

УДК 004.35

ИССЛЕДОВАНИЕ ТЕХНИКИ СПЕКТРАЛЬНОЙ ТРАССИРОВКИ ПУТЕЙ

Перевалов С.С., Данилов Е.А.

Пензенский государственный технологический университет, Пенза, e-mail: EvgenyDplus@mail.ru

В данной статье был проведен анализ такой техники рендеринга, как трассировки путей. Отмечается важная роль компьютерной графики в современном программном обеспечении. Выделены семейства физических моделей для техники трассировки путей: рассеяния, отражения, пропускания. Подробно рассмотрены особенности основных моделей в каждом семействе. Отмечено что применение сферических координат требует больших вычислительных затрат из-за дополнительного преобразования в связи с чем их применение сведено к минимуму. Все расчеты выполняются с использованием спектральной плотности излучения для материалов объекта и источников света, что позволяет лучше декодировать цвета. Разработано тестовое программное обеспечение, реализующие трассировку путей, на языке программирования C++. Программное обеспечение реализует несколько физических моделей BRDF и моделей затенения. Приводится архитектура разработанного программного обеспечения. Обосновывается модульное построение программного обеспечения. Основной класс SCLT содержит цикл общего алгоритма трассировки лучей. Экземпляры класса Tracer моделируют пути. Класс BRDF реализует основанные физически модели. Для моделей затенения есть базовый класс ShadingModel. Подробно описаны используемые классы их назначение и реализуемые методы с использованием языка UML.

Ключевые слова: трассировка путей, рендеринг, двулучевая функция отражательной способности, спектральная плотность излучения, физически-корректный рендеринг, ДФОС, PBR, SPD, язык C++

RESEARCH THE TECHNIQUE PATHS TRACING

Perevalov S.S., Danilov E.A.

Penza State Technological University, Penza, e-mail: EvgenyDplus@mail.ru

This article analyzes a rendering technique called «path tracing». Importance of computer graphics in modern software is discussed. Path tracing physical model families for scattering, reflection and transmittance are distinguished. Each family primary model features are reviewed in detail. It is noted that spherical coordinates' application requires significant computing power because of an additional transformation, so their usage is brought down to a minimum. All calculations are done with usage of radiance spectral density, which allows for improved color decoding. Proof-of-concept path tracing software is developed using C++ programming language. This software implements several physical BRDF and shading models. Software design is presented. Modular application design is rationalized. Main class SCLT includes general path tracing algorithm working cycle. Tracer class instances simulate ray paths. BRDF class implements physically based surface models. For shading models abstract class ShadingModel is implemented. Used classes, their purpose and implemented methods are depicted in detail using UML diagram language.

Keywords: path tracing, rendering, bidirectional reflectance distribution function, physically based rendering, spectral power distribution, BRDF, PBR, SPD, C++ language

Без компьютерной графики не обходится ни одна современная программа. Конструкторы, разрабатывая новые модели автомобилей и самолетов, используют трехмерные графические объекты. Важную роль в достижении реалистичности отображаемой сцены играет освещение. Техника трассировки путей моделирует освещение как можно более реалистично. Теоретическая база для предложенной техники приводится в [1], имеется описание и список применений. Статья [2] описывает попытку применения техники в условиях реального проекта. Также в исследовании использовались материалы [3, 4, 5], посвященные тонкостям реализации техники трассировки путей на практике.

Одной из основных функций рендерера является набор поддерживаемых им физических моделей. Все эти модели являются специализацией базового класса BRDF. Каждая модель, определенная в рендерере,

принадлежит к одному из следующих семейств:

- рассеяния;
- отражения;
- пропускания.

Первая реализованная модель диффузного семейства также является самой простой: модель Ламберта. Для моделей Орена-Найра и Торранса-Кука необходимо использовать другой подход. Их уравнения в значительной степени основаны на использовании направлений света в сферических координатах. Для их вычисления обычно требуется изменение системы координат, преобразование этих направлений в касательное пространство с началом в точке пересечения лучей. Таким образом, расчет сферических координат может быть согласованным для каждой точки пересечения и для каждого направления.

Но когда это возможно, лучше полагаться на тригонометрию и векторные опера-

ции, чтобы избежать затрат на вычисление преобразования. Вот почему все эти модели реализованы с использованием операций, которые не используют сферические координаты напрямую.

Для измеренной модели BRDF необходим другой подход. Наборы данных индексируются с использованием сферических координат. Вот почему, прежде всего, необходима генерация касательной пространственной оси. Для оси u будет использоваться нормаль в точке пересечения поверхности. Для касательных и битангентных векторов доступны разные подходы. Обычно система координат может быть создана строго из одного вектора, вырезая один из компонентов и инвертируя оставшиеся два. Это было бы хорошим методом для использования, потому что измеренный BRDF, как и все другие диффузные модели, представленные в этом тезисе, являются изотропными, поэтому им не нужно, чтобы тангенс был ориентирован в определенном направлении. Этот полезный метод используется в другой части рендерера, но для измеренной BRDF был исследован другой способ: возможно вычислить перекрестное произведение между нормалью и направлением для получения кандидата касательного вектора, а затем вычислить перекрестное произведение между ним и нормалью, чтобы получить битангенс. Это предпочтительный метод, используемый рендерером для вычисления тангенциального пространства для измеренной модели BRDF и выполнения изменения преобразования координат для направления, чтобы преобразовать их в сферические координаты. Теперь необходимо сделать еще один шаг: прямое считывание выборки из наборов данных, ближайших к сферической координате, полученной из предыдущего расчета, не подходит для визуализации. По этой причине специфические переназначить в сферической системе координат необходимо использовать. Повторное отображение задается следующей формулой (учитывая, что в рендерере y – это ось вверх):

$$remap(\theta_i, \phi_i, \theta_o, \phi_o) = (\sin \theta_i \sin \theta_o, \cos \theta_i \cos \theta_o, \Delta\phi).$$

Этот переназначение имеет два важных свойства:

– изотропность BRDF учитывается. Это дается $\Delta\phi$. Используя этот термин, пары направлений, которые имеют различные значения, могут быть связаны с тем же набором образцов, потому что они находятся на одинаковом расстоянии $\Delta\phi$ даже если значе-

ния ϕ не совпадают. Таким образом, BRDF не зависит от направления касательной;

– взаимность BRDF учитывается.

Последние модели, реализованные в рендерере – это модели отражения и передачи. Они используются вместе, чтобы создать стеклянный материал.

Ключевая особенность трассировщика пути рендеринга заключается в данных, используемых для оценки цвета: все расчеты выполняются с использованием спектральных распределений мощности для материалов объекта и источников света. Значение, полученное для каждого пикселя, затем преобразуется в трехцветные значения, а затем отображается цвет sRGB. Для регулирования общей мощности света в сцене был использован параметр яркости. Также sRGB Гамма-коррекция применяется, как и для другой модели, содержащейся в рендерере, который использует спектральные данные, чтобы лучше декодировать цвета, полученные из спектральных данных.

Рендерер поддерживает несколько типов рендеринга, основанных на различных видах цветовых данных, стандартных RGB или спектральных. Таким образом, существует требование, чтобы средство визуализации имело возможность изменять технику, используемую для визуализации сцены, удобным способом. Это также причина, по которой рендерер должен придерживаться принципа разделения задач, который требует разбиения программного обеспечения на модули, каждый из которых учитывает определенный набор информации и операций. Рендерер состоит из разных модулей, каждый из которых создан для управления определенной частью механизма рендеринга трассировки лучей.

Основным классом, который запускает процесс рендеринга, является *SCLT*. Он содержит основной цикл алгоритма трассировки лучей. Этот цикл проходит через каждый пиксель плоскости обзора и отслеживает текущий луч камеры с помощью экземпляра класса *Tracer* (рис. 1), чтобы получить цвет пикселя (или части пикселя,

в случае рендеринга с сглаживанием) изображения с помощью метода *getColor(Ray ray, int bounce)*, который имеет два параметра: первый – это трассируемый луч, реализованный в классе *Ray* и описываемый началом и направлением, второй – максимальное количество отскоков, которое разрешено лучу выполнить. Эти отскоки могли

быть получены по разным причинам: происходит переход луча между отражающим/преломляющим материалом в модели трассировки лучей Уиттеда или общее число максимально допустимых отказов при трассировке путей. Очевидно, в классе *SCLT* также есть общие настройки, необходимые для визуализации сцены: определение плоскости вида (*ViewPlane* класс) и определенные камеры (класс *Camera*).

Физически основанные модели реализованы как специализация базового класса BRDF (рис. 2). Что касается компонента трассировки и шейдинга, добавление новой модели BRDF требует только создания нового класса.

По умолчанию рендерер поддерживает следующие физически основанные модели BRDF:

- ламбертианский;

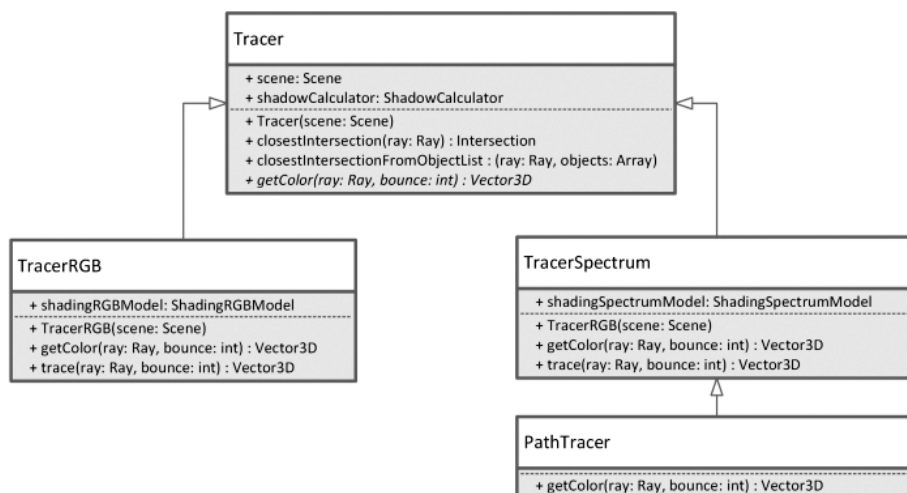


Рис. 1. Иерархия классов *Tracer*

Различные виды трассировщика реализуются в визуализаторе, каждый из учета для различных особенностей. Существует базовый класс *Tracer*, который реализует метод, общий для всей специализации трассировщиков: *closestIntersection(Ray ray)*. Этот метод используется для расчета ближайшего пересечения луча с объектами в сцене. Будет отображаться ближайший объект. Это стандартный метод, общий для всей модели трассировки лучей, реализованной в рендере. Поэтому он содержится в этом базовом классе.

Существует две специализации класса *Tracer*, каждая из которых учитывает различные методы цветопередачи: *TracerRGB* и *TracerSpectrum*. Как следует из названия, первый используется для рендеринга сцены с использованием стандартных данных RGB, второй – с использованием спектральных данных. Подклассы *Tracer* реализуют два метода: описанный ранее *GetColor(Ray ray, Int bounce)* и метод *trace(Ray ray, int bounce)*. Последний – это метод, который выполняет трассировку луча.

- Орена-Найяра;
- зеркальное отражение;
- зеркальная передача;
- Торранса-Кука.

Поскольку основной задачей визуализатора является отрисовка спектральной сцены, все модели BRDF реализованы с поддержкой только спектральных данных.

Различные BRDF связаны в *MaterialBRDF* класс для создания разных видов материала. Там также *MaterialRGB* класс, используемый в сцене RGB. Все они являются специализацией класса *Material*.

Предыдущие компоненты используют данные, взятые из класса *Scene*, который содержит определение сцены с информацией о трассировщике и модели затенения, которые будут использоваться во время рендеринга. *TracerModelFactory* и *ShadingModelFactory* создают экземпляры модели трассировки и затенения, определенной в описании сцены. В *SCLT* есть один экземпляр класса *Scene*, основанный на выборе пользователя, который передается как указатель через все ранее просмотренные классы, которые в этом нуждаются.

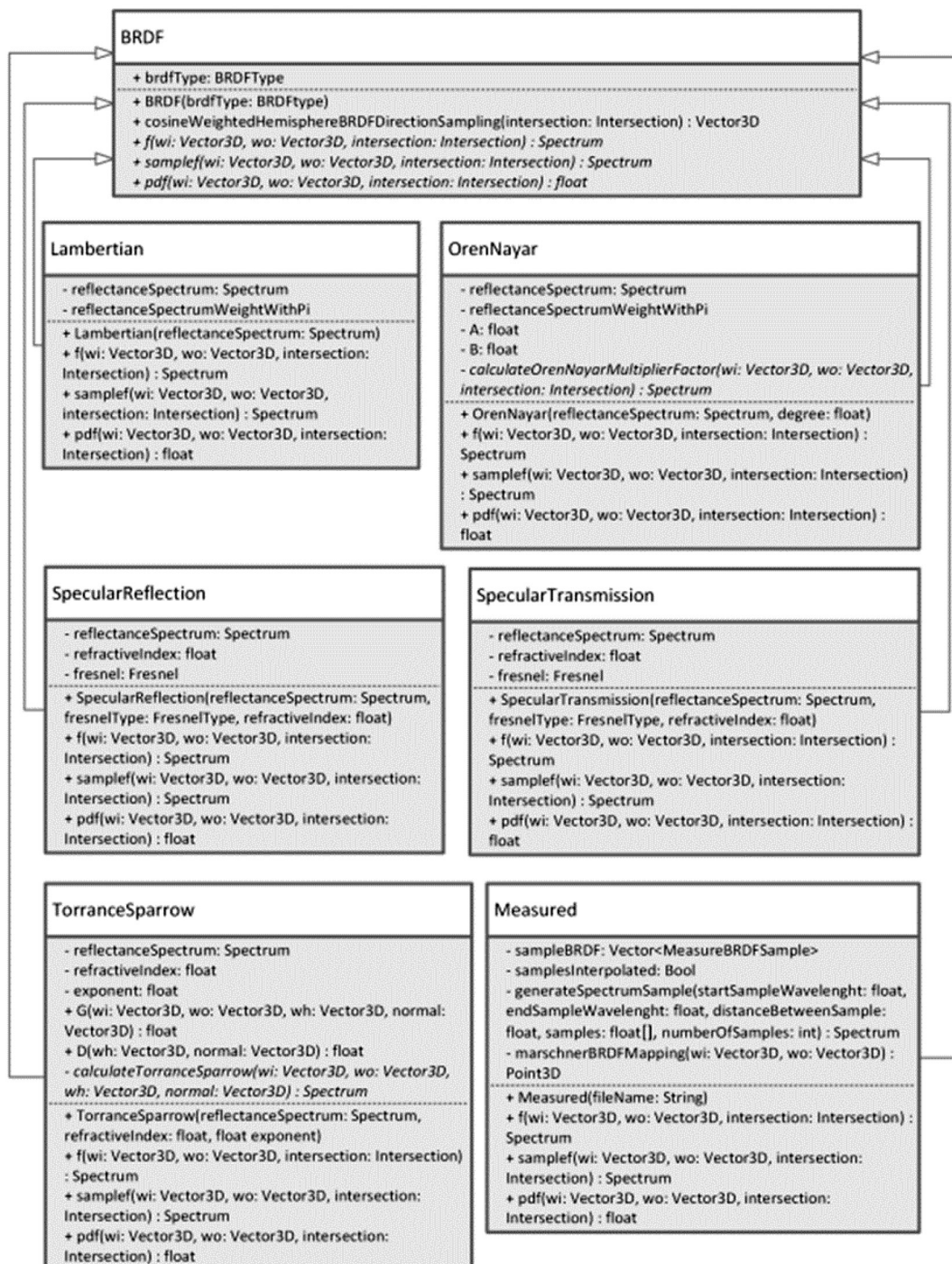


Рис. 2. Иерархия классов BRDF

Вся представленная структура была определена с акцентом на то, чтобы визуализатор был «подключаемой и расширяемой системой»: было бы легко добавить новые трассировщики или модели шейдинга в качестве специализации базовых классов, представленных ранее.

Для моделей затенения есть базовая модель `ShadingModel`. Этот класс расширен двумя специализациями, `ShadingRGBModel` и `ShadingSpectrumModel`. Все модели затенения, реализованные в рендерере, должны расширять один из этих классов, чтобы объявить о своей поддержке RGB или спектральных данных.

Рендерер поддерживает следующие модели затенения, каждый из которых реализуется своим классом. `Whitted` – класс, который реализует трассировку лучей Уиттеда с использованием эмпирических моделей BRDF Фонга/Блинна-Фонга и данных RGB.

`WhittedBRDF` – класс, который реализует трассировку лучей Уиттеда, используя физически корректные BRDF и спектральные данные. `PathBRDF` – класс, который реализует модель трассировки пути, используя, как и в случае с `WhittedBRDF`.

Список литературы

1. Jones M.W., Sramek M. 3D distance fields: A survey of techniques and applications // *Visualization and Computer Graphics*, 2006. – IEEE Transactions on 12 (4). – PP. 581-599.
2. Donnelly W. Per-Pixel Displacement Mapping with Distance Functions / *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*, 2005. – PP. 123-137.
3. Frisken S.F. Perry R.N. Designing with Distance Fields / *TR2006-054*, 2006. – PP. 60-66.
4. Jamriska O. Interactive Ray Tracing of Distance Fields // *Central European Seminar on Computer Graphics (CESCG)*, 2010. – PP. 27-34.
5. Koschier D., Deul C., Bender J. Hierarchical hp-Adaptive Signed Distance Fields // *Symposium on Computer Animation*, 2016. – PP. 189-198.