

ОПТИМИЗАЦИЯ МЕТОДА ТРАССИРОВКИ ЛУЧЕЙ

Серезенко А.А., Мальцева 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.

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.

1 Анализ метода трассировки лучей

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.

Другим бесспорным преимуществом метода трассировки лучей является качественная обработка эффектов прозрачности.

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.

2 Распараллеливание алгоритма

Трассировка лучей по своей сущности исключительно удачно подходит для параллельных вычислений. Для расчета отдельных лучей не используются общие данные, поэтому лучи могут трассироваться в произвольном порядке. Это означает, что алгоритм трассировки лучей теоретически может использовать преимущества современных процессорных технологий. Не смотря на то, 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.

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 Оптимизация метода трассировки лучей

Классическое распараллеливание метода трассировки лучей подразумевает применение пиксельных процессоров, выполняющих алгоритм трассировки для каждого пиксела экрана (и соответствующего ему элемента окна). Требуемое количество процессоров равно $N_{cpu} = X_{max} \times Y_{max}$, т.е. произведению разрешений экрана по вертикали и горизонтали. При этом фрагмент окна, соответствующий одному пикселу экрана размером $X_e \times Y_e, \text{мм}^2$ имеет площадь $h_x \times h_y, \text{мм}^2$ ($h_x = X_e / X_{max}$, $h_y = Y_e / Y_{max}$).

Основной целью предлагаемой оптимизации метода трассировки лучей является уменьшение количества параллельных процессоров для обработки кадра. Для решения задачи разобьем всю поверхность окна на участки (фреймы) размером $Fr_x \times Fr_y$, где Fr_x - размер фрейма по горизонтали, мм; Fr_y - размер фрейма по вертикали, мм (рис.1). При этом $Fr_x \sim h_x$, $Fr_y \sim h_y$.

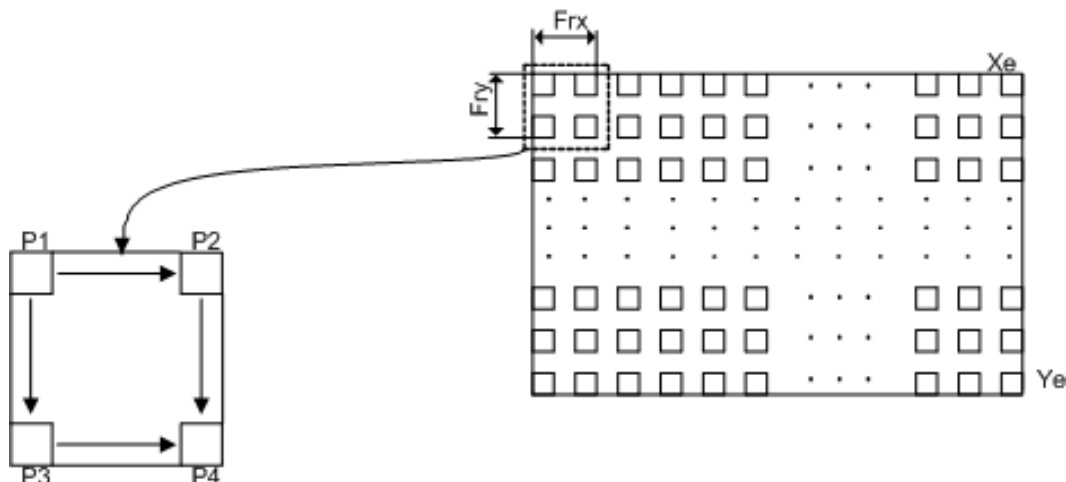


Рисунок 1 – Разбивка экрана на фреймы, пиксельная решетка для интерполяции

Принцип работы. Для каждого фрейма выполняется трассирование по 4-м угловым точкам P1, P2, P3, P4 (очевидно, что точки P2, P4 являются, в свою очередь, точками P1, P3 правого фрейма, а точки P3, P4 – точками P1, P2 нижнего фрейма). Если расхождения между полученными интенсивностями не превышает заданного значения ΔI_{max} (определяется на основе психологического и психофизического восприятия человеческим глазом цветовых оттенков и мощности излучения [3]), то данные передаются в процессор интерполяции. Если отличие существенно, то выполняется трассирование центральной точки фрейма и в процессор интерполяции передается 5 значений интенсивности.

4 Аппаратурная поддержка

Структура системы синтеза для поддержки предложенного метода представлена на рис.2, где rtpu – процессоры трассировки, iu – устройства интерполяции.

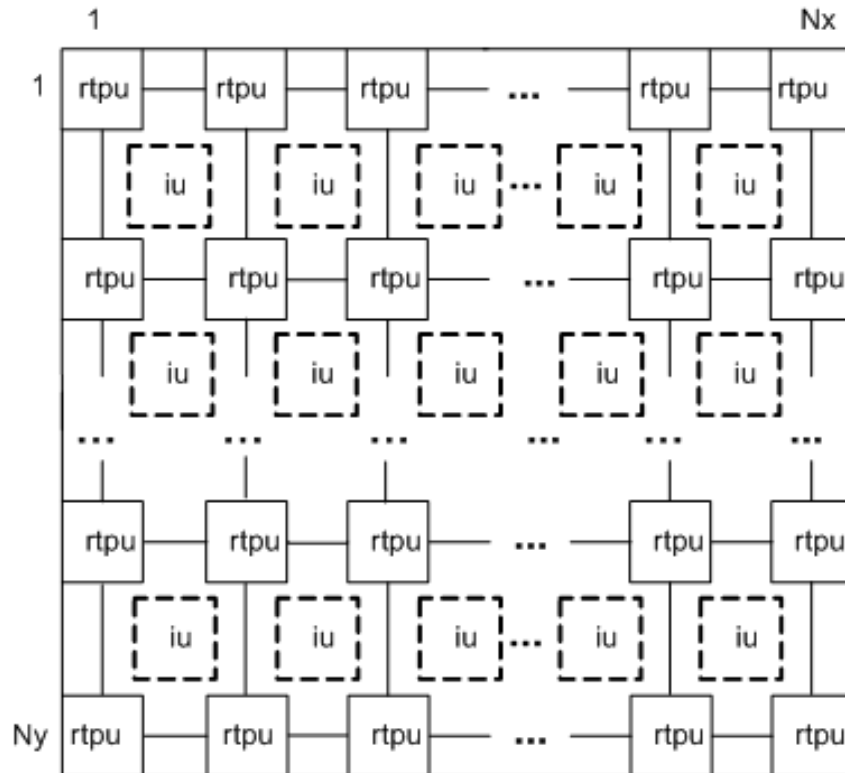


Рисунок 2 - Пример структуры системы синтеза

Количество процессор равно $N_{cpu} = N_x \times N_y$, где $N_x = X_e / Fr_x$, $N_y = Y_e / Fr_y$.

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: первый - трассировка лучей, второй - интерполяция. Такая схема может без труда применяться на более сложных конфигурациях.

5 Заключение

Предстоит решить немало проблем, прежде чем метод трассировки лучей станет достойной альтернативой растеризации в сфере рендеринга в реальном времени. 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. – режим доступа к сайту <http://www.pcgamer.com/>
- [2] NVIDIA Corporation. "Ray tracing today and tomorrow" Октябрь 2002. – режим доступа к сайту <http://www.nvidia.com/techbriefs.html/>
- [3] Новаковский С. В. Цвет в цветном телевидении. М.: Радио и связь, 1988. - 288 с.