

Об одной реализации метода фотонных карт для визуализации каустик

Ульянов Данила Ярославович
Нижегородский Государственный Технический Университет им. Р. Е. Алексева
Нижний Новгород, Российская Федерация
danila-ulyanov@narod.ru

Аннотация

В данной работе описывается метод визуализации каустик на основе техники трассировки лучей. Генерация карты фотонов и последующая визуализация выполняются на графическом процессоре. Для повышения эффективности используется пространственная хэш-функция и битоническая сортировка. Реализация алгоритма базируется на языке C++ и OpenGL / GLSL. Даны оценки производительности.

Keywords: Ray tracing, Caustics, Photon maps, GPU sorting.

1. ПОДХОДЫ К ПОСТРОЕНИЮ КАУСТИК

В основе расчётов каустик лежит, как правило, прямая трассировка лучей от источников света. Результат трассировки обычно сохраняется в специальную базу данных – *фотонную карту* – и используется на этапе визуализации. Данный подход моделирования глобального освещения, и каустик в частности, называется методом фотонных карт. Существует, по крайней мере, ещё один способ визуализации каустик, называемый *caustics mapping* [2]. Выбор между этими двумя методами, прежде всего, зависит от способа визуализации сцены. Это либо стандартный для графического конвейера метод растеризации примитивов, либо метод обратной трассировки лучей. Техники глобального освещения (GI), обеспечивающие попутно и построение каустик, пока не доступны для реального времени. Рассмотрим далее другие варианты получения нужных эффектов.

1.1 Caustics Mapping

Метод разработан для наиболее применимого, в настоящее время, способа визуализации сцены, называемого растеризацией. Примитивы, из которых состоит сцена, проецируются на плоскость экрана и сортируются по глубине. Реализация данного метода рассмотрена подробно в [2] (рис.1). Метод дан для сравнения с основным методом, реализованным в статье.

При проективном построении сцены авторы предлагают использовать отдельный вершинный буфер в качестве набора фотонов. В вершинном шейдере данные фотоны трассируются через сцену по законам отражения и преломления. Весь алгоритм выполняется в пространстве изображения (*image space*), поэтому информация о геометрии сцены передаётся через заранее построенные карты нормалей и глубины. Результатом работы алгоритма является карта каустик, которую можно отфильтровать и проективно наложить на геометрию сцены. Основные ограничения подхода схожи с таковыми в технике теневых карт (*shadow maps*). При реализации данного подхода могут возникать сложности с фильтрацией карты каустик и ее нежелательным зашумлением (см. рис. 2). Существенное ограничение на

применение метода налагает также ограниченный размер полученной карты.

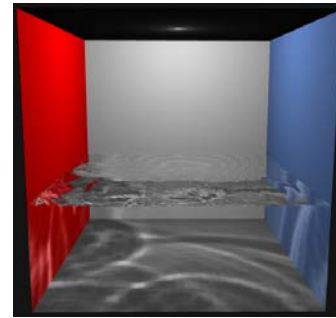


Рис. 1. Пример подводных каустик, построенных методом Caustics Mapping [2]

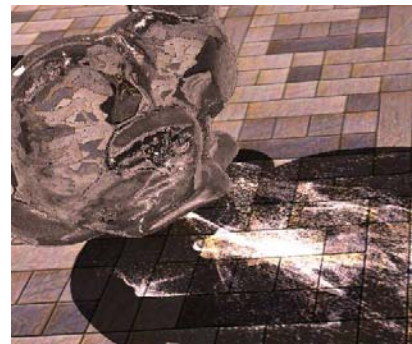


Рис. 2. Пример зашумления карты каустик [5]

Тем не менее, метод вполне пригоден для использования в графических приложениях реального времени.

1.2 Photon Mapping

Для метода визуализации сцены с помощью обратной трассировки лучей используется метод фотонных карт. Механизм трассировки лучей от наблюдателя (для визуализации) и от источника света (для построения фотонной карты) практически одинаков, что позволяет использовать для трассировки фотонов уже имеющийся в реализации расчёта изображения код трассировки. Об этом методе и пойдёт речь в далее.

2. ПОСТРОЕНИЕ ФОТОННОЙ КАРТЫ

Для реализации задачи построения каустик выбраны язык программирования C++ и OpenGL API совместно с шейдерным языком GLSL. Так как практически все расчёты производятся на GPU, важно отметить, что все тесты производились на компьютере, оснащённом видеокартой NVIDIA GeForce 250 GTS. Визуализация сцены производится

методом обратной трассировки лучей, описанном в работе [1].

На каждом кадре генерируется набор лучей, как показано на рис. 3. Для получения фотонной карты данные лучи – “фотоны” света – трассируются от источника света через геометрию сцены. В данном случае, поскольку сцена задаётся аналитически и достаточно проста, ускоряющие структуры для нахождения пересечений не востребованы и были опущены.

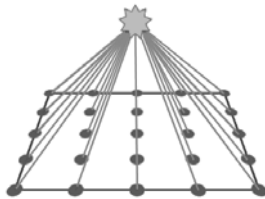


Рис. 3. Схема испускания фотонов

После трассировки конечное положение каждого фотона сохраняется в фотонной карте и может быть учтено при расчете цвета фрагмента на результирующем изображении.

Есть возможность значительно ускорить визуализацию сцены с каустиками по сравнению с полным перебором фотонной карты. Для этого нужно отсортировать фотонную карту так, чтобы уменьшить время доступа к ближайшим фотонам в окрестности точки соударения. Это можно сделать, используя пространственную хэш-функцию и битоническую сортировку (*bitonic sort*) [3]. Фотоны, непосредственно влияющие на фрагмент, при этом находятся бинарным поиском.

2.1 Производительность

Результат работы программы показан на рис 4. Все тесты для построенной программы проводились с разрешением фотонной карты 128x128, соответственно трассировалось 16384 фотона. По результатам тестов (см. рис. 5) видно, что частота кадров имеет примерно логарифмическую зависимость от общего числа обрабатываемых фрагментов и соответственно заметно убывает с увеличением выходного разрешения. При увеличении числа отражающих/преломляющих объектов частота кадров меняется не столь существенно, это объясняется тем, что наиболее затратными (по времени выполнения) этапами визуализации, являются этапы сортировки фотонной карты и работы с ней для конкретного фрагмента.



Рис. 4. Пример визуализации каустики для отражающего кольца методом Photon Mapping

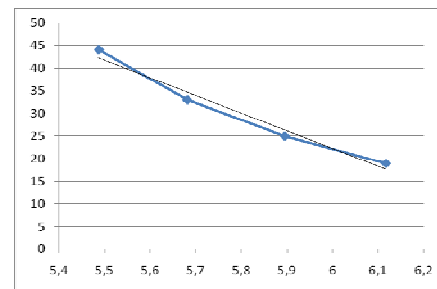


Рис. 5. График зависимости частоты кадров от $\log(N)$ где N - число фрагментов. Узловые точки соответствуют выходным разрешениям: 640x480, 800x600, 1024x768, 1280x1024

Доступная реализация техники Caustics Mapping [5] в разрешении 1024x768 выдаёт в среднем 70 кадров в секунду. В то время как наша программа дает 25 кадров в секунду, также обеспечивая реальное время. Другими словами Caustics Mapping (в данной реализации) не имеет однозначного преимущества.

3. ЗАКЛЮЧЕНИЕ

На данный момент обе технологии являются слишком ресурсоёмкими для использования в графических приложениях связанных с интерактивными развлечениями. Однако в скором времени ситуация может измениться. К тому же метод фотонных карт, как и трассировка лучей в целом, обеспечивает большую точность визуализации, и потому может быть предпочтён в ряде случаев, когда это имеет решающее значение.

4. ССЫЛКИ

- [1] Timothy J. Purcell, Ian Buck, William R. Mark, Pat Hanrahan. *Ray tracing on programmable graphics hardware* /SIGGRAPH'2002 (<http://graphics.stanford.edu/papers/rtongfx/>)
- [2] Musawir A. Shah, Jaakko Kontinen, S. N. Pattanaik, "Caustics Mapping: An Image-space Technique for Real-Time Caustics", IEEE Transactions on Visualization and Computer Graphics, March/April 2007, Vol 13(2), pp 272-280 (<http://graphics.cs.ucf.edu/caustics/final.pdf>)
- [3] Peter Kipfer. Chapter 46: Improved GPU Sorting/ GPU Gems 2. Apr 2005 (http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter46.html)
- [4] Steven Parker, William Martin, Peter-Pike J. Sloan, Peter Shirley, Brian Smits, Charles Hansen. *Interactive Ray Tracing*. 1999 Symposium on Interactive 3D Graphics, April, 1999 (<http://www.ppsloan.org/publications/rttr99.pdf>)
- [5] Резник С. Моделирование стеклянных поверхностей / Портал уральских разработчиков игр, июнь 2010. (<http://www.uraldev.ru/articles/id/39>)

Об авторе

Ульянов Данила Ярославович - студент Нижегородского Государственного Технического Университета им.П.Е.Алексеева, danila-ulyanov@narod.ru