Кластерная трассировка лучей в системах визуализации

Д.С. Котов, В.Н.Ланцов

Владимирский государственный университет
humangous@yandex.ru, lantsov@vlsu.ru

В современных системах рендеринга, используемых в САПР, применяются различные подходы к визуализации трёхмерной сцены. Наиболее популярными и перспективными методами построения изображения являются методы трассировки: трассировка лучей, трассировка путей и т.д. В основе любого из современных методов моделирования освещения в трёхмерной сцене лежит метод *трассировки*. При решении уравнения визуализации системе приходится решать ряд задач, которые можно разделить по некоторым признакам и по смысловой нагрузке.

Уравнение рендеринга имеет вид:

$$L(x,\omega) = L_e(x,\omega) + \int_{\Omega} f(x,\omega,\omega')\, L_i(x,\omega')\, (\omega' * n)\, d\omega'$$

В частности, интеграл в уравнении рендеринга можно представить в виде суммы интегралов:

$$L_r(x,\omega) = \int_{\Omega} f(x,\omega,\omega')\, L_i(x,\omega') \cos\theta d\omega' =$$

$$= \int_{\Omega} f(x,\omega,\omega')\, L_e(x,\omega') \cos\theta d\omega' +$$

$$\int_{\Omega} f_s(x,\omega,\omega')(L_d(x,\omega') + L_c(x,\omega')) \cos\theta d\omega' +$$

$$\int_{\Omega} f_d(x,\omega,\omega') L_d(x,\omega') \cos\theta d\omega' +$$

$$\int_{\Omega} f_d(x,\omega,\omega') L_c(x,\omega') \cos\theta d\omega'$$

Первый интеграл - первичное освещение; второй - зеркальное отражение света от поверхности; третий - диффузное рассеивание света; четвёртый интеграл - каустика, эффект диффузного рассеивания света после хотя бы одного зеркального рассеивания.

Будем рассматривать общее освещение в сцене как сумму первичного освещения и всего остального освещения - вторичного. В современных системах рендеринга также популярен подход раздельного просчёта по типам освещения: первичное и вторичное. Иногда каустика просчитывается отдельно. Объясняется это повышенной эффективностью и гибкостью подобного подхода.

Также, просчёт вторичного освещения несёт значительно большую вычислительную нагрузку, нежели чем первичное освещение. Количество трассируемых лучей на стадии вторичного освещения в разы больше, чем на стадии просчёта первичного освещения.

В предлагаемом подходе сделан акцент на новом способе просчёта вторичного освещения. При использовании структур разделения пространства, таких как бинарные и восьмеричные деревья и регулярные сетки, становится возможным упорядочить данные о геометрии в пространстве, что позволяет локализовать поиск ближайшей геометрии и при трассировке луча или группы лучей, попадающих в некоторую область трёхмерной сцены свести наборы тестов на пересечение луча с геометрией практически к поиску лишь по части всего дерева поиска. В таком случае задача сводится к некоторому набору тестов на пересечение лишь для отдельной области сцены.

При трассировке лучей вторичного освещения обычно определяется новая точка пересечения со сценой, из которой далее генерируются несколько лучей вторичного освещения. Рекурсия повторяется до тех пор, как не будет достигнут один из критериев прекращения трассировки. Для любой глубины трассировки для всего множества точек расчёта вторичного освещения имеем по несколько лучей, сгенерированных согласно функции ДФОС материала данной поверхности. В отличие от традиционных подходов, где лучи трассируются рекурсивно один за другим, предлагается лишь генерировать координаты новых лучей, а трассировку выполнить позднее. Принцип отложенной трассировки позволяет сказать, что мы имеем дело с некоторым набором векторов, имеющих свои координаты, где каждый вектор - это луч, сохранённый в виде набора данных, который пока не подвергся операции трассировки. Всё это подмножество таких векторов предлагается разделять на подклассы по признаку направления, где в один класс могут входить все векторы, направление которых не превышает данного телесного угла в общей для всего множества векторов системе координат. Другими словами, группа векторов - это набор векторов, указывающих примерно в одном и том же направлении с погрешностью ε, где ε - ширина телесного угла.

В качестве преимущества, позволяющего получить прирост производительности в рамках данного подхода, выступает тот факт, что для каждого сравнительно небольшого участка поверхности в сцене найдётся некоторый набор лучей, указывающих примерно в одном и том же направлении, если отражательная функция ДФОС одинакова для всей площади данного участка поверхности. Следовательно, вероятно эти лучи или некоторая их часть попадёт на одну и ту же, другую поверхность, при их трассировке. Предлагаемый метод позволяет минимизировать поиск по дереву при проверке на пересечение луча с геометрией сцены, и тем самым для большинства лучей данного участка поверхности скорее всего потребуется лишь некоторая часть треугольников в сцене, которые и будут проверяться в первую очередь на пересечение с лучом. Это позволяет кэшировать необходимые данные, существенно ускоряя работу трассировки.

Ещё одним преимуществом предлагаемого подхода представляется возможность разделения всего подмножества векторов лучей, подлежащих трассировке, на подмножества по признаку их месторасположения в сцене. Такое разделение предлагается, исходя из предположения о том, что место зарождения луча может играть значительную роль в том, куда этот луч попадёт в конечном итоге. Таким образом, предлагается разделять всё подмножество векторов на наборы векторов - кластеры. В качестве критерия отбора в тот или иной кластер становятся два условия:

1) направление вектора (достаточно, чтобы находилось в пределах данного телесного угла)

2) место зарождения луча - начало вектора. Это точка поверхности, откуда трассируется луч. Её координаты известны из каждого предыдущего шага трассировки.

В результате, алгоритм совместим с классическими алгоритмами трассировки, и трассировка здесь производится также - по шагам глубины трассировки. Отличие состоит в том, что внутри шага трассировки применяется другой алгоритм, который в до-

полнение ко всему вышесказанному легко приспособить к многопоточной и многопроцессорной обработке. В частности, современные графические чипы предлагают производить расчёты на своих мощностях, что позволяет получить доступ к производительности, гораздо более высокой, нежели чем на центральных процессорах персональных компьютеров.

Это даёт сразу два очевидных плюса:

1) рост скорости трассировки отдельного кластера, за счёт кэширования данных
2) масштабируемость на многопроцессорных системах.

На рисунке 1 показана иллюстрация того, каким образом и в каком порядке лучи группируются в кластеры.

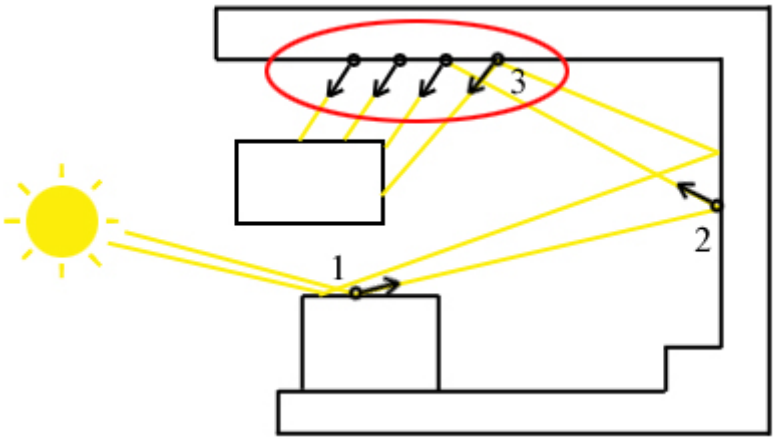

Рисунок 1. Группирование лучей в кластеры.

Таким образом, производится группировка всех лучей для данной глубины трассировки.

Рассмотрим объём работ в классическом подходе к трассировке. Рассматриваем пример, где требуется лишь определить точку пересечения луча со сценой и определить в ней цвет, т.к. этот пример наглядно покажет зависимость изменения времени рендеринга от количества тестов на пересечение с геометрией - одной из наиболее затратных стадий процесса визуализации. Работа алгоритма визуализации в этом случае выглядит похоже на:

```
for(rays=0; rays<=maxRays; rays++){
        for(i = 0; i < maxdepth; i++){
                start_trace_i + traverse_geometry + shade + end_trace_i;
        }
  }
```

Практически в любой известной задаче рендеринга временем исполнения стадий $start\_trace_i$ и end_trace_i можно пренебречь, а стадии traverse_geometry и shade занимают примерно равные доли от общего времени визуализации. Влияние и той и другой стадий на конечный результат очень важно, т.к. в случае нахождения ближайшего пересечения со сценой ключевым фактором является время прохождения этой стадии и точность полученного результата, а на стадии затенения производится работа с материалом и определение конечного цвета данной точки поверхности.

Для среднестатистической сцены предположим, что стадии traverse_geometry и shade равны по времени. Тогда, при сведении времени стадии traverse_geometry к нулю теоретический максимум полученной выгоды может достигнуть 50% прироста скорости общего времени визуализации одного кадра.

В предлагаемом подходе объём работ сокращается за счёт локализации поиска пересечения, и алгоритм функционирования системы визуализации близок к следующему:

```
start_trace0 + traverse_geometry + shade + end_trace0;

for(i = 1; i < maxdepth; i++){
        for(rays=0; rays<=maxRays; rays++){
                start_tracei;
                refToCluster;
                if(clusterAdded) numClusters++;
        }
        for(clusters=0; clusters < numClusters; clusters++){
                traverse_geometry;
        }
        for(rays=0; rays<=maxRays; rays++){
                shade + end_tracei;
        }
 }
```

В данном случае за счёт повторного использования одних и тех же участков дерева поиска геометрии в сцене сразу целым набором лучей отпадает необходимость каждый раз заново полностью проходиться по дереву поиска для каждого луча. В результате суммарное время визуализации одного кадра займёт меньше времени за счёт того, что количество проходов по дереву поиска геометрии будет стремиться к количеству кластеров, используемых для трассировки.

Таким образом, стоит отдельно заметить, что данный подход позволяет разбить задачу визуализации на ряд подзадач с определённой степенью гранулярности. Каждую из таких подзадач можно отдать на выполнение отдельному вычислительному модулю, способному производить арифметические и геометрические операции. Примерами таких модулей могут стать x86 процессоры персональных компьютеров, видео чипы от ATI и nVidia, поддерживающие вычисления общего назначения, а также высокопроизводительные компьютеры (серверы и кластерные системы). Разбиение задачи визуализации на некоторое множество кластеров позволяет гибко управлять загрузкой имеющихся вычислительных мощностей.

Преимуществом данного подхода является его гибкость и универсальность, т.к. его можно применять в любом из известных алгоритмов трассировки: в фотонной трассировке, при любых способах просчёта глобального освещения и т.д. Стоит также отметить, что метод не несёт с собой абсолютно никаких ухудшений качества картинки, поскольку направлен на иную организацию способа расчёта трассировки, без изменения принципа любого из существующих алгоритмов моделирования освещения.

Область применения данного метода: САПР, профессиональные пакеты 3d моделирования и визуализации, системы визуализации в медицине.

Литература

1. Philipp Dutre, Kavita Bala, "Advanced Global Illumination" - A.K.Peters 2006. - 366 p.p.
2. Franklin Crow "The aliasing problem in computer-generated shaded images" - Communications of the ACM, v.20 n.11, , Nov. 1977, 799-805 p.p.
3. Isshiki Tsuyoshi, Kunieda Hiroaki "Efficient Anti-Aliasing Algorithm for Computer Generated Images" - Dept. Electrical and Electronics Engineering, Tokyo Institute of Technology, Proceedings of the 1999 IEEE International Symposium on Volume 4, Jul 1999 - 532 - 535 p.p.
4. Schneider Philipp, Eberly David "Geometric Tools for Computer Graphics" - Morgan Kaufmann, 2002г. - 1056 p.p.
5. Shirley Peter, Keith Morley "Realistic Ray Tracing" - A.K. Peters, 2003г. - 235 p.p.
6. Wann Jensen Henrik "Realistic Image Synthesis Using Photon Mapping" - A.K. Peters, 2009г. - 193 p.p.