for a further reduction in memory. Functional objects were added to increase the complexity of scenes.

The proposed method can be effectively used in highly realistic computer graphics systems.

Keywords: rendering; polygons; levels of detail; functionally defined objects; GPUs.

Постановка проблемы

В последнее время возможности аппаратной визуализации существенно возросли. Однако, обработка сложных сцен по-прежнему является одной из самых фундаментальных проблем в компьютерной графике. Интерактивное отображение сложных сцен, является серьезной проблемой для алгоритмов рендеринга.

Анализ последних исследований и публикаций

В работе [1] рассмотрен алгоритм рендеринга, адаптированный к выходным данным. Его временная сложность слабо зависит от сложности сцены на входе. Классические алгоритмы, такие как алгоритм z-буфера, не соответствуют этому требованию, потому что их время рендеринга растет линейно с числом элементарных объектов в сцене. Несмотря на аппаратные реализации, эти алгоритмы не способны отображать очень сложные сцены в режиме реального времени.

Большое количество подходов было предложено для того, чтобы достичь максимума сложности сцен в интерактивных приложениях. Был предложен метод мульти-разрешения при моделировании сцен. В этом подходе объекты сцены хранятся на разных уровнях детализации. Соответствующий уровень детализации выбирается во время рендеринга в соответствии с метрической ошибкой, зависящей от точки наблюдения. Уровни детальности могут быть созданы автоматически с помощью алгоритмов упрощения треугольных сеток [2]. Однако некоторые сцены, такие как сцены сложной топологии, не могут быть упрощены без ущерба их общего внешнего вида.

Рендеринг на основе изображений – это другой подход к уменьшению сложности сцены. Этот метод использует производные представлений фиксированной сложности из графических данных для объектов более высокой сложности. Большое число подходов было предложено. Их можно разделить на две группы.

Первая группа использует предварительно вычисленные структуры данных [3–5]. Они отличаются друг от друга использованием геометрической информации и требованиями к памяти. Ограничение всех этих методов заключается в том, что максимальное качество изображения ограничено разрешением данных, которое фиксировано заранее. Также нельзя сделать эффективное динамическое изменение сцены.

Вторая группа изображений на основе структуры данных получает свое содержимое динамически, подобно стратегии кэширования. В работах [6–7] кэшированные данные получены путем представления части сцены с использованием обычного алгоритма Z-буфера. Использование точечных образцов для визуализации

https://doi.org/10.32782/KNTU2618-0340/2020.3.2-2.5

моделей поверхностей было описано в [8]. Такие примитивы применялись также для аморфных объектов в публикациях [9–11]. В работе [12] применялся рендеринг сложных сцен с использованием блоков небольшого проецируемого размера, взятых из пространственной иерархии. В работе [13] описан рендеринг сложного объекта из точечных образцов, полученных из ортогональных представлений объекта с использованием иерархического алгоритма для реконструкции изображения. Два последних метода используют точечный образец с несколькими разрешениями. В работе [14] описывается метод рендеринга для быстрого отображения больших сеток от данных сканера. Ограничение иерархии сфер строится на основе точек выборки поверхности, хранящих среднее значение атрибут, таких как нормали и цвета в каждом узле. В работе [15] описан метод, в котором преобразуется геометрия сцены в представление точечного образца путем построения иерархии слоистых изображений по глубине [4]. То есть, для каждого узла восьмеричного дерева, точечные выборки генерируются с помощью трассировки лучей объектов в фиксированном разрешении из трех ортогональных направлений. Для каждой точки образца, хранятся атрибуты положения и поверхности. Во время рендеринга происходит интерполяция между смежными уровнями восьмеричного дерева для выполнения сглаживания. Недостатком обоих методов является то, что представление точечной выборки, использованной для отображения, должно быть построено на этапе подготовки для фиксированного разрешения. Качество изображения, доступное после этого, всегда ограничено максимальной плотностью выборки генерируемых точечных образцов. Для того, чтобы получить хорошее качество изображения с близкого расстояния необходимо затратить большой объем памяти. Это особенно сложно, если уровень детализации меняется.

Цель исследования

Цель работы состоит в разработке метода рендеринга с учетом выходных данных. Основная идея метода заключается в генерации изображения сцены путем ее реконструкции из динамически выбранного набора случайных точек. Опорные точки представляют сложную геометрию сцены, поэтому не каждый треугольник должен обрабатываться отдельно во время рендеринга. На первом этапе выбираются точки случайной выборки таким образом, чтобы они покрывали проекции объектов в плоскости изображения приблизительно равномерно. На втором этапе, метод реконструирует видимость между выбранными опорными точками.

Изложение основного материала исследования

Разработка метода

Предположим, что задано множество выборочных точек, которые были выбраны случайно, независимо, и равномерно распределены на проекции объектов на плоскости Предположим, что сцена S, состоящая из N треугольников, задана вместе с функцией затенения, которая определяет цвет для каждой точки в S и которая может быть оценена как O (1). Функция затенения может зависеть от текущей точки обзора и глобальной настройки освещения. Это включает в себя, например, затенение Фонга, текстуры и карты окружающей среды. Для каждого кадра дается плоская перспективная проекция. Метод рендеринга состоит из двух основных шагов: выбор опорных точек и реконструкция изображения.

Эффективный алгоритм выбора опорных точек – это ключ к достижению интерактивной частоты кадров. Реализован метод, который эффективно генерирует точечный образец представления сцены на лету, в соответствии с данными наблюдателя. Опорные точки должно быть равномерно распределены по проекциям объектов в плоскости изображения.

https://doi.org/10.32782/KNTU2618-0340/2020.3.2-2.5

На этапе предварительной обработки, сцена подразделяется на иерархию групп треугольников, которые показывают аналогичный коэффициент масштабирования во время проекции. Для каждой точки зрения, выбирается динамически подходящий набор групп из иерархии. Оценивается максимальный коэффициент проекции для определения количества выбранных опорных точек из каждой группы. В большинстве случаев достаточно рассматривать только расстояние между группой и зрителем для того, чтобы оценить коэффициент проекции. В особых случаях лучшие результаты могут быть достигнуты за счет дополнительного учета ориентации треугольников. Точки выборки затем выбираются в пределах группы по проекционной площади треугольников, используя предварительно вычисленные структуры поиска. Можно показать, что это приближение не вредит качеству изображения и времени извлечения группы из иерархии, т. е. сцены с равномерно распределенными векторами нормалей поверхности.

После выбора набора опорных точек для группы объектов, они хранятся в кэше образца, для того, чтобы уменьшить затраты. Идентифицируются треугольники с большими проекционными областями. Только треугольники с небольшой площадью проекции будут заменены точками выборки. Эта стратегия гарантирует, что скорость рендеринга не снизится.

Далее происходит реконструкция изображения. Необходимо восстановить изображение от выборочных точек. Закрытые точки должны быть удалены. После этого изображение получается интерполяцией между видимыми точками выборки. Если плотность образца достаточно высока, так что каждый пиксель в изображении получает примерно точку из объекта переднего плана, сцена будет реконструирована правильно.

Используя метод сплаттинга [15], реконструкция может быть отрегулирована как в качестве, так и в скорости. Использование большого сплатта постоянного цвета и глубины ускоряет ход реконструкции. Качественные результаты могут быть получены путем взвешенного усреднения видимых опорных точек, с использованием гауссовых фильтров. Шума и артефактов при сглаживании можно избежать за счет более длительного, а не реального, времени реконструкции.

Метод включает реконструкцию на едином уровне детализации для всех частей изображения. Метод позволяет иметь гибкий компромисс между качеством изображения и скоростью восстановления.

По-пиксельный алгоритм восстановления принимает все точки выборки в произвольном порядке. Проецирует их на плоскость изображения и генерирует их как пикселы, используя z-буфер для того, чтобы решить видимость. Предполагаем, что каждый пиксель полностью покрыт треугольниками. Изображение считается правильным, если каждый пиксель изображения показывает цвет, который можно найти на любом фрагменте треугольника через этот пиксель. Это определение правильности немного отличается от алгоритмов рендеринга, таких как рей-трэйсинг или Z-буферизация с субпикселями.

Чтобы получить правильное изображение при по-пиксельной реконструкции, каждый пиксель должен получить хотя бы один образец из области переднего плана. Дополнительно, одна из точек переднего плана должна иметь наименьшее значение глубины среди всех опорных точек. Так, что это приведет к перезаписи всех остальных опорных точек в z-буфер и воспроизведет правильно окрашенный пиксель.

Определим количество опорных точек, которые необходимо выбрать, чтобы гарантировать, что каждый пиксел получает хотя бы одну точку из фрагмента треугольника. Во-первых, предполагаем, что на изображении нет скрытых поверхностей. Поскольку каждый пиксель по определению полностью покрыт видимой поверхностью, можно рассматривать простую модель. Для учета окклюзий моделируем

https://doi.org/10.32782/KNTU2618-0340/2020.3.2-2.5

процесс как два шага случайного эксперимента: пусть H-проекция и закрытые поверхности измеряются в пикселях. Формула $A := V+h$ – это общее значение объема проектируемой площади (эти значения будут оцениваться далее). Закрытая область представлена h дополнительными ячейками, которые выбираются с той же вероятностью. Используя нормальное приближение в биномиальном распределении, можно показать, что $A \cdot \ln v + O(A)$ образца точек достаточно, чтобы гарантировать, что каждый пиксель получает хотя бы один образец от видимой части объекта с высокой вероятностью. Эксперименты показывают, что абсолютное значение точки выборки – это хороший выбор на практике.

По-пиксельный алгоритм восстановления иногда слишком дорог для запуска в интерактивном режиме. Простой способ снижения затрат на рендеринг – это использовать метод сплаттинга, то есть вместо генерации одиночных пикселей, генерировать более большие квадратичные зоны с длиной стороны D, которые заливаются одинаковым цветом и значением глубины, выполняющем сравнение глубины для каждого пиксела. Это уменьшает размер выборки. Можно охватить все изображения площадью на $\frac{1}{d^2}$. Качество изображения падает с увеличением коэффициента d. Однако, результаты всегда лучше, чем полученные простым уменьшением разрешения на коэффициент d. Значения $d \approx 2..5$ – хороший компромисс. Значение $d=2$ обеспечивает хорошее качество изображения. Даже для сцен с высокочастотными деталями, сохраняя время вычисления около 75%. В то время как более высокие значения могут привести к нежелательным визуальным артефактам. Для получения более качественных реконструкций используется сплаттинг только для удаления скрытых опорных точек. Затем заполняется изображение с использованием средневзвешенного значения соседних видимых точек с помощью весовых функций Гаусса. Необходимо использовать большой размер выборки (примерно в десять раз больше размера пикселя реконструкции), чтобы избежать низкочастотные шумы, которые подчеркиваются фильтрованием.

Теперь рассмотрим алгоритм эффективного выбора опорных точек. Для эффективного решения нет необходимости гарантии точного и равномерного распределения точек выборки по объектам на плоскость изображения. Реконструкция изображения будет правильной, если части изображения содержат слишком много точек выборки. Они просто вызывают дополнительные расходы на обработку. Необходимо только обеспечить что плотность образца нигде не опускается ниже идеальной плотности отбора проб. Время работы алгоритма реконструкции будет увеличено пропорционально завышению прогнозируемой площади. Получим функцию плотности вероятности для выбора выборочных точек. Рассмотрим бесконечно малый фрагмент поверхности s и вычислим коэффициент проекции, $\mathrm{prj}(s)$, по которому он масштабируется при проецировании на плоскость изображения. Коэффициент проекции для такого фрагмента поверхности s задается z-расстоянием до центра проекции, измеренным ортогонально к плоскости проекции. Начинаем описание процесса отбора образцов, предполагая, что все значения площади проекции треугольников известны заранее. Выбираем случайные точки выборки в соответствии с вероятностью функции плотности f за два шага. Сначала выбирается треугольник с вероятностью, пропорциональной его прогнозируемой площади. Во-вторых, случайная точка на треугольнике выбирается с помощью случайной линейной комбинации его вершин. Случайный треугольник может быть выбран эффективно с помощью списков распределения. Список распределения строится для треугольников, поступающих в произвольном порядке, вставки указателя на треугольник и дополнительного значения площади треугольника в списке. Начинаем со спроецированной площади первого

https://doi.org/10.32782/KNTU2618-0340/2020.3.2-2.5

треугольника, и увеличиваем значение суммированной площади с помощью площади проецирования каждого вставленного треугольника. Таким образом, получаем функцию, пропорциональную дискретному распределению функции вероятности выбора треугольников. Чтобы выбрать случайный треугольник, определим случайное число. Для этого число выбирается из интервала [0,1] с использованием стандартного псевдо-генератора случайных чисел с равномерным распределением. Масштабируем значение посредством общей суммы области проецирования, а затем выполняем поиск первой записи в списке с помощью значения суммированной области, которое больше случайного значения. Алгоритм выбирает треугольники в соответствии с заданной функцией распределения по суммированным значениям площадей проецирования. Результат статистики обратной функции распределения применяется к случайной переменной, равномерно распределенной в интервале [0, 1]. При двоичном поиске инверсия дискретной функции распределения, заданной списком распределения, может быть выполнена эффективно. Используя этот метод, выбирается выборочная точка в течение $O\log_2 n$ времени для n треугольников, если значения проецируемых площадей всех треугольников известны. Для оценки значений площади проецирования сцена делится на группы треугольников, показывающих аналогичный коэффициент проекции. Группы выбираются динамически от предвычисленной иерархии групп треугольников. Для каждой выбранной группы, определяется количество опорных точек, которые она должна получить. Определяется с помощью произведения проекции площади всех треугольников в группе и верхней границы для проекционного фактора этой группы. Это гарантирует, что плотность выбора не падает ниже минимальной плотности образца. Для каждой группы список рассылки хранится в соответствии с проекционной площадью треугольников, используемых для выбора опорных точек. Разделение сцены на группы должно связывать три фактора, так как суммарный коэффициент проекции является произведением трех независимых величин: коэффициента глубины $\frac{1}{z^2}$, коэффициента ориентации $\cos\alpha$ и искажения в направлении границ изображения $1/\cos\beta$. Влияние искажения на фактор проекции довольно мало. Для типичного значения максимального угла зрения оно может быть проигнорировано. Остается связать лишь два других фактора. Строим восьмеричное дерево для треугольников сцены на шаге предварительного вычисления. Каждый узел дерева имеет габаритный ящик. Треугольники, которые пересекают сетку восьмеричного дерева, хранятся в наименьшем внутреннем узле, и имеют габаритный ящик, который полностью содержит треугольник. Каждый узел восьмеричного дерева содержит список распределения значений не проецируемых площадей всех треугольников, содержащихся в этом узле. Внутренние узлы получают свой список в виде объединения дочерних списков и списков собственных треугольников в фиксированном порядке. В общей сложности необходим только один список, хранящий все значения суммированной области. Узлы размечают свою часть списка с помощью указателей на их участках. Таким образом, общий объем хранящихся данных составляет O (n), где n - число треугольников. Предварительное вычисление может быть выполнено за время $O(n\log_2 n)$.

Когда наблюдатель перемещается в новое положение, выбирается массив восьмеричного дерева. Алгоритм рекурсивно пересекает восьмеричное дерево от корня. Рекурсия выбирает все пересеченные узлы, для которых отношение между минимальным и максимальным коэффициентами глубины не больше заданного постоянного значения. Можно показать, что время обхода восьмеричного дерева определяется соотношением между наименьшим и максимально возможным значением

https://doi.org/10.32782/KNTU2618-0340/2020.3.2-2.5

глубины в сцене. Таким образом, оно всегда ограничено отношением расстояния передней плоскости отсечения к диаметру сцены. Логарифмический рост числа доменов в дереве позволяет обрабатывать очень большие сцены без значительных затрат на обработку поля.

Динамическая версия структуры данных для выборки может быть получена путем применения некоторых незначительных изменений к статическому варианту данных. Основная проблема в том, что список рассылки хранения накопленных значений площади не может быть эффективно обновлен. Можно заменить динамически сбалансированное дерево поиска для списка рассылки. Это приводит к динамическому времени обновления $O(t)$ с t, обозначающим высоту пространственного восьмеричного дерева при сохранении времени выборки $O\log_2 n$, как в статическом случае. Если сцена состоит из однородных объектов, распределенных, по крайней мере, в одном пространственном измерении, накладные расходы на поддержание дополнительного динамически сбалансированного дерева поиска могут быть незначительными. Для таких сцен само восьмеричное дерево относительно хорошо сбалансировано, и его можно использовать как «дерево распределения».

Функционально заданные объекты

Для описания поверхностей используются функции отклонения (второго порядка) от базовой квадрики [16]. Функция задается алгебраическим неравенством второй степени с тремя неизвестными x, y, z в виде $F(x, y, z) \geq 0$. Поверхности рассматриваются как замкнутые подмножества Евклидова пространства, определяемые описывающей функцией $F(x, y, z) \geq 0$. Где F – непрерывная вещественная функция; x, y, z – задаваемая координатными переменными точка в E^3. Здесь $F(x, y, z) > 0$ задает точки внутри поверхности, $F(x, y, z) = 0$ – точки на границе и $F(x, y, z) < 0$ – точки, лежащие снаружи и не принадлежащие поверхности.

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

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

где x, y и z – пространственные переменные.

Можно записать это неравенство в матричном виде:

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

Поверхности генерируются с помощью квадрик и представляются композицией базовой квадрики и возмущений:

$$F'(x, y, z) = F(x, y, z) + \sum_{i=1}^{N} f_i R_i(x, y, z), \quad (3)$$

где fi- форм-фактор; $R(x, y, z)$ – возмущение:

https://doi.org/10.32782/KNTU2618-0340/2020.3.2-2.5

$$R_i(x,y,z)=\begin{cases} Q_i^3(x,y,z),\ if\ \ Q_i(x,y,z)\ge 0; \\ 0,\ if\ \ Q_i(x,y,z)<0; \end{cases} \tag{4}$$

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

Геометрическая модель создаёт условия для конструирования объектов и их композиций различной сложности. Для этого используется множество геометрических операций *Ф*, определяемое математически следующим образом [16]:

$$\Phi_j : M^1 + M^2 + ... + M^n \to M\ . \tag{5}$$

Для формирования моделей сложных объектов на базе функций возмущения используются теоретико-множественные, осуществляемые с применением булевых операций объединения и пересечения. Бинарная операция (n=2) (5) объектов G_1 и G_2 означает операцию $G_3 = \Phi_j(G_1, G_2)$ с определением:

$$f_3 = \psi(f_1(x,y,z),\ f_2(x,y,z)) \ge 0\,, \tag{6}$$

где ψ – непрерывная вещественная функция двух переменных.

Выводы

Так как метод генерирует опорные точки «на лету» из первоначального геометрического описания сцены, сцену можно отображать с произвольных точек зрения. Из-за накладных расходов генерация опорных точек «на лету» выполняется медленнее, чем рендеринг предварительно вычисленных наборов точек. Используется схема кэширования, чтобы избежать этих накладных расходов. Наборы выборок с низким коэффициентом передискретизации создаются динамически. Они могут быть повторно использованы для нескольких кадров. Предлагаемое представление сцены позволяет динамические ввод и удаления объектов. В этой работе представлен метод рендеринга с учетом выходных данных. Основная идея метода заключается в генерации изображения сцены путем ее реконструкции из динамически выбранного набора случайных точек выборки поверхности. Опорные точки представляют сложную геометрию сцены, поэтому не каждый треугольник должен обрабатываться отдельно во время рендеринга. Логарифмический рост времени рендеринга позволяет отображать сцены высокой сложности (рис. 1).

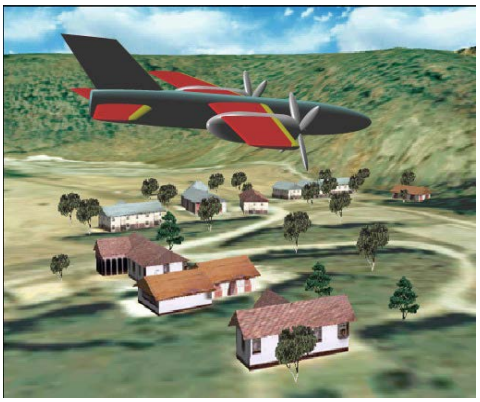

Рис. 1. Сцена, состоящая из 5×10^6 треугольников и функционально заданного объекта.

https://doi.org/10.32782/KNTU2618-0340/2020.3.2-2.5

Метод реализован на C++, используется OpenGL с расширениями nVidia для рендеринга. Метод был реализован с применением центрального процессора Intel Core2 CPU E8400 3.0 GHz и графического процессора GPU 470 GTX.

Список использованной литературы

1. Вяткин С. И., Романюк О. В., Обидник М. Д. Особенности полигонального моделирования в системах компьютерной графики. *Современные проблемы радиоэлектроники, телекоммуникаций и приборостроения СПРТП–2009*: материалы IV Международной научно-технической конференции. (Винница, 8–10 октября 2009 р.). Винница, 2009. С. 24.
2. Вяткин С. И., Романюк А. Н., Костюкова Н. С. Базы данных и моделирующие комплексы для систем визуализации реального времени. *Информационные технологии и информационная безопасность в науке, технике и образовании*: материалы Международной научно-практической конференции. (Севастополь, 5–10 сентября 2011 г.). Севастополь, 2011. С. 156–157.
3. Gortler S., Grzeszczuk R., Szeliski R., Cohen M. The Lumigraph. Proceedings of the *SIGGRAPH '96*: 23rd Annual Conference on Computer Graphics and Interactive Techniques. (New Orleans, August 4-9, 1996). New York: Association for Computing Machinery, 1996. P. 43–54.
4. Shade J., Gortler S., He L, Szeliski R. Layered Depth Images. Proceedings of the *SIGGRAPH '98*: 25th Annual Conference on Computer Graphics and Interactive Techniques. (Orlando, July 19-24, 1998). New York: Association for Computing Machinery, 1998. P. 231–242.
5. Levoy M., Hanrahan P. Light Field Rendering. Proceedings of the *SIGGRAPH '96*: 23rd Annual Conference on Computer Graphics and Interactive Techniques. (New Orleans, August 4-9, 1996). New York: Association for Computing Machinery, 1996. P. 31–42.
6. Shade J., Lischinski D., Salesin D., DeRose T., Snyder J. Hierarchical Image Caching for Accelerated Walkthroughs of Complex Environments. Proceedings of the *SIGGRAPH '96*: 23rd Annual Conference on Computer Graphics and Interactive Techniques. (New Orleans, August 4-9, 1996). New York: Association for Computing Machinery, 1996. P. 75–82.
7. Schaufler G. Per-Object Image Warping with Layered Impostors. Proceedings of the *SIGGRAPH '98*: 25th Annual Conference on Computer Graphics and Interactive Techniques. (Orlando, July 19-24, 1998). New York: Association for Computing Machinery, 1998. P. 145–156.
8. Levoy M., Whitted T. The Use of Points as a Display Primitive. Norh Carolina : University of Norh Carolina at Chapel Hill, 1985. 19 p. (Technical report 85-022).
9. Вяткин С. И. Моделирование неоднородностей при визуализации атмосферных эффектов. *Вестник компьютерных и информационных технологий*. 2016. № 7. С. 9–14.
10. Vyatkin S., Romaniuk S., Romaniuk O. Visualization of 3D-amorphous Objects Using Free Forms. *Электротехнические и компьютерные системы.* 2015. № 19 (95). С. 227−230.
11. Вяткин С. И., Романюк А. Н., Поддубецкая М. П. Анимация трехмерных объектов. *Измерительная и вычислительная техника в технологических процессах*. 2013, № 1 (42). С. 207–211.
12. Vyatkin S. I., Romanyuk A.N., Pavlov S. V., Moskovko M. V., Askarova N., Sagymbekova A., Wojcik W., Kotyra A. Fast Ray Casting of Function-Based Surfaces. *Przeglad Elektrotechniczny*. 2017. № 5. P. 83−86. DOI:10.15199/48.2017.05.16

https://doi.org/10.32782/KNTU2618-0340/2020.3.2-2.5

13. Grossman J., Dally W. Point Sample Rendering. Proceedings of the *Rendering Techniques '98*: Eurographics Workshop in Vienna. (Vienne, June 29 – July 1, 1998). Vienne: Springer, 1998. P. 181–192.
14. Rusinkiewicz S., Levoy M. Qsplat: A Multiresolution Point Rendering System for Large Meshes. Proceedings of the *SIGGRAPH '00*: 27th Annual Conference on Computer Graphics and Interactive Techniques. (New Orleans, July 23-28, 2000). New York: ACM Press/Addison-Wesley Publishing Co., 2000. P. 343–352.
15. Pfister H., Zwicker M., J. van Baar, Gross M. Surfels: Surface Elements as Rendering Primitives. Proceedings of the *SIGGRAPH '00*: 27th Annual Conference on Computer Graphics and Interactive Techniques. (New Orleans, July 23-28, 2000). New York: ACM Press/Addison-Wesley Publishing Co., 2000. P. 335–342.
16. Vyatkin S. I. Complex Surface Modeling Using Perturbation Functions, *Optoelectronics, Instrumentation and Data Processing*. 2007. Vol. 43. Issue 3. P. 226–231.

References

1. Vyatkin, S. I., Romanyuk, O. V., & Obidnik, M. D. (2009). Osobennosti poligonal'nogo modelirovaniya v sistemakh komp'yuternoy grafiki. Proceedings of the *Sovremennyye problemy radioelektroniki, telekommunikatsiy i priborostroyeniya SPRTP–2009*: *materialy IV Mezhdunarodnoy nauchno-tekhnicheskoy konferentsii*. (Vinnitsa, October 8–10, 2009), Vinnitsa, pp. 24.
2. Vyatkin, S. I., Romanyuk, A. N., & Kostyukova, N. S. (2011). Bazy dannykh i modeliruyushchiye kompleksy dlya sistem vizualizatsii real'nogo vremeni. Proceedings of the *Informatsionnyye tekhnologii i informatsionnaya bezopasnost' v nauke, tekhnike i obrazovanii*: *materialy Mezhdunarodnoy nauchno-prakticheskoy konferentsii,* (Sevastopol', September 5–10, 2011), Sevastopol, pp. 156–157.
3. Gortler, S., Grzeszczuk, R., Szeliski, R., & Cohen, M. (1996). The Lumigraph. Proceedings of the *SIGGRAPH '96*: *23rd Annual Conference on Computer Graphics and Interactive Techniques*. (New Orleans, August 4-9, 1996). New York: Association for Computing Machinery, pp. 43–54.
4. Shade, J., Gortler, S., He, L, & Szeliski, R. (1998). Layered Depth Images. Proceedings of the *SIGGRAPH '98*: *25th Annual Conference on Computer Graphics and Interactive* Techniques. (Orlando, July 19-24, 1998). New York: Association for Computing Machinery, pp. 231–242.
5. Levoy, M., & Hanrahan, P. (1996). Light Field Rendering. Proceedings of the *SIGGRAPH '96*: *23rd Annual Conference on Computer Graphics and Interactive Techniques*. (New Orleans, August 4-9, 1996). New York: Association for Computing Machinery, pp. 31–42.
6. Shade, J., Lischinski, D., Salesin, D., DeRose, T., & Snyder, J. (1996). Hierarchical Image Caching for Accelerated Walkthroughs of Complex Environments. Proceedings of the *SIGGRAPH '96*: *23rd Annual Conference on Computer Graphics and Interactive Techniques*. (New Orleans, August 4-9, 1996). New York: Association for Computing Machinery, pp. 75–82.
7. Schaufler, G. (1998). Per-Object Image Warping with Layered Impostors. Proceedings of the *SIGGRAPH '98*: *25th Annual Conference on Computer Graphics and Interactive Techniques*. (Orlando, July 19-24, 1998). New York: Association for Computing Machinery, pp. 145–156.
8. Levoy, M., & Whitted, T. (1985). The Use of Points as a Display Primitive. Norh Carolina : University of Norh Carolina at Chapel Hill. (Technical report 85-022).
9. Vyatkin, S. I. (2016). Modelirovaniye neodnorodnostey pri vizualizatsii atmosfernykh

https://doi.org/10.32782/KNTU2618-0340/2020.3.2-2.5

effektov. *Vestnik komp'yuternykh i informatsionnykh tekhnologiy*. **7**, 9–14.
10. Vyatkin, S., Romaniuk, S., & Romaniuk, O. (2015). Visualization of 3D-amorphous objects using free forms. *Elektrotekhnicheskiye i komp'yuternyye sistemy*. **19** (95), 227−230.
11. Vyatkin, S. I., Romanyuk, A. N., & Poddubetskaya, M. P. (2013). Animatsiya trekhmernykh ob"yektov. *Izmeritel'naya i vychislitel'naya tekhnika v tekhnologicheskikh protsessakh*. **1** (42), 207–211.
12. Vyatkin, S. I., Romanyuk, A.N., Pavlov, S. V., Moskovko, M. V., Askarova, N., Sagymbekova, A., Wojcik, W., & Kotyra, A. (2017). Fast Ray Casting of Function-Based Surfaces. *Przeglad Elektrotechniczny*. **5**, 83−86. DOI:10.15199/48.2017.05.16
13. Grossman, J., & Dally, W. (1998). Point Sample Rendering. Proceedings of the *Rendering Techniques '98*: *Eurographics Workshop in Vienna*. (Vienne, June 29 – July 1, 1998), Vienne: Springer, pp. 181–192.
14. Rusinkiewicz, S., & Levoy, M. (2000). Qsplat: A Multiresolution Point Rendering System for Large Meshes. Proceedings of the *SIGGRAPH '00*: *27th Annual Conference on Computer Graphics and Interactive Techniques*. (New Orleans, July 23-28, 2000). New York: ACM Press/Addison-Wesley Publishing Co., pp. 343–352.
15. Pfister, H., Zwicker, M., van Baar, J., & Gross, M. (2000). Surfels: Surface Elements as Rendering Primitives. Proceedings of the *SIGGRAPH '00*: *27th Annual Conference on Computer Graphics and Interactive Techniques*. (New Orleans, July 23-28, 2000). New York: ACM Press/Addison-Wesley Publishing Co., pp. 335–342.
16. Vyatkin, S. I. (2007). Complex Surface Modeling Using Perturbation Functions, *Optoelectronics, Instrumentation and Data Processing*. **43**, 3, 226–231.

Вяткин Сергей Иванович – к.т.н., ст.н.с. лаборатории синтезирующих систем визуализации Института автоматики и электрометрии СО РАН, e-mail: sivser@mail.ru, ORCID: 0000-0002-1591-3588.

Романюк Александр Никифорович – д.т.н., профессор, заведующий кафедрой программного обеспечения Винницкого национального технического университета, e-mail: rom8591@gmail.com, ORCID: 0000-0002-2245-3364.

Рейда Александр Николаевич – к.т.н., доцент, доцент кафедры программного обеспечения Винницкого национального технического университета, e-mail: rom8591@gmail.com, ORCID: 0000-0001-8231-6268.

Романюк Оксана Владимировна – к.т.н., доцент, доцент кафедры программного обеспечения Винницкого национального технического университета, e-mail: rom8591@gmail.com, ORCID: 0000-0003-0235-8615.

https://doi.org/10.32782/KNTU2618-0340/2020.3.2-2.5