

ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ ПОСТРОЕНИЯ ФОТОРЕАЛИСТИЧНЫХ ИЗОБРАЖЕНИЙ

Жданов Дмитрий Дмитриевич

Санкт-Петербургский Национальный Исследовательский Университет ИТМО, Санкт-Петербург

Ершов Сергей Валентинович, Поздняков Сергей Георгиевич

Институт Прикладной Математики им. М.В. Келдыша РАН, Москва

Аннотация

Данная статья посвящена разработке объектно-ориентированной компьютерной модели построения фотореалистичных изображений геометрически и оптически сложных сцен. В работе проводится классификация объектов сцены и предлагается универсальный программный интерфейс, достаточный для проведения физически корректных светотехнических вычислений и построений фотореалистичных изображений. Кроме того, иерархическая модель объектов сцены и оптических элементов позволяет на уровне стохастического лучевого моделирования строить геометрические и оптические модели практически не ограниченной сложности. В качестве примера приводится результат моделирования изображения, наблюдаемого на жидкокристаллическом дисплее автомобиля при разных условиях внешнего освещения.

Ключевые слова: Сцена, объект сцены, оптический элемент, программный интерфейс, рендеринг, стохастическое лучевое моделирование, фотореалистичное изображение.

1. ВВЕДЕНИЕ

Решение задачи построения физически корректного изображения включает в себя два основных программно-методических компонента. Во-первых, это методические, алгоритмические и программные модели трассировки лучей и построения изображения сцен с потенциально сложными компонентами. Во-вторых, это методические, алгоритмические и программные модели объектов сцены, используемые в программных алгоритмах трассировки лучей и решениях уравнения рендеринга, то есть базовые компоненты модели рендеринга. Оба программных компонента должны обеспечивать эффективную и физически корректную трассировку лучей в сценах, содержащих десятки и сотни миллионов геометрических объектов, допускающих сложные условия освещения и имеющих сложные оптические свойства, такие как двунаправленное рассеивание, поляризационное рассеивание, объемное рассеивание, двойное лучепреломление и т.п.

Однако эффективность и физическая корректность моделей отдельных программных компонентов рендеринга не гарантирует высокую эффективность и физическую корректность общей программной реализации рендеринга. Обязательным условием общей высокой эффективности и физической аккуратности рендеринга является согласованность программных интерфейсов всех объектов, формирующих общий программный продукт. Отсутствие согласованности может привести к потере физической корректности или эффективности вычислений. Например, если модель рассеивания света на поверхности обеспечивает физически корректное спектрально-поляризационное преобразование излучения, а программа трассировки лучей не позволяет передать состояние поляризации света от одного объекта сцены до другого, то физическая корректность модели данного объекта просто теряется.

Правильное проектирование программных интерфейсов должно учитывать сложную физическую организацию сцены. Сцена может содержать объекты, организация и программный интерфейс которых отличается от организации основных объектов сцены. Это так называемые оптические элементы [6], которые для сцены являются «черным ящиком», видимым только посредством определенного программного интерфейса.

Данная работа ориентирована главным образом на проектирование программных интерфейсов, связанных с вычислительной частью, оставляя за кадром вопросы, связанные с подготовкой, хранением и графическим представлением данных в пользовательском интерфейсе. Кроме того, вопросы поляризации (поляризационного рассеивания и распространения света) рассматриваются только для прямой трассировки лучей (от источника света до изображения).

2. ПРИМЕНЕНИЕ ДВУНАПРАВЛЕННОЙ СТОХАСТИЧЕСКОЙ ТРАССИРОВКИ ЛУЧЕЙ ДЛЯ РЕШЕНИЯ УРАВНЕНИЯ РЕНДЕРИНГА

Разработка программных интерфейсов рендеринга должна базироваться на выбранном подходе решения уравнения рендеринга. Уравнение рендеринга определяет яркость в направлении наблюдения как сумму собственной яркости наблюдаемого объекта и яркости, рассеянной данным объектом в направлении наблюдения [4]. Для статических сцен уравнение рендеринга, определяющее яркость объекта в точке \vec{p} , направлении \vec{v} и для цветовой компоненты c , можно записать следующим образом:

$$L(\vec{p}, \vec{v}, c) = \tau(\vec{p}, \vec{v}, c) \left(L_0(\vec{p}, \vec{v}, c) + \frac{1}{\pi} \int_{4\pi} BSDF(\vec{p}, \vec{v}, \vec{v}', c) E_\omega(\vec{p}, \vec{v}', c) d\omega \right) \quad (1)$$

где:

$L_0(\vec{p}, \vec{v}, c)$ – собственная яркость объекта в точке наблюдения,

$\tau(\vec{p}, \vec{v}, c)$ – пропускание (прозрачность) среды между наблюдателем и точкой наблюдения,

$BSDF(\vec{p}, \vec{v}, \vec{v}', c)$ – функция двунаправленного рассеивания от источника освещения в направлении \vec{v}' на наблюдателя,

$E_\omega(\vec{p}, \vec{v}', c)$ – локальная освещенность объекта в точке наблюдения по направлению \vec{v}' , созданная источником света в телесном угле $d\omega$.

Основным методом решения уравнения рендеринга является метод трассировки лучей. В лучевом подходе интегрирование по сфере выполняется посредством трассировки лучей в направлении всех возможных источников излучения. На данный момент наиболее эффективным и универсальным решением уравнения (1) является двунаправленная стохастическая трассировка лучей, когда составляющая

первичной яркости вычисляется на трассах обратных лучей, а составляющая вторичной яркости – в областях пересечения трасс прямых и обратных лучей. Кроме того, данное решение является физически корректным, поскольку учитывает условия освещения и наблюдения в точках интегрирования яркости и в состоянии корректно выполнить интегрирование бесконечного цикла лучей на трассе от источников освещения до наблюдателя [1-3].

Двунаправленная стохастическая трассировка лучей схематично представлена на рис. 1.

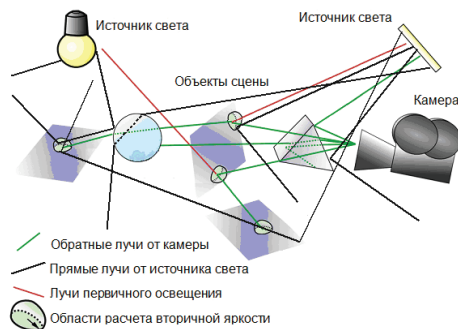


Рис 1: Двунаправленная стохастическая трассировка лучей.

Циклы прямой и обратной трассировки лучей повторяются до тех пор, пока требуемое качество изображения не будет достигнуто.

Естественно, приведенная схема показывает лишь общую картину процесса построения изображения, однако она позволяет выделить основные программные компоненты модели рендеринга, участвующие в построении изображения:

- Программные модули трассировки прямых и обратных лучей.
- Программные модули сцены и ее компонентов.

Программная модель стохастической трассировки луча (прямого и обратного) полностью определяется программными моделями элементов сцены, определенными на уровне из базовых интерфейсов. Схематично алгоритм трассировки луча сводится к бесконечному циклу прямолинейных сегментов. Конечные и начальные параметры луча (координаты, направление, цвет, состояние поляризации) каждого сегмента обуславливаются граничными условиями сцены, такими как геометрия объектов сцены и оптические свойства геометрических поверхностей и сред. Стохастический выбор события, переводящий луч из одного сегмента в другой, осуществляется с помощью метода «русской рулетки», позволяющего математически корректно решать уравнение рендеринга (1), являющееся бесконечной суммой сегментов трассы луча.

3. БАЗОВЫЕ ИНТЕРФЕЙСЫ ОБЪЕКТОВ СЦЕНЫ

Основной задачей объектно-ориентированного проектирования является обеспечение взаимодействия объектов на уровне их базовых интерфейсов. Применительно к задаче рендеринга это значит, что объекты сцены и компоненты рендеринга взаимодействуют друг с другом только на уровне базовых интерфейсов. Например, при поиске пересечения траектории луча с геометрическим

объектом сцены, алгоритм не знает (и не должен знать), геометрический объект какого типа оказался на пути луча (элемент треугольной сетки, поверхность сплайна, аналитическая поверхность и т.п.). Алгоритму достаточно знать координаты точки пересечения луча с поверхностью и локальную информацию о поверхности в точке пересечения. Основные преимущества объектно-ориентированной организации это:

- Возможность изменения реализации объектов сцены без изменения алгоритма работы программы. Например, изменение способа хранения данных, определяющих двунаправленную функцию рассеивания, ограничивается данным объектом и не затрагивает компоненты рендеринга.
- Возможность реализации принципиально новых объектов сцены без изменения алгоритма работы программы. Например, для того, чтобы добавить новый тип аналитической геометрии сцены, достаточно лишь реализовать базовый интерфейс геометрического объекта для нового типа аналитической геометрии.

Реализация алгоритмов на уровне базовых интерфейсов приводит к тому, что рендеринг становится не зависимым от данных сцены. Как следствие упрощается алгоритм и программная реализация рендеринга, поскольку отпадает необходимость проверки типов объектов сцены и ветвлений алгоритма в зависимости от типа объекта сцены (упрощение алгоритма всегда приводит к большей надежности и устойчивости работы программы).

Специфика проектирования базовых интерфейсов объектов сцены и рендеринга заключается в том, физические принципы преобразования светового излучения на объектах сцены накладываются на модель стохастической двунаправленной трассировки луча, используемую в рендеринге. Поэтому программный интерфейс рендеринга и объектов сцены это своего рода парадигма, на которой строится процесс построения физически корректного изображения. То есть модель рендеринга и степень его физической корректности полностью определяется программными интерфейсами его компонентов.

Рассмотрим основные типы объектов сцены и их базовые интерфейсы:

- Геометрия сцены. Базовый метод геометрического объекта это поиск пересечения луча с поверхностью. Кроме факта пересечения данный метод должен возвращать локальную информацию о поверхности (дистанцию до поверхности, нормаль в точке пересечения, ссылку на оптические свойства поверхности и т.п.). Для ускорения трассировки лучей геометрия может использовать пространственную иерархию [7].
- Источники света. Базовые методы источника света должны информировать сцену о мощности излучения, генерировать случайный луч, испускаемый с заданной функцией распределения вероятностей, вычислять интенсивность и яркость излучения в заданном направлении и искать (аналогично геометрическому объекту) пересечение луча с объектом источника света.
- Приемник излучения. Базовые методы приемника излучения это прием и накопление лучей, идущих от источника света, испускание обратных лучей и вычисление

значений накопленной световой величины для заданных лучей.

- Оптические свойства поверхности. Базовые методы данного объекта должны обеспечивать вычисление первичной (вторичной) яркости поверхности и вычислять направление луча, рассеянного на поверхности с заданной функцией распределения вероятностей.
- Оптические свойства среды. Базовые методы данного объекта должны обеспечивать преломление / отражение луча на гладкой границе раздела двух сред, вычисление прямолинейной длины свободного пробега и цветового ослабления в среде, переносить луч на заданную дистанцию и вычислять направление луча, «рассеянного» на неоднородностях среды.

Хотя реальная программная система имеет более «широкий» программный интерфейс, приведенный базовый интерфейс является достаточным для построения физически корректных изображений.

4. СПЕЦИАЛЬНЫЕ ОПТИЧЕСКИЕ ЭЛЕМЕНТЫ И ИХ ПРОГРАММНЫЕ ИНТЕРФЕЙСЫ

В ряде случаев, объекты сцены являются физически более сложными элементами, например, дифракционными элементами, и не могут быть описаны в рамках текущего объектно-ориентированного программного интерфейса. В качестве решения данной проблемы предлагается использовать специализированные объекты, которые, с одной стороны, позволяют решать задачи физически аккуратного моделирования, а с другой стороны, могут быть интегрированы в существующую объектно-ориентированную систему построения изображения. Для решения этой проблемы вводится концепция оптических элементов.

Оптический элемент – это объект программно моделируемой оптической системы или устройства, который позволяет использовать оптически более сложные модели взаимодействия света с объектами сцены, чем модели, предлагаемые текущим объектно-ориентированным программным интерфейсом [6].

Оптический элемент расширяет функциональность систем компьютерной графики, но при этом не раскрывает собственную функциональность. Поэтому с точки зрения сцены оптический элемент – это «черный ящик», который принимает на вход лучи, преобразует их и затем выдает их назад программе построения изображения, трассирующей лучи в сцене. Схематично процесс взаимодействия лучей с оптическим элементом показан на рис. 2.

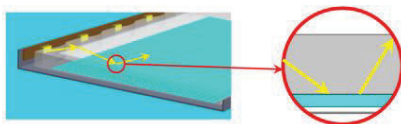


Рис 2: Распространение лучей в сцене, содержащей оптический элемент.

Поскольку оптический элемент является «черным ящиком», его содержимое скрыто от рендеринга, то есть изображение сцены не будет содержать изображения деталей оптического элемента. Такое поведение напоминает взгляд сквозь линзу. Мы не видим поверхности линзы, но видим изображение, которое она формирует.

По местоположению в сцене оптический элемент может быть независимым компонентом с собственной внешней границей или быть присоединенным к поверхности сцены. На рис. 3 представлены примеры микроструктурных рассеивающих оптических элементов, нанесенных на нижнюю поверхность светопроводящей пластины.

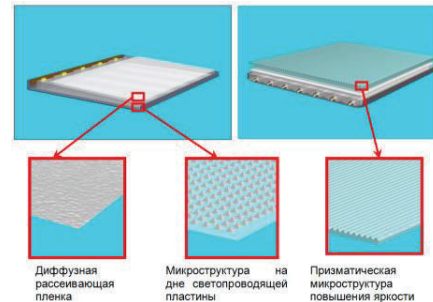


Рис 3: Микроструктурные рассеивающие оптические элементы.

Особенностью данных элементов является сложность рассеивающей микроструктуры, содержащей десятки (и сотни) миллионов независимых рассеивающих геометрических элементов сложной формы.

С точки зрения программной реализации оптический элемент это независимая специализированная сцена и ее базовый программный интерфейс должен покрывать основные объекты сцены:

- Геометрия сцены (поиск пересечения луча с оптическим элементом).
- Оптические свойства («расширенное» преобразование луча на оптическом элементе, когда точка входа луча в и выхода из оптического элемента могут не совпадать).
- Источник света (мощность излучения, генерация случайного луча и вычисление интенсивности и яркости излучения оптического элемента в заданном направлении).
- Приемник излучения (сохранение луча, падающего от источника света, испускание обратного луча и вычисление накопленных световых характеристик для заданного луча).

Необходимо отметить, что базовый интерфейс оптического элемента не предполагает методов расчета первичной и вторичной яркости на самом элементе, поскольку оптический элемент «прозрачен» для падающих на него лучей.

5. ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ РЕАЛИЗАЦИЯ МЕТОДА ДВУНАПРАВЛЕННОЙ ТРАССИРОВКИ ЛУЧЕЙ

Объектно-ориентированный алгоритм двунаправленной стохастической трассировки лучей был реализован в программном комплексе INSPIRER2 [5]. Отличительными чертами разработанной программной модели являются:

- Реализация моделей объектов сцены и алгоритмов трассировки лучей в спектральном пространстве.
- Реализация поляризационных вычислений в моделях объектов сцены и методах прямой трассировки луча.
- Включение моделей оптических элементов как специальных объектов сцены и их поддержка в методах прямой и обратной стохастической трассировки лучей.

Программная модель позволяет создавать фотореалистичные изображения геометрически и оптически сложных сцен. В качестве примера приводятся результаты моделирования изображения экрана жидкокристаллического дисплея в салоне автомобиля при различных условиях внешнего освещения. Наиболее сложным элементом модели является жидкокристаллический дисплей, схематично представленный на рис. 4. Свет от светодиодных источников распространяется в среде светопроводящей пластины, рассеивается на микроструктуре, заданной с помощью соответствующего оптического элемента, и выходит наружу через ряд рассеивающих и поляризационных фильтров, включая TFT матрицу, формирующую изображение.

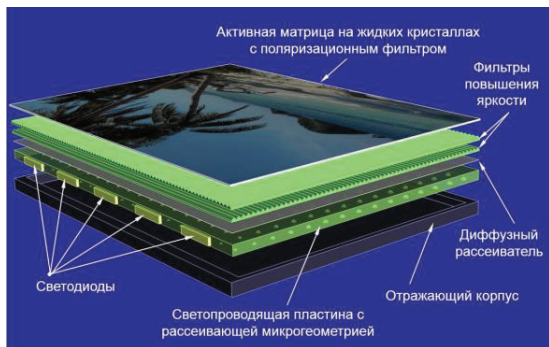


Рис 4: Модель жидкокристаллического дисплея.

Результаты моделирования при различных условиях внешнего освещения салона автомобиля представлены на рис. 5 (а-г).

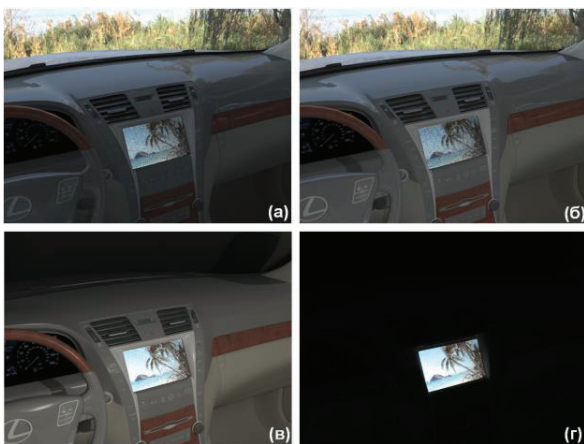


Рис 5: Результаты моделирования при различных условиях внешнего освещения.

На рис. 5(а) использовалось дневное освещение, заданное в HDR1 файле, на рис. 5(б) было добавлено салонное освещение, рис. 5(в) был создан для одного салонного освещения, а рис. 5(г) не имел никакого внешнего освещения.

Шум на изображении экрана дисплея объясняется спецификой работы алгоритма и частично связан с относительно малой мощностью (по отношению к внешнему освещению) светодиодных источников света жидкокристаллического дисплея.

6. ЗАКЛЮЧЕНИЕ

В рамках программного комплекса INSPIRER2 был реализован объектно-ориентированный алгоритм двунаправленной стохастической трассировки лучей. Данный алгоритм базируется на физически корректных моделях сцены и на уровне базовых интерфейсов позволяет осуществить физически корректный и эффективный рендеринг оптически и геометрически сложных сцен. Введенная концепция оптического элемента расширяет объектно-ориентированную модель рендеринга специальными объектами, которые не могут быть представлены в рамках базовой модели сцены.

Объектно-ориентированная модель рендеринга была реализована в спектральном пространстве. Поляризационные эффекты были поддержаны для прямой трассировки лучей, что обеспечивает физически корректный результат наблюдения в неполяризованном пространстве.

7. БЛАГОДАРНОСТИ

Работа поддержана грантами РФФИ № 12-01-00560 и 13-01-00454, а также компанией Integra Inc.

Авторы выражают свою признательность В.Г. Соколову и А.А. Гарбулю за предоставленные примеры.

8. ССЫЛКИ

- [1] Henrik Wann Jensen, Per Christensen. High quality rendering using ray tracing and photon mapping / SIGGRAPH '07.
- [2] Matt Pharr, Greg Humphreys. Physically Based Rendering - From Theory to Implementation // Morgan Kaufmann, 2004.
- [3] Toshiya Hachisuka and Henrik Wann Jensen. Stochastic progressive photon mapping. ACM Trans. Graph., 28(5):1-8, 2009.
- [4] Kajiya, J. T. The rendering equation / Siggraph 1986: 143.
- [5] <http://www.integra.jp/en/index.html>
- [6] Волобой А.Г., Галактионов В.А., Ершов С.В., Жданов Д.Д. Оптические элементы как средство расширения функциональности программ оптического моделирования // Труды 16-ой межд. конф., ГрафиКон' 2006, Россия, Новосибирск, июль 1-5, 2006, с. 182-191.
- [7] Ingo Wald, Carsten Benthin, and Philipp Slusallek. A Simple and Practical Method for Interactive Ray Tracing of Dynamic Scenes / report, Saarland University, 2002

Abstract

The article is devoted to design of the object-oriented model of physically accurate rendering of the geometrically and optically complex scenes. The article proposes classification of the scene objects and base program interfaces for each kind of scene object which is sufficient for physically accurate light simulations and rendering. Moreover the article proposes the hierarchical scene organization which allows on the base of stochastic ray tracing creating very complex models from viewpoint of geometrical and optical scene organization. As an example the article is illustrated with rendering results of LCD display of car dashboard under different conditions of external illumination.

Keywords: Scene, scene object, optical element, program interface, stochastic ray tracing, physically accurate rendering.