

# Оптимизация представления карт освещенности и яркости для их интерактивной визуализации.

Б.Х. Барладян, А.Г. Волобой, Л.З. Шапиро  
Институт прикладной математики им. М.В.Келдыша РАН, Москва

## Аннотация

В статье рассматриваются различные способы представления карт освещенности и яркости, являющиеся результатом работы программ оптического моделирования. Предложен способ экспорта карт освещенности в VRML формат с упаковкой текстур. Благодаря этому экспорту возможно существенное снижение количества визуализируемых треугольников и памяти, используемой под текстуры, что позволяет добиться интерактивной скорости визуализации больших моделей практически при сохранении качества изображения.

**Ключевые слова:** реалистичные изображения, карта освещенности, интерактивная визуализация, атласы, VRML.

## 1. ВВЕДЕНИЕ

Получение карт освещенности и яркости в современных программах оптического моделирования, обычно, занимает значительное время. Это время особенно существенно, если речь идет о моделировании поверхностей со сложными оптическими свойствами, описываемыми двунаправленными функциями отражения и преломления, в условиях непрямого освещения. В то же время полученные результаты моделирования освещения хотелось бы визуализировать в интерактивном режиме, предоставить возможность пользователю передвигаться по сцене. Для этого карты освещенности должны быть загружаемы в независимые системы визуализации, основанные на использовании графических ускорителей, как например [1]. Другим актуальным направлением интерактивной визуализации является трассировка лучей в режиме реального времени [2, 3, 4, 5]. Возможность использовать рассчитанные заранее результаты моделирования освещенности существенно повышает качество и реалистичность визуализации. В связи с этим возникает задача экспорта полученных результатов моделирования, т.е. карт освещенности и яркости, в системы отображения результатов в достаточно эффективной форме.

В нашей системе оптического моделирования [6] вторичная освещенность вычисляется на треугольной сетке в виде значений освещенности в вершинах треугольников. Для получения точного решения используется достаточно мелкая, адаптивно создаваемая сетка. Дополнительное разбиение такой сетки производится в областях, где имеет место большой градиент освещенности. Таким образом, исходные результаты моделирования представляются в виде значений освещенностей в вершинах треугольников на детальной треугольной сетке. Следует также учитывать, что для получения фотореалистичных изображений в системах архитектурной или промышленной визуализации широко используются текстуры, модулирующие диффузное

отражение поверхностей. Яркость результирующего изображения в данной точке должна вычисляться как произведение освещенности видимой точки поверхности на коэффициент диффузного рассеяния в ней. Освещенность в точке поверхности вычисляется путем интерполяции освещенности, сохраненной в вершинах треугольника, а значение коэффициента диффузного рассеяния в точке задается материалом поверхности и может модулироваться текстурой.

Одним из популярных, широко используемых в Интернете, и стандартизированных форматов представления трехмерных сцен является VRML (*Virtual Reality Modeling Language*), для которого существует большое количество бесплатных и коммерческих браузеров. Поэтому разработка экспорта рассчитанных карт яркости в VRML формат позволяет существенно расширить возможности визуализации результатов моделирования освещенности. VRML формат 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). Некоторые браузеры поддерживают такое расширение, но в общем случае на это нельзя рассчитывать. Таким образом, учитывая, что в большинстве сцен, используемых в архитектуре и промышленности, широко применяются текстуры, единственным практическим подходом является представление карт освещенности и яркости с помощью текстур для всей поверхности сцены. Представление карт освещенности яркости с помощью текстур также позволяет уменьшить размерность треугольной сетки, экспортируемой в VRML. Поскольку генерируемое изображение теперь зависит в основном не от треугольной сетки, а от текстуры, то можно использовать существенно более грубую треугольную сетку.

## 2. ОСНОВНЫЕ ПРОБЛЕМЫ И ПУТИ ИХ РЕШЕНИЯ

Основной проблемой использования текстур для представления карт освещенности и яркости в VRML формате является проблема размерности. Графические ускорители, используемые всеми браузерами, поддерживающими VRML формат, имеют ограничения по объему используемых текстур. При превышении этого объема часть текстур перестает воспроизводиться, либо скорость воспроизведения существенно падает. Следует учесть, что при использовании обычных текстур в представлении сцены широко применяется их повторяемость. Небольшое по размеру изображение, например, поверхность дерева, обоев или, даже часть фасада дома повторяется много раз для представления большой поверхности – стены или здания. Использование этой повторяемости для

представления карт освещенности и яркости становится невозможным, поскольку одинаковые (повторяющиеся) части геометрии, как правило, имеют разную освещенность. В силу этих причин размерность используемых текстур возрастает многократно.

Другой причиной, требующей высокого разрешения результирующих текстур, является использование в сцене обычных текстур высокого разрешения. Результирующая текстура объединяет карту освещенности и обычную текстуру, поэтому ее разрешение должно быть достаточным, чтобы без искажений передать рисунок изначальной обычной текстуры.

И, наконец, третьей причиной, требующей высокого разрешения результирующих текстур, является наличие в сцене участков с большим градиентом яркости. Изображение на этих участках будет смазано, если разрешение текстуры будет недостаточным. На рис.1 представлено изображение двери в салоне самолета при недостаточном разрешении текстуры. Проблема вызвана тенью на двери, которая при недостаточном разрешении текстуры размывается и происходит так называемая "утечка света" (тени в данном случае). На рис.2 представлено изображение двери со снятым наличником, где хорошо видна исходная тень, размывтая на рис.1. На рис.3 представлено изображение той же двери в самолете, полученной при существенном увеличении разрешения текстуры. Проблема с "утечкой света" практически исчезла.



**Рис. 1.** Изображение двери в самолете при недостаточном разрешении текстуры.



**Рис. 2.** Изображение двери в самолете со снятым наличником.



**Рис. 3.** Изображение двери в самолете при увеличенном разрешении текстуры.

Этот пример иллюстрирует основную проблему построения текстур для представления карт яркости в VRML формате. Высокое качество изображений требует использования текстур высокого разрешения, в то время как максимальный объем текстур, который может использоваться графическими ускорителями, ограничен.

Следующей проблемой является ограничение не только полного объема текстур, но также их количества и максимальной размерности отдельных текстур. Эти ограничения зависят от используемого графического ускорителя и драйвера. Типичные значения, при которых графический ускоритель может использоваться эффективно, – это несколько десятков текстур при разрешении отдельной текстуры не более 4096x4096 точек. Была реализована возможность контролировать эти параметры в процессе экспорта карт освещенности. Таким образом, пользователь может найти оптимальное соотношение между качеством итогового изображения и эффективностью его визуализации для достижения интерактивной скорости.

Следует отметить, что при подготовке текстур приходится разбивать всю геометрию сцены на компоненты – небольшие части, для которых можно определить такое отображение поверхности на плоскость текстуры, что искажения будут находиться в заданных пределах. Наиболее практичным и достаточно эффективным подходом для решения этой задачи является разбиение всей геометрии сцены на группы связанных между собой треугольников, нормали которых отличаются от выбранного направления не больше чем на заданную величину. Для каждой такой группы строится отдельная текстура. В результате этого шага получается очень большое количество текстур, необходимых для полного покрытия сцены. Для реальных сцен это число может составить тысячи или даже десятки тысяч. Многие из этих текстур могут иметь небольшие размеры, поэтому наиболее разумным решением является упаковка некоторого множества таких текстур в одну – так называемый Атлас. Microsoft DirectX SDK [7] предоставляет соответствующие процедуры, позволяющие эффективно упаковывать множество текстур в один такой атлас. Как правило, разрешение атласа по каждому измерению выбирается кратным степени двух ( $2^n$ ), что позволяет избежать дополнительных расходов памяти при передаче текстур в графический ускоритель. Полное количество атласов и их конкретное разрешение выбирается пользователем в зависимости от требования к качеству отображения карт освещенности и возможностей оборудования, в первую очередь графического ускорителя. Задача программы, экспортирующей результаты моделирования освещенности в VRML формат, состоит в обеспечении максимально возможного качества экспортируемых карт освещенности при заданных ограничениях – количестве и разрешении атласов.

### 3. ОПТИМИЗАЦИЯ КАЧЕСТВА ТЕКСТУРЫ ПРИ ЗАДАННЫХ ОГРАНИЧЕНИЯХ

Оптимизация качества экспортируемых карт освещенности происходит в несколько этапов.

На первом этапе вся геометрия сцены (треугольная сетка) разбивается на компоненты так, чтобы для каждой из них

можно было определить отображение поверхности на плоскость текстуры, и искажения лежали бы в заданных пределах. В нашей системе используется разбиение на группы связанных между собой треугольников, поскольку текстура, представляющая карту освещенности, должна быть разрезана вдоль границ связанной компоненты. Здесь возможны скачки яркости. Для уменьшения искажения отображения в группу собираются треугольники, нормали у которых отличаются от выбранного направления не больше чем на заданную величину.

На втором этапе для каждой такой группы строится карта яркости – текстура заданного разрешения в RGB пространстве. Для построения текстуры может использоваться как линейный, так и не линейный операторы сжатия динамического диапазона яркостей [8]. Исходное разрешение текстуры выбирается достаточно высоким (~1-0.1см в физических единицах на реальных поверхностях сцены) так, чтобы возможно было получить изображения высокого качества. Если для данной группы треугольников в сцене назначена обычная текстура, то разрешение результирующей текстуры выбирается не ниже разрешения обычной текстуры, чтобы избежать ее размывания.

На третьем этапе происходит сжатие построенных текстур до минимально допустимого значения. Вычисляется среднее значение яркости в RGB пространстве для всех последовательных четверок текстурных пикселей. Разрешение текстуры можно уменьшить в четыре раза (в два раза по каждому измерению), если для каждой четверки это среднее значение отличается от значения каждого пикселя из четверки меньше заданного пользователем порога. Процедура повторяется столько раз, сколько это допустимо. Использование такой процедуры сжатия существенно сокращает необходимый объем памяти потому, что в реальных сценах всегда существуют не освещенные поверхности (внутренние поверхности реальных объектов). Для них текстуры можно сжать до минимального (16x16) размера.

На четвертом этапе построенные текстуры упаковываются в атласы с использованием DirectX [7]. Для этого множество построенных текстур необходимо разбить на группы, каждая из которых будет упакована в отдельный атлас. Количество групп должно быть меньше или равно максимальному числу атласов, заданному пользователем. В простейших случаях мы можем просто собрать в одну группу текстуры с суммарным разрешением примерно равным максимальному разрешению атласа. Но в действительности это возможно только для небольших и простых сцен. Для больших реальных сцен суммарный объем (разрешение) текстур, построенных на третьем этапе, существенно превосходит суммарное разрешение максимально допустимого количества атласов. Поэтому в реальности приходится использовать меньшее разрешение текстур для того, чтобы вписаться в заданные ограничения по размерности. Желательно при этом уменьшить разрешение всех текстур примерно одинаково, чтобы их качество было бы равномерно по всей сцене. Для этого желательно разбить исходное множество текстур на группы с приблизительно одинаковым суммарным разрешением, каждая из которых затем будет упакована в отдельный атлас. Для этой цели используется следующий алгоритм:

(1) Все текстуры, построенные на этапе 3, сортируются в порядке убывания их размерности (количеству пикселей).

(2) Выбирается некоторая величина  $A$ , определяющая максимальную суммарную размерность текстур, которые можно собрать в одну группу. Максимальное разрешение атласа используется в качестве начального приближения для этой величины.

(3) Мы добавляем текстуры в заданную группу до тех пор, пока их суммарное разрешение не превысит заданную величину  $A$ . Группа начинает формироваться с наибольших текстур. Если данная текстура не может быть добавлена в формируемую группу, то мы пытаемся добавить следующую, меньшую по размеру. Как результат этой процедуры некоторое количество групп  $N_a$  будет получено.

(4) Процедура, описанная в п. 1-3, фактически определяет некоторую функцию  $N_a = F(A)$ , которая задает результирующее количество групп (атласов) для максимального суммарного разрешения группы  $A$ . Максимальное количество атласов задано пользователем, поэтому для вычисления максимального суммарного разрешения группы  $A$  необходимо решить соответствующее нелинейное уравнение  $F(A) = N$ , где  $N$  – максимальное количество атласов, заданное пользователем. Решение находится методом деления отрезка пополам, поскольку функция  $F(A)$  не возрастающая ( $F(A_1) \leq F(A_2)$  при  $A_1 > A_2$ ).

При решении этого уравнения определяется состав групп исходных текстур для каждого из будущих атласов.

(5) На пятом этапе каждая из определенных выше групп текстур упаковывается в отдельный атлас. При упаковке определяются текстурные координаты внутри данного атласа для всех треугольников, которые к нему отнесены. После этого создается результирующая текстура для данного атласа, т.е. текстура, представляющая карту освещенности или яркости для той части поверхности сцены (тех треугольников), которая отнесена к данному атласу. Текстура строится путем растеризации карты освещенности для каждого из треугольников на текстуре атласа. При этом в каждом пикселе текстуры суммируются значения освещенностей (либо яркостей) от всех треугольников поверхности сцены, чьи проекции на текстурную плоскость его пересекают. Процедура повторяется несколько раз со сдвигом текстурных координат вершин растеризуемых треугольников для сглаживания результирующей текстуры атласа.

#### 4. РЕЗУЛЬТАТЫ

Предложенная схема построения карт освещенности и яркости с упаковкой их в атласы была опробована на сценах размерности 1-3 млн. треугольников. Количество исходных текстур (компонент) составляло ~10000. Количество создаваемых атласов составляло ~50 при разрешении каждого 2048x2048.

На приведенных рисунках представлен салон самолета Boeing. Практически все освещение салона – не прямое. На рис. 4 показана изначально рассчитанная карта освещенности, сохраненная в вершинах треугольников. На рис. 5 представлена визуализация полученного в результате

экспорта VRML файла с упакованными в атласы текстурами. Для визуализации использовался VRML браузер Cortona [1].

Из сравнения рисунков видно, что качество представления освещенности практически не ухудшилось. В то же время полученный VRML файл может быть визуализирован в интерактивном режиме, и пользователь может путешествовать по салону самолета в реальном времени.



Рис. 4. Изначальная карта освещенности салона самолета.



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

Работа поддержана 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](http://www.keldysh.ru/pages/cgraph/publications/cgd_publ.htm).

## 5. ЛИТЕРАТУРА

[1] <http://www.parallelgraphics.com/products/cortona/>

[2] Andreas Dietrich, Ingo Wald, Holger Schmidt, Kristian Sons, and Philipp Slusallek. Realtime Ray Tracing for Advanced Visualization in the Aerospace Industry Proceedings of the 5th Paderborner Workshop Augmented and Virtual Reality in der Produktentstehung, 2006 (<http://graphics.cs.uni-sb.de/~dietrich/>)

[3] Ingo Wald and Philipp Slusallek. State-of-the-Art in Interactive Ray-Tracing in State of the Art Reports, EUROGRAPHICS-2001, Manchester, United Kingdom, September 3-7, 2001, pp. 21-42.

[4] Ingo Wald, Philipp Slusallek, and Carsten Benthin. Interactive distributed ray tracing of highly complex models. In Proceedings of the 12th EUROGRAPHICS Workshop on Rendering, June 2001, London, pp. 274 -285.

[5] Ingo Wald, William R. Mark, Johannes Günther, Solomon Boulos, Thiago Ize, Warren Hunt, Steven G. Parker, Peter Shirley. State of the Art in Ray Tracing Animated Scenes in State of the Art Reports, EUROGRAPHICS-2007, <http://www.sci.utah.edu/~wald/Publications/2007///Star07/download/star07rt.pdf>

[6] А.Г. Волобой, В.А. Галактионов, Машинная графика в задачах автоматизированного проектирования. «Информационные технологии в проектировании и производстве» № 1, 2006, стр. 64-73.

[7] <http://msdn2.microsoft.com/en-us/library/bb206321.aspx>

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

## Abstract

Luminance and illuminance representation via illumination maps (i-maps) or shaded texture is considered. The method of export of i-maps calculated by optical simulation program into VRML format with texture compressing and packing is described. Different optimization methods of quality and size of these textures are considered. 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, atlas, VRML.*

## Authors:

Boris H. Barladyan, PhD, senior researcher, Keldysh Institute for Applied Mathematics RAS.

E-mail: [obb@gin.keldysh.ru](mailto:obb@gin.keldysh.ru)

Alexey G. Voloboy, PhD, senior researcher, Keldysh Institute for Applied Mathematics RAS.

E-mail: [voloboy@gin.keldysh.ru](mailto:voloboy@gin.keldysh.ru)

Lev Z. Shapiro, PhD, senior researcher, Keldysh Institute for Applied Mathematics RAS.