

Использование двухуровневой трассировки лучей для визуализации динамически изменяемых сцен.

Б.Х. Барладян, А.Г. Волобой, Л.З. Шапиро
Институт прикладной математики им. М.В.Келдыша РАН, Москва

Аннотация

В статье описывается алгоритм двухуровневой трассировки лучей. Все современные подходы к трассировке используют вспомогательные, ускоряющие структуры, которые обычно создаются заранее, при подготовке сцены. Время, необходимое для построения таких структур, может быть существенным и иногда превышать время непосредственно генерации изображения. При динамическом изменении геометрии сцены возникает задача быстрого воссоздания ускоряющих структур. Сам процесс трассировки был существенно ускорен в последнее время. Поэтому изменение вспомогательных структур для динамических сцен является на данном этапе главным сдерживающим фактором использования трассировки лучей в интерактивных приложениях. В статье предлагается решение этой проблемы с помощью алгоритма двухуровневой трассировки лучей. Время динамической модификации вспомогательных структур было ускорено в десятки раз и достигло миллисекунд, что является приемлемым для интерактивного приложения.

Ключевые слова: реалистичные изображения, интерактивная визуализация, трассировка лучей, BSP дерево, динамически изменяемая геометрия.

1. ВВЕДЕНИЕ

В настоящее время все большее внимание уделяется разработке методов трассировки лучей, обеспечивающих максимальную интерактивность взаимодействия пользователя с системой. Хорошо известно, что методы генерации изображений, основанные на трассировке лучей, позволяют создавать корректные и фото реалистичные изображения, однако именно трассировка лучей в этих системах является наиболее затратной с точки зрения процессорного времени процедурой. В силу всех этих причин создание изображений в системах основанных на трассировке лучей, обычно, не использовалось в интерактивных системах визуализации и системах визуализации реального времени.

Для интерактивной генерации изображений использовалась, в основном, растеризация треугольников с использованием возможностей современных графических ускорителей. Учитывая относительную дешевизну, высокую эффективность и доступность таких ускорителей этот подход в настоящее время доминирует на рынке графических интерактивных приложений.

Тем не менее, системы генерации изображений, основанные на трассировке лучей, пользуются заслуженным интересом у значительного числа пользователей, поскольку позволяют создавать более качественные изображения. Они существенно более корректно учитывают различные

физические эффекты, такие как точные отражения, преломления света, тени, позволяют задавать сложные свойства поверхностей, освещение источниками света с гониодиаграммами или изображениями с высоким динамическим диапазоном и т.д. Системы, основанные на трассировке лучей, являются наиболее эффективным подходом для геометрического моделирования сложных оптических устройств [1].

Следует также учитывать, что время создания изображения в системе, основанной на трассировке лучей, растет с увеличением сложности сцены, как правило, как логарифм ее сложности (числа треугольников). В то время как для генерации изображения, основанного на растеризации треугольников, время растет линейно.

Эти достоинства систем генерации изображений с помощью трассировки лучей вызывают непрекращающийся интерес к различным методам ее ускорения посредством распараллеливания алгоритма [2], использования SIMD (SSE) инструкций [3, 4] или графических ускорителей [5]. В результате в последнее время скорость трассировки лучей была существенно увеличена.

Все это позволило создавать графические приложения, основанные на трассировке лучей, со скоростью генерации изображений близкой к интерактивной. При этом для ускорения трассировки лучей такие системы используют специальные алгоритмы, основанные на разбиении всего пространства сцены (обычно ограничивающего параллелепипеда) на множество подпространств, образуя из них линейное множество или некоторую иерархию. В итоге разбиения пространства строятся вспомогательные структуры, хранящие индексы объектов сцены, принадлежащих каждому подпространству. Существует большое множество различных видов вспомогательных структур: равномерное разбиение [6], октарные деревья, BSP и kd деревья [7, 8]. Время, необходимое для построения таких структур, может быть существенным и иногда значительно превышать время генерации изображения для одного положения камеры.

Большое время генерации вспомогательных структур остается приемлемым, пока создание изображений ограничивается статическими сценами, в которых изменяется только положение камеры или расположение источников света. Однако в современных САПР [9], системах высококачественной презентации продуктов [10] пользователь работает с динамическими изменяемыми сценами, когда интерактивно меняется также и геометрия сцены. При этом пользователя системы интересует полное время от изменения сцены до получения изображения или результата моделирования, включая время перестроения ускоряющих структур. Таким образом, задача быстрого перестроения вспомогательных ускоряющих структур становится особенно актуальной для использования качественной визуализации методом трассировки лучей.

2. ДВУХУРОВНЕВАЯ ТРАССИРОВКА ЛУЧЕЙ

Система интерактивной визуализации, разработанная в Институте прикладной математики РАН, использует SIMD (SSE) инструкции для ускорения трассировки лучей [11]. Эффективная реализация трассировки лучей на базе SSE инструкций потребовала замены ранее использовавшейся ускоряющей структуры, построенной на многоуровневом, равномерном разбиении пространства сцены, на структуру, основанную на BSP дереве. При обычной, не когерентной трассировке использование новой структуры не дает существенного выигрыша по сравнению с предыдущей. Однако одновременная трассировка четверки лучей с использованием SSE инструкций существенно проще и эффективнее реализуется для структур основанных на BSP дереве. В то же время построение ускоряющих структур на основе BSP деревьев является более медленной процедурой по сравнению с построением многоуровневого, равномерного разбиения. Это существенно обострило проблему перестроения ускоряющих структур для динамически изменяемых сцен. Для решения проблемы была предложена схема двухуровневой трассировки лучей. Будем говорить о верхнем (первом) и нижнем (втором) уровнях трассировки.

В большинстве современных системах визуализации сцены представлены в виде иерархического графа (рис. 1). Листья графа представляют объекты сцены, свойства поверхностей и материалы, источники света и т.д. Для нас важным является факт, что вся геометрия сцены состоит из объектов, которые пользователь может добавлять в сцену либо убирать из нее, двигать в пространстве, или модифицировать каким-либо образом. Эти действия применяются к одному или нескольким объектам сцены.

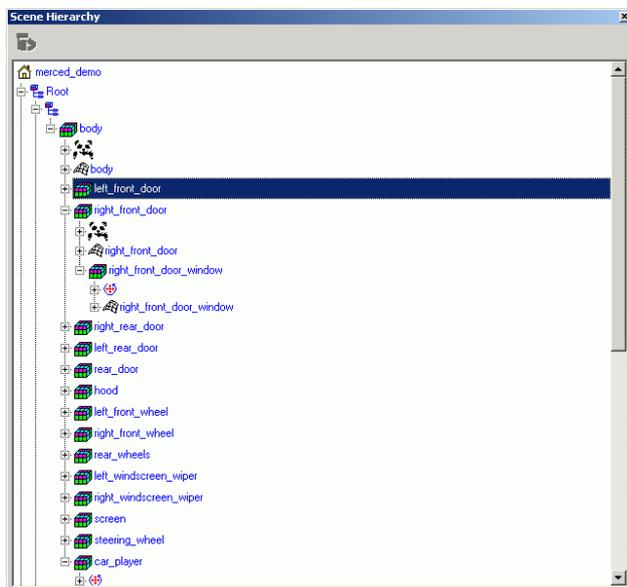


Рис. 1. Представление сцены в виде иерархического графа.

Трассировка верхнего уровня определяет объекты сцены, с которыми возможно пересечение трассируемого луча. Для этого все объекты сцены помещаются в параллелепипеды, их ограничивающие. Граф сцены разворачивается в линейный

список объектов (точнее, ограничивающих параллелепипедов), и затем для них строится вспомогательная структура BSP дерева.

На втором, нижнем уровне, луч трассируется непосредственно внутри ограничивающего параллелепипеда. В общем случае трассировка луча на нижнем уровне может быть разной для разных параллелепипедов, если они содержат объекты различных типов. Каждый объект имеет свою ускоряющую структуру. При этом все копии одного объекта в сцене (т.е. все объекты сцены, которые можно получить один из другого применением аффинного преобразования) являются одним типом объекта, имеют одинаковые ускоряющие структуры для трассировки нижнего уровня, но являются разными включениями в структуры трассировки верхнего уровня.

Организация сцены для двухуровневой трассировки проиллюстрирована на рис. 2.



Рис. 2. Организация сцены для двухуровневой трассировки лучей. Красными контурами показаны параллелепипеды, ограничивающие объекты сцены.

При изменениях сцены ускоряющие структуры модифицируются следующим образом:

- при добавлении копии объекта в сцену модифицируется только ускоряющая структура трассировки верхнего уровня;
- при удалении объекта из сцены модифицируется только ускоряющая структура трассировки верхнего уровня;
- при движении объекта, его поворотах либо масштабировании модифицируется только ускоряющая структура трассировки верхнего уровня;
- при добавлении в сцену нового, уникального объекта создается его ускоряющая структура и модифицируется ускоряющая структура трассировки верхнего уровня.

Одно из наиболее естественных действий при построении сцены - это добавление объектов в сцену из библиотеки. Так как объекты библиотеки были подготовлены заранее, то

вместе с ними можно хранить также и ускоряющие структуры для трассировки объекта. В результате отпадает необходимость создавать вспомогательные структуры в динамическом режиме, и требуется только модификация структур трассировки верхнего уровня.

Вторым важным преимуществом двухуровневой трассировки является то, что она позволяет обеспечить эффективную трассировку объектов различной природы. Для объектов разных типов хранятся разные вспомогательные структуры, наиболее эффективные для каждого типа. На первом этапе разработки системы визуализации рассматривались только геометрические объекты, состоящие из массивов треугольников. В настоящее время реализуются расширения для других типов объектов. В частности, это могут быть элементы наблюдения, представляющие собой виртуальные светочувствительные сенсоры, которые, не оказывая влияния на распространение света, регистрируют лучи в процессе их трассировки и накапливают распределение световой энергии. Предложенный подход позволяет моделировать распространение света и визуализировать объекты различной физической природы, поскольку каждый объект сцены может иметь собственные вспомогательные структуры и собственный алгоритм трассировки.

Для ускорения трассировки лучей на обоих уровнях используется BSP дерево. Соответственно на обоих уровнях используется одинаковая ускоряющая структура данных и один и тот же алгоритм трассировки лучей до пересечения с объектом трассировки на данном уровне: ограничивающим параллелепипедом на верхнем уровне и объектом сцены или его примитивом на нижнем. Однако алгоритм построения дерева для каждого уровня имеет свою специфику. Для построения BSP дерева нижнего уровня (в случае, если объект состоит из треугольников) наиболее дорогостоящей по времени процедурой является нахождение пересечения треугольника (примитива объекта) с параллелепипедом данного уровня BSP дерева. Размеры параллелепипедов, ограничивающих треугольники, могут изменяться при этом по всем трем координатам, что в свою очередь требует сортировки возможных положений разделяющих плоскостей. При этом для построения BSP дерева верхнего уровня процедура нахождения пересечения параллелепипеда, ограничивающего объект, с параллелепипедом уровня BSP дерева не является критической, а сортировка возможных положений разделяющих плоскостей не нужна вовсе.

Эти упрощения существенно ускоряют процесс построения BSP дерева верхнего уровня. Однако эта процедура происходит при каждом изменении сцены. Следовательно, именно здесь желательно достичь максимального ускорения. В работе [12] было предложено два дополнительных подхода для ее ускорения:

- выбирать для деления координату соответствующую максимальной стороне параллелепипеда данного уровня;
- выбирать положение разделяющей плоскости между геометрическим центром данного уровня и медианой набора объектов ему соответствующих.

Оба подхода были опробованы и немного ускорили построение BSP дерева. Однако они привели к замедлению трассировки на 15-20% для ряда тестируемых сцен. Так как нас интересует суммарное время генерации изображения, то это подходы были отвергнуты.

3. РЕЗУЛЬТАТЫ

Двухуровневая трассировка лучей используется в нашей системе для построения фотореалистичных изображений, для оптического моделирования и ряда вспомогательных операций, необходимых в процессе назначения атрибутов, выбора объектов, анализа сцены и т.д. Время перестроения ускоряющих структур для сцен, состоящих из нескольких десятков объектов и сотен тысяч треугольников, составляет миллисекунды. Оно становится заметным только для сцен, состоящих из тысяч объектов.

Предлагаемый подход был проверен на персональном компьютере Pentium IV, 3.2 GHz, 2 Gb RAM. Для динамической сцены (рис. 3), состоящей из 19 объектов (количество треугольников – 137286), полное время генерации ускоряющих структур составило 7.89 секунды. А время перестроения структур при изменении геометрии (например, открытие двери, как показано на рис. 3b) – менее 0.01 секунды.



(a)



(b)

Рис. 3. Пример сцены с динамически изменяемой геометрией (в сцену заложена возможность открытия дверей и окон автомобиля, движения дворников).

Как уже упоминалось, время перестроения ускоряющих структур возрастает с количеством объектов. Ниже приведены результаты построения ускоряющих структур для двух сцен с большим количеством объектов. Для первой сцены, состоящей из 6804 объектов (100215 треугольников), время полного создания ускоряющих структур составило 2.94 секунды, а время перестроения 0.1 секунды. Для второй сцены, состоящей из 20968 объектов (302822 треугольников), время полного создания ускоряющих структур составило 9.55 секунды, а время перестроения – 0.25 секунды.

Приведенные результаты показывают, что даже для сцен с большим количеством объектов время перестроения вспомогательных структур вполне приемлемо для интерактивного взаимодействия пользователя с программой.

Работа поддержана грантами РФФИ № 06-07-89165, № 05-01-00345-а, а также компанией Integra Inc. (Япония).

Версия статьи с цветными иллюстрациями размещена по адресу

http://www.keldysh.ru/pages/cgraph/publications/cgd_publ.htm.

4. ЛИТЕРАТУРА

[1] А.Г. Волобой, В.А. Галактионов, Машинная графика в задачах автоматизированного проектирования. «Информационные технологии в проектировании и производстве» № 1, 2006, стр. 64-73.

[2] Robert Cook, Thomas Porter, and Loren Carpenter. Distributed raytracing. In ACM SIGGRAPH Computer Graphics, volume 18, 1984, pp. 137–144.

[3] Ingo Wald and Philipp Slusallek. State-of-the-Art in Interactive Ray-Tracing in State of the Art Reports, EUROGRAPHICS-2001, Manchester, United Kingdom, September 3-7, 2001, pp. 21-42.

[4] Ingo Wald, Philipp Slusallek, and Carsten Benthin. Interactive distributed ray tracing of highly complex models. In Proceedings of the 12th EUROGRAPHICS Workshop on Rendering, June 2001, London, pp. 274 -285.

[5] Timothy John Purcell, Ian Buck, William R. Mark, and Pat Hanrahan. Ray tracing on programmable graphics hardware. ACM Transactions on Graphics, Vol. 21, No.3, 2002 (Proceeding of SIGGRAPH), pp. 703-712.

[6] Akira Fujimoto, Tanaka Takayuki, Iwata Kansei. ARTS: Accelerated Ray-Tracing System, IEEE Computer Graphics and Application, Vol. 6, № 4, 1986, pp.16-26.

[7] Andrew Glassner. An Introduction to Raytracing. Academic Press, 1989.

[8] Vlastimil Havran. Heuristic Ray Shooting Algorithms. PhD thesis. Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic, November 2000. Available at <http://www.cgg.cvut.cz/~havran/DISSVH/phdthesis.html>

[9] Б.Х. Барладян, А.Г. Волобой, Л.З. Шапиро, Интеграция моделирования освещенности методом трассировки лучей в системы автоматизированного проектирования. 16-ая Международная конференция по компьютерной графике и ее приложениям – Графикон-2006, Новосибирск, 2006, стр. 275-278.

[10] Terminal V. Virtual Reality Center. <http://www.vr-center.at>

[11] A. Adinetz, B. Barladian, V. Galaktionov, L. Shapiro, A. Voloboy, Physically Accurate Rendering with Coherent Ray Tracing. Proceeding of GraphiCon'2006 - The 16-th International conference of Computer Graphics and Applications, Novosibirsk, 2006, pp. 8-15.

[12] Ingo Wald, Carsten Benthin and Philipp Slusallek A Simple and Practical Method for Interactive Ray Tracing of Dynamic Scenes. Technical Report 2002-04, Saarland University. Available at <http://graphics.cs.uni-sb.de/Publications/TR/2002/Dyn/DynamicRayTracing.pdf>

Abstract

The two level ray tracing algorithm is considered. Modern ray tracing systems use auxiliary structures to accelerate ray tracing process. Generation of such structures (made during preprocessing) usually is time consuming and sometimes can be longer than final image generation. In interactive systems the geometry can be changed dynamically which leads to auxiliary structures re-generation. Taking into account that recent research allowed to speed up ray tracing drastically the acceleration structure update became bottleneck for interactive systems based on ray tracing visualization. The two level ray tracing algorithm is proposed as a remedy for the problem. In our algorithm the speedup of auxiliary structure generation is tens times. Absolute time of structure update was reduced to milliseconds that are acceptable for interactive visualization.

Keywords: *realistic rendering, interactive rendering, ray tracing, BSP tree, dynamic geometry.*

Authors:

Boris H. Barladyan, PhD, senior researcher of the Keldysh Institute for Applied Mathematics RAS.

E-mail: obb@gin.keldysh.ru

Alexey G. Voloboy, PhD, senior researcher of the Keldysh Institute for Applied Mathematics RAS.

E-mail: voloboy@gin.keldysh.ru

Lev Z. Shapiro, PhD, senior researcher of the Keldysh Institute for Applied Mathematics RAS.