

РОССИЙСКАЯ АКАДЕМИЯ НАУК
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ имени М.В. Келдыша

Б.Х. Барладян, А.Г. Волобой, Л.З. Шапиро

**Представление вторичной освещенности
в виде текстур**

Москва

2011

Б.Х. Барладян, А.Г. Волобой, Л.З. Шапиро

Представление вторичной освещенности в виде текстур

Аннотация

В статье рассматриваются один аспект преобразования карт освещенности, являющиеся результатом работы программ оптического моделирования, для сцен, содержащих обычные текстуры, в VRML формат. Описанный алгоритм позволяет корректно учесть рассчитанную вторичную освещенность при визуализации текстурных поверхностей. В конечном итоге разработанное преобразование существенно снижает количество визуализируемых треугольников и памяти, используемой под текстуры, что при визуализации позволяет добиться интерактивной скорости для больших моделей. При этом оно практически не снижает качество итогового изображения.

Ключевые слова: реалистичные изображения, карта освещенности, интерактивная визуализация, вторичная освещенность, VRML.

Boris H. Barladyan, Alexey G. Voloboy, Lev Z. Shapiro

Conversion of global illumination into VRML textures

Abstract

One aspect of export of illumination maps (i-maps) into VRML format is considered in the paper. The aspect concerns mapping of indirect illumination onto textured surfaces. The described algorithm correctly calculates physical values of illumination stored in i-maps for each pixel of resultant texture containing as initial usual texture as well as calculated illumination. The VRML files can be browsed in any available browser. Due to proposed method it is possible to reduce the number of triangles and the volume of texture memory. This leads to interactive speed of rendering of huge models preserving high quality of illuminance distribution.

Keywords: realistic images, illumination maps, interactive visualization, global illumination, VRML.

Работа поддержана грантом Президента РФ «Ведущие научные школы» НШ-8129.2010.9, грантом РФФИ № 09-01-00472, а также фирмой Integra Inc. (Япония).

Электронный вариант статьи с цветными иллюстрациями размещен на сайте http://www.keldysh.ru/pages/cgraph/publications/cgd_publ.htm.

Содержание

1. Введение.....	4
2. Построение карты освещенности в виде текстуры.....	6
2.1.Растрезация.....	7
2.2. Определение освещенности в пикселе	7
2.3. Преобразование яркости в выходные величины.....	8
3. Результаты.....	9
Литература.....	10

1. Введение

Картами освещенности называется сохранение физических значений освещенности в вершинах треугольников для всей сцены. Карты освещенности создаются с помощью алгоритмов моделирования распространения света в различных средах, таких как, например, Монте-Карло прямая трассировка лучей[1], и требуют значительных вычислительных ресурсов. Но при этом использование карт освещенности в современных программах реалистической визуализации позволяет учесть эффекты вторичного освещения. Достижение интерактивности визуализации возможно разными путями. Так, сейчас очень популярным является направление Real Time Ray Tracing (RTRT) с использованием мощных графических процессоров. Однако не все пользователи могут или хотят использовать мощные графические ускорители. Некоторая их часть хотела бы видеть физически корректную реалистичную картинку на своем ноутбуке или на планшете (например, iPad) широкое распространение которых в последний год носит взрывной характер. Для таких устройств важную роль играет представление карт освещенности в стандартизированном формате, загружаемом в независимые системы визуализации, основанные на использовании обычных графических ускорителей, как например [2]. В связи с этим возникает задача экспорта полученных результатов моделирования, т.е. карт освещенности, в системы отображения результатов в достаточно эффективной форме.

Нами были разработаны алгоритмы экспорта карт освещенности в один из популярных форматов представления трехмерных сцен VRML (Virtual Reality Modeling Language) [3]. Стандарт формата VRML позволяет формировать изображения трехмерных сцен как с использованием цвета, определенного в вершинах треугольников, так и с использованием текстур. Однако он не позволяет использовать оба подхода одновременно, т.е. получать изображения в виде произведения текстуры на интерполированное значение цвета (освещенности).

Существенным ограничением этого подхода является не корректное воспроизведение участков сцены с большой освещенностью. Это ограничение обусловлено представлением яркости в виде произведения двух величин: умножение свойства поверхности (текстуры) на интерполированное из вершин значение освещенности может только уменьшить значение яркости в данной точке, но не может увеличить его. Обрезание освещенности по некоторой максимальной величине оператором сжатия динамического диапазона яркостей делается при таком подходе для освещенностей, накопленных в вершинах треугольников. Непосредственно при трассировке лучей это преобразование делается в данной точке для яркости, являющейся (в первом приближении) произведением освещенности на диффузный цвет в данной точке. Поскольку оператор сжатия динамического диапазона яркостей является нелинейным, то результат, в некоторых случаях, может существенно отличаться от желаемого.



Рис. 1. Изображение сцены с использованием произведения текстурного цвета на карту освещенности.

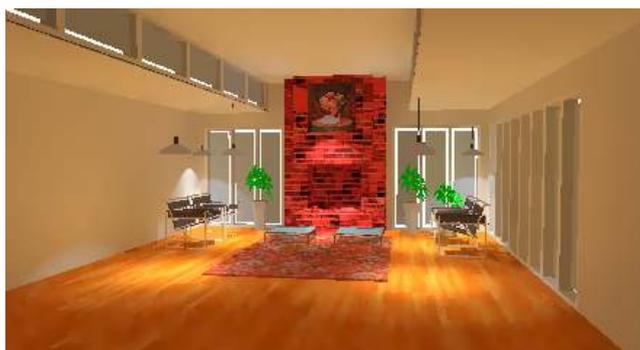


Рис. 2. Изображение сцены. Полученное в результате метода обратной трассировки лучей с применением оператора сжатия динамического диапазона яркостей к результирующему значению яркости.

Приведенные примеры (рис. 1 и 2) показывают, насколько существенным может быть этот недостаток для текстурированных сцен с яркими пятнами от источников света. Яркие пятна практически не воспроизводятся при произведении текстурного цвета на освещенность в вершинах, как это показано на рис. 1. Освещение сцены становится “плоским”, что не соответствует человеческому восприятию реальных сцен. Однако, если оператор сжатия динамического диапазона яркостей применяется к результирующему значению яркости, как в случае метода обратной трассировки лучей, то итоговое изображение содержит корректное изображение ярко освещенных участков (рис. 2).

Таким образом, учитывая, что в большинстве сцен, используемых в архитектуре и промышленности, широко применяются текстуры, единственным практическим подходом является преобразование карт освещенности в виде текстур для всей поверхности сцены (рис. 3). Представление карт освещенности с помощью текстур также позволяет уменьшить размерность треугольной сетки, экспортируемой в VRML, поскольку качество генерируемого изображения теперь зависит в основном не от треугольной сетки, а от текстуры.



Рис. 3. Изображение той же сцены в VRML браузере с использованием текстур для представления карт яркости.

В нашей предыдущей работе [4] были рассмотрены проблемы преобразования текстур для представления карт освещенности в VRML формат и основные способы их решения. В настоящей работе более подробно рассматривается алгоритм преобразования карт освещенности в текстуры, содержащие как текстурную компоненту, так и компоненту рассчитанной освещенности.

2. Построение карты освещенности в виде текстуры

Фактически для каждого текстурированного треугольника необходимо преобразовать карту освещенности, т.е. освещенности, накопленные в вершинах треугольника, в текстурные значения. Разрешение результирующей текстуры выбирается не ниже разрешения обычной текстуры, чтобы избежать ее размывания. Вопрос дальнейшей минимизации размеров текстур был нами рассмотрен в [4].

Каждый треугольник сцены имеет определенные оптические характеристики. Для экспорта в VRML формат используются следующие параметры:

1. color – цвет (в RGB пространстве с диапазоном компонент $[0, 1]$), Этот цвет является общим параметром для диффузного отражения, диффузного пропускания и светимости, если все эти эффекты моделируются для данного участка поверхности.
2. kd – коэффициент диффузного отражения.
3. ktd – коэффициент диффузного пропускания.
4. SelfLum – светимость поверхности в случае, если она является поверхностным источником света в сцене (задается в nit в RGB пространстве),
5. Все эти параметры могут модулироваться обычной текстурой (в RGB пространстве с диапазоном компонент $[0, 1]$).

Для вершин треугольника имеются предварительно рассчитанные значения суммарной (прямой и вторичной) освещенностей с обеих сторон его поверхности (физические значения в lux, в RGB пространстве). Для остальных точек треугольника значения освещенностей определяются линейной интерполяцией значений освещенностей в его вершинах.

Предварительно построенное отображение поверхности сцены на текстурную плоскость определяет текстурные координаты вершин треугольника на текстуре, отображающую карту яркости. Отображение остальных точек треугольника на текстуру также определяется линейной интерполяцией текстурных координат в его вершинах.

Алгоритм построения текстуры, содержащую освещенность, для данного треугольника состоит из следующих шагов:

1. Определение всех пикселей текстуры, которые пересекают отображение треугольника на текстуру.
2. Отображение найденных пикселей (точнее их центров) на треугольник и расчет освещенности в данной точке треугольника.
3. Преобразование физических величин освещенности в RGB пространство, если пользователь хочет получить реалистичное изображение, или в искусственные цвета (Color Contours), если пользователь хочет визуализировать распределение освещенности.

2.1 Растеризация

Для растеризации полагаем, что одна из сторон (основание) проекции треугольника на текстурную плоскость параллельна оси X текстуры. Если это не так, то текстурный треугольник всегда можно разбить на два треугольника, удовлетворяющих этому условию.

Для этой стороны выделяется ряд пикселей, соответствующий ей. Ряд располагается вдоль оси X текстуры. Граничными пикселями для него будут пиксели, в которые попадают вершины основания текстурного треугольника.

Далее переходим к следующему ряду, сдвигаясь на один пиксель по оси Y текстуры в направлении вершины противоположной основанию. Граничные пиксели для этого ряда определяются из величины смещения по оси Y и наклона боковых сторон треугольника.

Так, передвигаясь по рядам пикселей (вдоль оси Y) до вершины, противоположной основанию, определяются все требуемые пиксели на текстуре. Для этих пикселей и будет рассчитано значение яркости текстуры с учетом освещенности.

2.2 Определение освещенности в пикселе

Точка, соответствующая центру выделенного выше пикселя отображается на пространственный треугольник. Если эта точка лежит вне текстурного треугольника, то она сдвигается параллельно оси X текстурной плоскости до стороны текстурного треугольника. Для первой полосы делается также сдвиг параллельно оси Y в направлении вершины противоположной основанию, если необходимо.

Пусть (a, b, c) - барицентрические координаты этой точки в системе координат треугольника. Тогда текстурные координаты этой точки выражаются в виде линейной комбинации координат вершин текстурного треугольника (p_1, p_2, p_3)

$$p = a * p1 + b * p2 + c * p3$$

В силу линейности преобразования треугольника на текстурную плоскость таким же выражением (с такими же коэффициентами) будут выражаться и координаты пространственной точки через координаты вершин пространственного треугольника.

$$P = a * P1 + b * P2 + c * P3$$

Значения освещенности в заданной точке треугольника с каждой стороны определяется по следующей формуле:

$$I_{Lpos} = a * I_{Lpos1} + b * I_{Lpos2} + c * I_{Lpos3}$$

$$I_{Lneg} = a * I_{Lneg1} + b * I_{Lneg2} + c * I_{Lneg3},$$

где

- I_{Lpos} , I_{Lneg} – значения освещенности в точке треугольника с положительной и отрицательной его стороны,
- a , b , c – барицентрические координаты точки, определенные выше,
- I_{Lpos1} , I_{Lpos2} , I_{Lpos3} – значения освещенностей в вершинах треугольника с положительной стороны,
- I_{Lneg1} , I_{Lneg2} , I_{Lneg3} – значения освещенностей в вершинах треугольника с отрицательной стороны.

Итоговый расчет яркости для пикселя производится по следующей формуле:

$$L = color * (kd * I_{Lpos} + ktd * I_{Lneg}) / PI + SelfLum,$$

где

- L – значение яркости,
- $color$ – цвет поверхности в данной точке, который для текстурированной поверхности зависит от точки,
- kd , ktd , $SelfLum$ – определенные выше оптические характеристики поверхности.

2.3 Преобразование яркости в выходные величины

Полученные величины яркости должны для визуализации быть преобразованы либо в RGB пространство монитора, либо в искусственные цвета (Color Contours) в зависимости от задачи, решаемой пользователем.

В первом случае для этой цели используется оператор сжатия динамического диапазона яркостей (Tone Mapping Operator), описанный в работе [5].

Во втором случае фотометрические значения яркости вычисляются, используя стандартную Y CIE кривую. Затем полученные значения преобразуются в искусственные цвета, используя таблицу соответствия диапазонов энергии, заданную пользователем.

3. Результаты

Предложенный алгоритм построения треугольников с текстурами, содержащими физическую освещенность, был использован в программе экспорта сцен с рассчитанной освещенностью в VRML формат. Полученные текстуры минимизируются и упаковываются в атласы с оптимизацией суммарного размера текстур при сохранении качества реалистичного изображения. Программа была опробована на сценах размерности в несколько млн. треугольников. Количество исходных текстур (компонент) может составлять десятки тысяч. Количество создаваемых атласов составляло 40-50 при разрешении каждого 2048x2048. Время работы алгоритма упаковки освещенности в текстуры составило несколько процентов от общего времени преобразования сцены в VRML формат и может считаться несущественным.



Рис. 4. Изначальная карта освещенности, визуализированная с помощью трассировки лучей.



Рис. 5. Та же карта освещенности, экспортированная в VRML формат с упакованными текстурами.

На приведенных рисунках представлено помещение, освещенное большим количеством искусственных источников света. Практически все освещение не прямое (вторичное), источники света загорожены плафонами или направлены на потолок для создания «мягкого» освещения. Для реалистичной визуализации этой сцены учет вторичного освещения критичен. На рис. 4 показана изначально рассчитанная карта освещенности, визуализированная с

помощью трассировки лучей. На рис. 5 представлена визуализация полученного в результате экспорта VRML файла с упакованными в атласы текстурами. Для визуализации использовался VRML браузер Cortona [2].

Из сравнения рисунков видно, что качество представления освещенности практически не ухудшилось. Однако создание первого изображения потребовало 46 секунд (разрешение 1024x768, время указано для компьютера Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz). В то же время полученный VRML файл может быть визуализирован в интерактивном режиме, и пользователь может путешествовать по помещению в реальном времени.

Литература

[1] Monte Carlo ray tracing

[2] Cortona3D Viewer

<http://www.parallelgraphics.com/products/cortona/>

[3] VRML specification

<http://www.web3d.org/x3d/vrml/>

[4] Б.Х. Барладян, А.Г. Волобой, Л.З. Шапиро. Оптимизация представления карт освещенности и яркости для их интерактивной визуализации. Труды 19-й международной конференции ГрафиКон, Москва, 2009, стр. 267-270.

[5] Б.Х. Барладян, А.Г. Волобой, В.А. Галактионов, Э.А. Копылов. Эффективный оператор сжатия динамического диапазона яркостей. "Программирование", № 5, с. 35-42, 2004.