

Быстрая визуализация освещенности салона самолета

*Б.Х. Барладян,
снс, к.т.н., доцент, obb@gin.keldysh.ru
А.Г. Волобой,
снс, к.ф.-м.н., доцент, voloboy@gin.keldysh.ru
В.А. Галактионов,
зав. отдела, д.ф.-м.н., проф., vlgal@gin.keldysh.ru
Л.З. Шапиро,
снс, к.ф.-м.н., доцент
ИГМ им. М.В. Келдыша РАН, Москва*

Аннотация

В статье рассматриваются способ представления освещенности салона самолета, являющиеся результатом работы программ оптического моделирования. Предложен способ экспорта этой 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.

Abstract

Representation of simulated illumination of aircraft cabin is considered. The method of export of illumination 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. 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.

1. Введение.

Проектирование правильного освещения салона самолета и кабины летчиков является важным и достаточно затратным этапом создания летательного аппарата. Пассажирские салоны самолета разрабатываются по индивидуальному заказу для крупных авиационных компаний, и их проектирование и производство составляют большую часть стоимости самолета. При разработке кабины летчиков важным фактором является аспекты безопасности: хорошая читаемость приборов, отсутствие отблесков и ярких отражений. Поэтому очень важным моментом проектирования становится физически корректный расчет освещенности будущего салона или кабины.

Расчет корректной освещенности в современных программах оптического моделирования[1], обычно, занимает значительное время. Это время

особенно существенно, если речь идет о моделировании поверхностей со сложными оптическими свойствами, описываемыми двунаправленными функциями отражения (применяемые для задания свойств ткани сидений), в условиях, когда практически все освещение в салоне – не прямое.

В то же время полученные результаты моделирования освещения хотелось бы визуализировать в интерактивном режиме, предоставить возможность пользователю виртуально передвигаться по сцене. Для этого рассчитанная освещенность должна быть сохранена в специальных картах освещенности. При этом эти карты должны быть представлены в виде, загружаемом в независимые системы визуализации, основанные на использовании графических ускорителей [2]. Другим актуальным направлением интерактивной визуализации является трассировка лучей в режиме реального времени [3]. В процессе расчета освещенность сохраняется на треугольной сетке сцены. Однако такая сетка не может быть эффективно визуализирована из-за большого числа элементов. Поэтому возникает задача экспорта полученных результатов моделирования (карт освещенности) в системы визуализации в более эффективной форме.

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 освещенности в VRML формат позволяет существенно расширить возможности визуализации результатов моделирования освещенности. VRML формат позволяет формировать изображения трехмерных сцен с использованием текстур. Таким образом, учитывая, что в сценах, описывающих модель самолета, широко применяются текстуры, единственным практическим подходом является представление карт освещенности и яркости с помощью текстур для всей поверхности сцены.

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 недостаточно. 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 той же двери в самолете, полученной при существенном увеличении разрешения текстуры. Проблема с "утечкой света" практически исчезла.



Рис.1. Проблемы при недостаточном разрешении текстуры.



Рис.2. Изображение при увеличенном разрешении текстуры.

Поэтому 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.

2. Оптимизация качества текстур.

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. Исходное разрешение текстуры выбирается достаточно высоким ($\sim 1-0.1$ см 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.

На четвертом этапе построенные текстуры упаковываются в атласы с использованием DirectX[3]. Для этого множество построенных текстур необходимо разбить на группы, каждая из которых будет упакована в отдельный атлас. 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 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 **Na** будет получено.

(4) Процедура, описанная в п. 1-3, фактически определяет некоторую функцию **Na = 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(A₁) ≤ F(A₂)** при **A₁ > A₂**).

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 путем растеризации карты освещенности для каждого из треугольников на текстуре атласа. При этом в каждом пикселе текстуры суммируются значения освещенно-

стей (либо яркостей) от всех треугольников поверхности сцены, чьи проекций на текстурную плоскость его пересекают. Процедура повторяется несколько раз со сдвигом текстурных координат вершин растеризуемых треугольников для сглаживания результирующей текстуры атласа.

3. 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, создаваемых атласов составляло ~50 при разрешении каждого 2048x2048.

На приведенных рисунках представлен салон самолета Boeing. Практически все освещение салона – не прямое. На левом рис. 3 показана изначально рассчитанная карта освещенности, сохраненная в вершинах треугольников. На правом рис. 3 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. В то же время полученный VRML файл может быть визуализирован в интерактивном режиме, и пользователь может путешествовать по салону самолета в реальном времени.



Рис. 3. Изначальная карта освещенности салона Boeing и экспортированная в VRML формат с упакованными текстурами.

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

Литература

- [1] А.Г. Волобой, В.А. Галактионов, Машинная графика в задачах автоматизированного проектирования. «Информационные технологии в проектировании и производстве» № 1, 2006, стр. 64-73.
- [2] <http://www.parallelgraphics.com/products/cortona/>
- [3] Andreas Dietrich, Ingo Wald, Holger Schmidt, Kristian Söns, 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/>)
- [4] <http://msdn2.microsoft.com/en-us/library/bb206321.aspx>