

ОПТИМИЗАЦИЯ МЕТОДА ТРАССИРОВКИ ЛУЧЕЙ

Сереженко А.А., Мальчева Р.В.

Донецкий национальный технический университет

Работа посвящена особенностям оптимизации и применения метода трассировки лучей в 3D - графике. Кратко описан алгоритм, структура, возможности, применение и пример использования.

1. Введение

С момента появления алгоритмов расчета отражения и преломления света (ray tracing, трассировка лучей) их вычисление обычно рассматривалось как статический процесс и представляло собой весьма трудоемкий способ получения красивых изображений. В последнее время появились возможности расчета лучей в интерактивном режиме, даже при использовании довольно доступного аппаратного обеспечения. С учетом повышения эффективности таких расчетов и благодаря преимуществам современных разработок в области микропроцессоров, в частности, таких как технология Hyper-Threading и многоядерных процессоров, можно ожидать, что в ближайшем будущем эти технологии будут применяться как в интерактивных приложениях, так и в приложениях реального времени.

2. Описание метода

Основной сегодняшней целью относительно метода трассировки лучей является усовершенствование метода в сторону уменьшения количества параллельных процессоров для обработки кадра (читай дальше – фрейма).

Введем обозначения X_p – процент расхождения, X_{max} – размер фрейма по вертикали, Y_{max} – размер фрейма по горизонтали, F_{gx} – шаг интерполяции фрейма по горизонтали, Y_{gx} – шаг интерполяции фрейма по вертикали, N_{cpu} – количество процессоров, задействованных в обработке кадра (рис 1.)

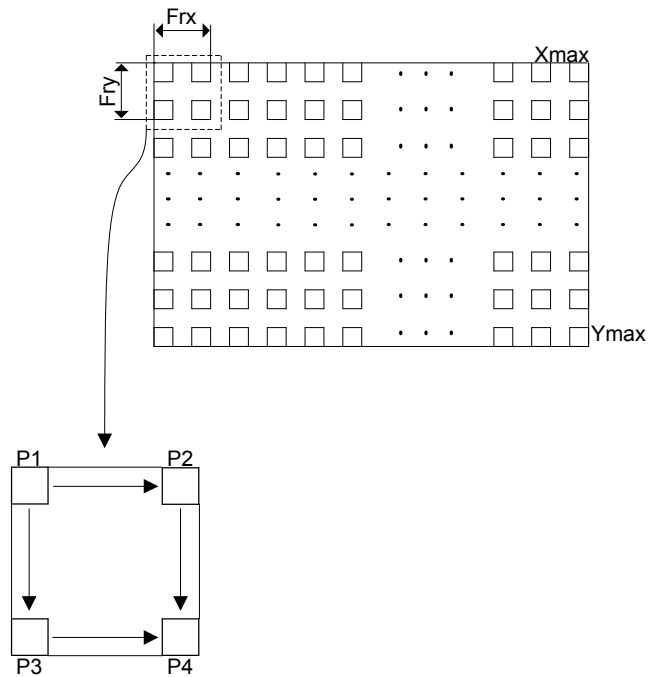


Рисунок 1. Фрейм, пиксельная решетка.

Принцип работы.

X_p – процент расхождения определяется исходя из восприятия человеческим глазом цветовых оттенков и интенсивности.

Если цветовой тон или интенсивность в пикселях (P1-P2), (P3-P4) отличается более чем на X_p %, то шаг интерполяции F_{rx} стоит уменьшить в 2 раза.

Если цветовой тон или интенсивность в пикселях (P1-P3) отличается более чем на X_p %, то шаг интерполяции F_{ry} стоит уменьшить в 2 раза.

Используя классический метод трассировки лучей, что бы обработать 1 кадр размером X_{max} x Y_{max} , потребуется $(X_{max} * Y_{max})$ процессоров. После выполнения поставленной задачи, можно получить заметное уменьшение N_{cpu} , т.е. N_{cpu} уже не будет равен количеству пикселей в кадре, а

$$N_{cpu} = (X_{max}/F_{rx} * Y_{max}/F_{ry}) + (N_{frx}-1) * (N_{fry}-1).$$

Пример:

Пусть есть структура системы синтеза, представленная на рисунке 2, где R_{tpu} – ray tracing processors, I_u – interpolators units.

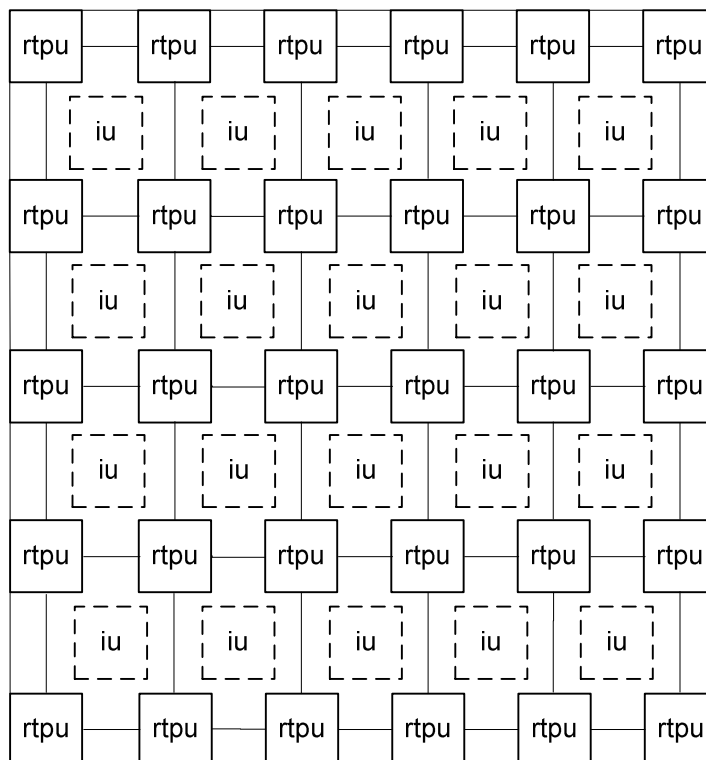


Рисунок 2. Пример структуры системы синтеза.

Из структуры видно, что:

$$N_{cpu} = (X_{max}/Fr_x * Y_{max}/Fr_y) + (N_{frx}-1)*(N_{fry}-1) = 6*5 + (6-1)*(5-1) = 6*5 + 6*4 = 54 \text{ процессоров.}$$

Достоинства алгоритма.

Отражения - это одна из областей, в которых метод трассировки лучей превосходно показывает себя. При методе трассировки лучей отражения отображаются идеально, причём без сложных алгоритмов, поскольку всё просчитывается основным алгоритмом рендеринга. Ещё одним преимуществом можно считать вывод отражений частей объекта друг на друга.

Еще одним достоинством метода является возможность рендеринга гладких объектов без интерполяции их полигональными поверхностями (например, треугольниками).

Другим бесспорным преимуществом метода трассировки лучей является качественная обработка эффектов прозрачности. С помощью алгоритма растеризации крайне сложно выводить эффекты прозрачности, поскольку расчёт прозрачности напрямую зависит от порядка рендеринга. Чтобы получить хорошие результаты, необходимо отсортировать прозрачные полигоны в порядке самых дальних от камеры к самым близким, а уже потом выполнять рендеринг.

Недостатки алгоритма.

Серьёзным недостатком метода обратного трассирования является производительность. Метод растеризации и сканирования строк использует когерентность данных, чтобы распределить вычисления между пикселями. В то время как метод трассирования лучей каждый раз начинает процесс определения цвета пикселя заново, рассматривая каждый луч наблюдения в отдельности. Впрочем, это разделение влечёт появление некоторых других преимуществ, таких как возможность трассировать больше лучей, чем предполагалось для устранения контурных неровностей в определённых местах модели. Также это регулирует отражение лучей и эффекты преломления, и в целом — степень фотореалистичности изображения.

Распараллеливание алгоритма.

Трассировка лучей по своей сущности исключительно удачно подходит для параллельных вычислений. Для расчета отдельных лучей не используются общие данные, поэтому лучи могут рендериться в произвольном порядке. Это означает, что алгоритм трассировки лучей теоретически может использовать преимущества современных процессорных технологий. При том, что большинство приложений могут лишь частично выполняться в параллельном режиме, трассировка лучей сравнительно легко адаптируется к таким технологиям параллельной обработки данных, как SIMD, Hyper-Threading и многоядерные процессоры.

В случае применения Hyper-Threading Technology на двухъядерной платформе, запросы на построение лучей могут быть равномерно распределены между доступными процессорами. Для облегчения этой задачи можно инициировать два потока трассировки лучей, каждый из которых выполняет половину расчетов всей задачи. Такая схема может без труда применяться на более сложных конфигурациях; производительность операций трассировки лучей масштабируется почти в прямой зависимости от количества доступных процессоров.

Сложнее обстоят дела с использованием технологии SIMD. Алгоритм трассировки лучей становится зависимым от пропускной способности памяти, поскольку каждый луч рассчитывается как проходящий через некоторую пространственную структуру и проверяется на предмет пересечения с несколькими примитивами для определения ближайшей точки пересечения.

3. Заключение

Предстоит решить немало проблем, прежде чем метод трассировки лучей станет достойной альтернативой растеризации в сфере рендеринга в реальном времени. Преимущества метода трассировки лучей не такие

революционные, чтобы оправдать существенное снижение производительности. Сильные моменты алгоритма связаны с отражениями и прозрачностью, поскольку два этих эффекта сложнее всего вывести на существующих алгоритмах растеризации. Мир вокруг нас не состоит целиком из очень прозрачных или сияющих объектов, поэтому наше зрение вполне может удовлетвориться грубым приближением.

Литература:

1. Morris, Daniel. “5 Hot New Gaming Technologies” *PC Gamer Online*, July 1999. October 1999, <http://www.pcgamer.com/>.

2. NVIDIA Corporation. “Ray tracing today and tomorrow” Октябрь 2002, <http://www.nvidia.com/techbriefs.html/>).