

# Система интерактивного расчета глобального освещения для гибридных сцен

Денис Боголепов<sup>1</sup>, Дмитрий Сопин<sup>1</sup>, Данила Ульянов<sup>2</sup>, Вадим Турлапов<sup>1</sup>

<sup>1</sup>Нижегородский государственный университет им. Н.И. Лобачевского

<sup>2</sup>Нижегородский государственный технический университет им. П.Е. Алексеева

denisbogol@gmail.com, sopindm@gmail.com, danila-ulyanov@ya.ru

## Аннотация

Разработана система интерактивной визуализации для сцен гибридной природы с поддержкой глобального освещения и динамической геометрии. Под гибридностью понимается сочетание полигональных моделей с объектами других типов, таких как поверхности второго порядка, неявно заданные функции, фрактальные множества и объемные данные. В основе системы – конвейер трассировки лучей, который может быть сконфигурирован для задач с различным балансом скорости и качества визуализации. Для ускорения вычислений используется SAH BVH дерево, которое строится на ГПУ в реальном времени. Прототип системы разработан на базе NVIDIA CUDA и обеспечивает высокую производительность визуализации гибридных сцен.

**Ключевые слова:** глобальное освещение, двунаправленная трассировка путей, GPU, GPGPU, CUDA.

## 1. ВВЕДЕНИЕ

Трассировка лучей – метод построения изображений компьютерных сцен путем отслеживания взаимодействий лучей света с объектами сцены. Классическая трассировка лучей Уиттеда является простейшим алгоритмом глобального освещения, поскольку позволяет моделировать эффекты затенения, идеального отражения и преломления. Однако реалистичная визуализация невозможна без расчета вторичного освещения, типичными примерами которого являются каустики и перенос цвета между соседними поверхностями. Распределенная трассировка лучей Кука добавила в процесс расчета генерацию случайных направлений, что позволило учесть некоторые эффекты вторичного освещения, включая ограниченный перенос цвета. Данный метод был обобщен в стохастической трассировке путей Кайя, которая позволила учесть все пути света в сцене. Однако сходимость изображения достигается медленно и требует большого объема вычислений. Дальнейшим развитием метода является двунаправленная трассировка путей, основная идея которой состоит в единовременном испускании лучей из источников света и объектива виртуальной камеры. Данный подход позволяет значительно повысить скорость сходимости при расчете вторичного освещения.

Методы на основе трассировки лучей традиционно использовались в компьютерной графике для синтеза изображений высокого качества. Параллельная природа задачи и непрерывный рост вычислительных возможностей со временем вывели данные методы за рамки медленной предварительной визуализации. Первые системы интерактивной трассировки лучей требовали либо высокопроизводительных суперкомпьютеров [1]–[3], либо были ограничены небольшими сценами [4] и простыми эффектами освещения [5]. Сегодня существуют интерактивные системы визуализации, которые на обычном потребительском оборудовании

позволяют решать задачи, практически недоступные традиционному алгоритму растеризации [6]–[9].

Для повышения производительности трассировки лучей используются ускоряющие структуры. Долгое время основным показателем их эффективности служила скорость визуализации, при этом трудоемкость построения структур практически игнорировалась. В результате первые интерактивные системы трассировки позволяли обрабатывать сцены только со статической геометрией. Подобная ситуация привела к пересмотру ускоряющих структур, направленному на разработку алгоритмов быстрого построения или обновления при сохранении высокой скорости визуализации. Исследования показали, что поддержка динамических сцен возможна для всех основных структур, включая регулярные и иерархические сетки,  $k$ - $d$  деревья и иерархии ограничивающих объемов [10].

Наряду с динамическими сценами значительное внимание получило интерактивное глобальное освещение. Исследователи предложили различные техники, которые поддерживают полное или частичное моделирование вторичного освещения на графической аппаратуре. В работе [11] представлен оптимизированный метод внешней преграды (*ambient occlusion*), который обеспечивает качественное моделирование затенения, но не учитывает многократных отражений света. В работе [12] описан метод отражающих теневых карт (*reflective shadow maps*), который позволяет обрабатывать только один отскок вторичного освещения. В работе [13] представлен метод мгновенной излучательности (*instant radiosity*), который сводит расчет вторичного освещения к прямому освещению от виртуальных точечных источников света. В работах [14]–[15] авторы реализовали варианты метода фотонных карт, который является универсальным решением для моделирования большинства эффектов глобального освещения.

Целью настоящей работы является разработка системы для интерактивного расчета глобального освещения на ГПУ без допущений (*unbiased*). В основе системы лежит ускоряющая структура на базе SAH BVH дерева, для которой в работе [16] были представлены эффективные алгоритмы построения и обхода. Программный конвейер визуализации может быть настроен на выполнение таких методов, как испускание лучей, трассировка лучей, стохастическая трассировка путей и двунаправленная трассировка путей. Указанные методы обладают различными возможностями и трудоемкостью, обеспечивая компромисс между скоростью и качеством визуализации в конкретной задаче. Вне зависимости от метода визуализация сохраняет корректность, основу которой составляют физически обоснованные модели материалов и источников света.

## 2. ОСНОВНЫЕ МОДЕЛИ И МЕТОДЫ

В данном разделе излагаются основные модели и методы, которые лежат в основе разработанной системы и показывают полноту моделирования глобального освещения.

## 2.1 Взаимодействие света с поверхностями

В общем случае световая энергия падает на поверхность в некоторую точку  $P$  с направления  $\Psi$  и покидает в другой точке  $Q$  по направлению  $\Theta$ . Для описания этой зависимости используется функция объемного рассеяния (*BSSRDF*). Если световая энергия покидает поверхность из той же точки, то отражательные свойства могут быть описаны более простой двунаправленной функцией отражения (*BRDF*). Данная функция определяется в точке  $x$  как отношение яркости, отраженной по направлению  $\Theta$ , к облученности поверхности, падающей с направления  $\Psi$ :

$$f_r(x, \Psi \rightarrow \Theta) = \frac{dL(x \rightarrow \Theta)}{dE(x \leftarrow \Psi)} = \frac{dL(x \rightarrow \Theta)}{L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi} \quad (1)$$

Наряду с двунаправленной функцией отражения для обратной стороны поверхности вводится двунаправленная функция пропускания (*BTDF*). Для совместного описания отражающих и пропускающих свойств часто используется двунаправленная функция рассеяния (*BSDF*).

## 2.2 Уравнение визуализации

Для каждой точки  $x$  поверхности и каждого направления  $\Theta$  уравнение визуализации определяет исходящую яркость  $L(x \rightarrow \Theta)$  как сумму излучаемой яркости  $L_e(x \rightarrow \Theta)$  и отражаемой яркости  $L_r(x \rightarrow \Theta)$ :

$$L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + L_r(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{H_x} f_r(x, \Psi \rightarrow \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi \quad (2)$$

Такая форма уравнения называется полусферической, поскольку содержит интеграл по полусфере возможных направлений. Альтернативной формой является площадная, которая заменяет интегрирование по полусфере на интегрирование по поверхностям, видимым из точки  $x$ :

$$L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + L_r(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_A f_r(x, \bar{x}\bar{y} \rightarrow \Theta) L(y \rightarrow \bar{y}\bar{x}) V(x, y) G(x, y) dA, \quad (3)$$

Через  $G(x, y)$  обозначен геометрический член, описывающий взаимное расположение поверхностей в точках  $x$  и  $y$ :

$$G(x, y) = \frac{\cos(N_x, \bar{x}\bar{y}) \cos(N_y, \bar{y}\bar{x})}{r_{xy}^2} \quad (4)$$

Через  $V(x, y)$  обозначена функция видимости, которая равна единице, если точки  $x$  и  $y$  взаимно видимы, и нулю в противном случае. Данная функция вычисляется с помощью операции испускания луча  $r(x, \Psi)$ , определяющей ближайшую точку соударения заданного луча с объектом сцены:

$$r(x, \Psi) = \{y \mid y = x + t\Psi\} \quad (5)$$

$$\tau = \min_{t>0} \{t \mid x + t\Psi \in A\}$$

Очевидно, точки  $x$  и  $y$  взаимно видимы тогда и только тогда, когда существует такое направление  $\Psi$ , что  $r(x, \Psi) = y$ .

## 2.3 Стохастическая трассировка путей

Через каждый пиксель экранной плоскости алгоритм трассировки путей испускает первичный луч, для которого определяется ближайшая точка соударения  $x$  со сценой. Яркость сгенерированного луча вычисляется из уравнения визуализации методом Монте-Карло. Для повышения эффективности вычислений отраженная яркость разделяется на две составляющие:

$$L_r(x \rightarrow \Theta) = L_{direct}(x \rightarrow \Theta) + L_{indirect}(x \rightarrow \Theta) = \int_{A_{sources}} L_e(y \rightarrow \bar{x}\bar{y}) f_r(x, \bar{x}\bar{y} \rightarrow \Theta) G(x, y) V(x, y) dA_y + \int_{H_x} L_r(r(x, \Psi) \rightarrow -\Psi) f_r(x, \Psi \rightarrow \Theta) \cos(N_x, \Psi) d\omega_\Psi \quad (6)$$

Прямое освещение  $L_{direct}(x \rightarrow \Theta)$  поступает в точку  $x$  непосредственно от источников света. Вторичное освещение  $L_{indirect}(x \rightarrow \Theta)$  падает в точку  $x$  после, по крайней мере, одного отражения от другой поверхности сцены.

### 2.3.1 Расчет прямого освещения

Для генерации теневых лучей используется следующий двухшаговый алгоритм:

- С помощью дискретного распределения  $p_L(k)$  выбирается источник света  $k$ .
- С помощью условной плотности вероятности  $p(y|k)$  на поверхности источника  $k$  выбирается точка  $y$ .

Составная плотность вероятности для выбора точки  $y$  на объединенной поверхности источников равна  $p_L(k)p(y|k)$ , что ведет к следующей оценке яркости теневого луча:

$$\langle L_{direct}(x \rightarrow \Theta) \rangle = \frac{L_e(y \rightarrow \bar{x}\bar{y}) f_r(x, \bar{x}\bar{y} \rightarrow \Theta) G(x, y) V(x, y)}{p_L(k) p(y|k)} \quad (7)$$

В текущей версии системы для выбора источника света используется дискретное равномерное распределение.

### 2.3.2 Расчет вторичного освещения

Для заданной плотности вероятности  $p(\Psi)$  генерируется случайное направление  $\Psi$ , при этом оценка компоненты вторичного освещения имеет вид:

$$\langle L_{indirect}(x \rightarrow \Theta) \rangle = \frac{L_r(r(x, \Psi) \rightarrow -\Psi) f_r(x, \Psi \rightarrow \Theta) \cos(N_x, \Psi)}{p(\Psi)} \quad (8)$$

В данном выражении отраженная яркость  $L_r$  остается неизвестной величиной, что приводит к рекурсивной процедуре вычисления вторичного освещения. Для остановки рекурсии устанавливается максимальное число итераций или применяется принцип русской рулетки. В настоящей работе для уменьшения дисперсии оценок используется выборка по значимости: плотность вероятности  $p(\Psi)$  строится пропорционально BSDF поверхности.

## 2.4 Двунаправленная трассировка путей

Для повышения скорости сходимости используется двунаправленная трассировка путей, которая испускает лучи одновременно из источников света и виртуальной камеры.

- Световой путь описывается последовательностью  $x_0, x_1, x_2, \dots, x_{N_s}$ , где через  $x_{i+1}$  обозначена точка, видимая из точки  $x_i$  по направлению  $\Theta_{x_i}$ .
- Видовой путь описывается последовательностью  $y_0, y_1, y_2, \dots, y_{N_e-1}$ , где через  $y_{j+1}$  обозначена точка, из которой видна точка  $y_j$  по направлению  $\Theta_{y_{j+1}}$ .

Все узлы сгенерированных путей соединяются теневыми лучами, и соответствующие вклады добавляются к яркости пикселя, что приводит к следующей оценке:

$$\langle L_r(x \rightarrow \Theta) \rangle = \sum_{i=0}^{N_s} \sum_{j=0}^{N_e-1} w_{ij} C_{ij} \quad (9)$$

Здесь через  $C_{ij}$  обозначен вклад теневого луча, который соединяет  $i$ -ый узел светового пути с  $j$ -ым узлом видовой пути. Через  $w_{ij}$  обозначены нормировочные коэффициенты теневых лучей, которые необходимо учитывать для корректной сходимости метода.

## 3. АРХИТЕКТУРА СИСТЕМЫ

Для расчета изображения разработан программный конвейер трассировки лучей, все этапы которого выполняются на ГПУ и могут быть разделены на следующие две стадии.

Light Stage

1. На основе информации об источниках света для каждого пикселя изображения генерируется световой луч.
2. Для каждого сгенерированного луча вычисляется ближайшая точка соударения с видимым объектом сцены.
3. На основе свойств поверхностей в точках соударения генерируются вторичные лучи (или принимается решение о прекращении светового пути).
4. Выполняется переход к шагу 2.

View Stage

1. На основе информации о камере для каждого пикселя изображения генерируется первичный луч.
2. Для каждого сгенерированного луча вычисляется ближайшая точка соударения с видимым объектом сцены.
3. Каждая точка соударения соединяется теневыми лучами со всеми префиксами светового пути.
4. Каждый теневой луч тестируется на соударение с любым объектом между световым и видовым путем. Если пересечение найдено, то теневой луч удаляется.
5. На основе свойств поверхностей в точках соударения вычисляется вклад каждого теневого луча в яркость соответствующего пикселя.
6. На основе свойств поверхностей в точках соударения генерируются вторичные лучи (или принимается решение о прекращении видового пути).
7. Выполняется переход к шагу 2.

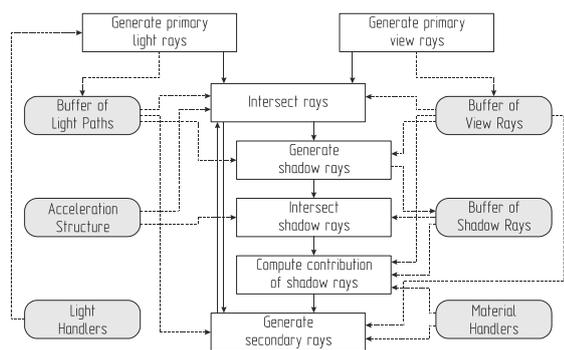


Рис. 1. Схема работы конвейера трассировки лучей.

Данный конвейер наряду с двунаправленной трассировкой путей позволяет реализовать другие методы визуализации. Полное отключение генерации световых лучей соответствует стохастической трассировке путей. Если при этом генерация вторичных лучей выполняется только для расчета идеального отражения или преломления, то конвейер реализует классический алгоритм трассировки лучей. Полное отключение вторичного освещения соответствует простому алгоритму испускания лучей. В результате возможна адаптация к широкому кругу задач – от полного моделирования глобального освещения до визуализации анимированных сцен в реальном времени с упрощенным освещением.

**3.1 Адаптация конвейера к архитектуре ГПУ**

Для обработки данных на графическом процессоре вся информация должна быть представлена в виде набора массивов (буферов). Однако во многих системах моделирования и форматах хранения используется более удобное описание компьютерной модели в виде графа сцены. В настоящей работе разработан универсальный механизм представления модели, который на пользовательском уровне функционирует как граф сцены, а на нижнем уровне обеспечивает эффективную сериализацию данных в набор массивов. Указанные массивы могут использоваться как для загрузки на

графический процессор, так и для передачи на другие вычислительные узлы при расчете на кластерной системе.

Другой особенностью настоящей работы является расширяемая подсистема материалов и источников света. Реализация указанных возможностей основана на концепции обработчиков, которые имеют уникальную реализацию для различных источников и материалов и представлены в виде указателей на функции с универсальной сигнатурой вызова (аппаратная поддержка доступна в последнем поколении графических ускорителей NVIDIA Fermi).

**3.1.1 Сериализация данных**

На верхнем уровне сцена представлена в виде дерева, узлы которого служат только для организации объектов сцены в иерархию и не содержат конкретной информации. Для хранения специфических данных (включая сетки треугольников) используются ядра, которые могут быть включены в один или несколько узлов.

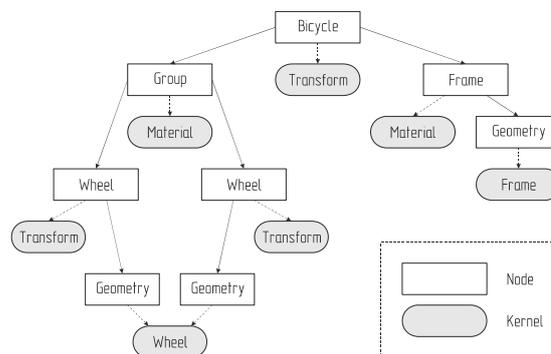


Рис. 2. Пример графа сцены для модели велосипеда.

Одной из ключевых особенностей конвейера является эффективная сериализация данных в массивы для передачи на ГПУ. Данный механизм основан на внутреннем представлении сцены в виде набора хранилищ данных различного типа, общая функциональность которых описывается следующим базовым интерфейсом:

- *Add node*. Добавляет информацию об узле графа сцены в хранилище. Память для данных ядра при этом не выделяется.
- *Delete node*. Удаляет информацию об узле из хранилища. Если для данных ядра в хранилище была выделена память, то она освобождается.
- *Allocate*. Выделяет в хранилище память для данных ядра. Запись данных ядра в хранилище не выполняется.
- *Write*. Записывает данные из ядра узла в хранилище. Память должна быть выделена заранее. При записи учитываются значения атрибутов узлов, примерами которых служат трансформации и материалы.
- *Free*. Освобождает память, которая была выделена в хранилище для данных ядра. Информация об узле из хранилища не удаляется.

Идентификация объектов в хранилище осуществляется по узлам, поэтому каждый узел не может включаться в граф сцены более одного раза. Однако включение одного ядра несколько раз возможно, поскольку оно может соответствовать нескольким узлам. При этом для каждого узла место в хранилище будет выделено отдельно, что необходимо ввиду специфики обработки данных на ГПУ (структуры с указателями обрабатываются неэффективно). Раздельное хранение информации об узлах необходимо также из-за возможных различий в свойствах узлов. Например, одина-

ковым ядрам геометрии с различными атрибутами трансформации соответствует различная хранимая информация.

### 3.1.2 Первичные лучи и сглаживание

В процессе расчета изображения каждому пикселю приписывается яркость  $L_{pixel}$ , которая выражается интегралом по экранной плоскости:

$$L_{pixel} = \int_{Screen} L(p \rightarrow eye)h(p)dp = \int_{Screen} L(x \rightarrow \bar{x}p)h(p)dp \quad (10)$$

Здесь через  $x$  обозначена точка, видимая из глаза  $eye$  через точку  $p$  экранной плоскости, а через  $h(p)$  – весовая функция (в текущей версии используется прямоугольный фильтр для усреднения падающей яркости по площади пикселя). Для оценки данного интеграла применяется метод Монте-Карло. На каждой итерации первичные лучи направляются на случайные точки в пределах соответствующих пикселей, при этом результирующее изображение получается путем усреднения нескольких проходов визуализации. Данный метод не влечет дополнительных расходов в трассировке путей и обеспечивает высокое качество сглаживания.

### 3.1.3 Поиск пересечения с объектами сцены

Наиболее трудоемким этапом визуализации является поиск ближайшего пересечения луча с объектами сцены. Для повышения производительности применяются различные ускоряющие структуры, которые хранят информацию о расположении объектов сцены и позволяют избежать полного перебора. В настоящей работе для этой цели используется иерархия ограничивающих объемов (BVH), для которой в работе [16] предложен эффективный алгоритм построения на основе эвристики площадей поверхностей (SAH), позволяющий в реальном времени визуализировать сложные динамические сцены.

Отличительной особенностью настоящей работы от большинства аналогичных проектов является возможность прямой визуализации ряда важных для виртуальной реальности объектов без предварительной триангуляции. Примерами таких объектов служат поверхности второго порядка, поверхности Безье, изоуровни скалярных полей, пространственные фрактальные множества, фрагменты сплошных сред. Важной областью применения сплошных сред является прямая визуализация объема, которая может быть легко интегрирована в полигональную модель (вода в прозрачном аквариуме или клубы дыма в помещении). Необходимым условием обработки таких объектов является наличие алгоритма пересечения с лучом. Для реализации указанных возможностей процедура поиска пересечения разбивается на несколько процедур, каждая из которых работает с определенным типом геометрических объектов.

### 3.1.4 Источники света и прямое освещение

В рамках настоящей работы разработана расширяемая подсистема источников света. На данный момент реализована поддержка изотропных точечных источников, направленных источников, непрерывных диффузных источников и бесконечно удаленных непрерывных источников, таких как верхний свет и карты окружения. Система поддерживает загрузку карт окружения из различных HDRi форматов и обеспечивает их эффективную обработку на основе выборки по значимости. Для задания характеристик всех источников используются физические величины – энергетическая яркость или сила света.

При расчете прямого освещения все источники света могут рассматриваться как один логический источник или обрабатываться независимо друг от друга. В первом режиме вне зависимости от числа источников в каждой точке соударе-

ния генерируется только один теневой луч. Во втором режиме теневые лучи генерируются для каждого источника света, а результаты расчетов суммируются.

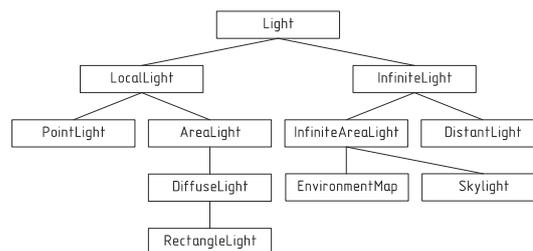


Рис. 3. Диаграмма классов для описания источников.

Первый режим обеспечивает высокую скорость работы, но при этом страдает скорость сходимости изображения. Для получения качественного результата во втором режиме требуется меньшее число итераций, но каждый проход визуализации более затратный. Тем не менее, при расчете финального изображения второй режим отработает быстрее и является предпочтительным.

### 3.1.5 Материалы и вторичное освещение

Для описания свойств поверхностей разработана расширяемая подсистема материалов, в основе которой лежит разделение механизмов рассеяния света на три типа: диффузное (*diffuse*), размытое зеркальное (*glossy*) и зеркальное (*specular*). Для описания конкретных типов рассеяния используются двунаправленные функции отражения (BRDF) и пропускания (BTDF), которые в большинстве случаев удобно обозначать общим термином  $BxDF$ . Текущая версия системы поддерживает модели Ламберта и Орена-Найара для диффузного отражения, модели Уорда и Блинна для размытого зеркального отражения, а также модели идеального зеркального отражения и преломления.

Большинство реальных материалов описываются комбинацией нескольких типов отражения или пропускания. Таким образом, рассеивающие свойства поверхности можно представить взвешенной суммой  $BxDF(f_c)$  различного типа, которая выражает BSDF ( $f_s$ ) поверхности:

$$f_s(x, \Psi \rightarrow \Theta) = \sum_{i=1}^N \alpha^i f_x^i(x, \Psi \rightarrow \Theta) \quad (11)$$

$$\sum_{i=1}^N \alpha^i \leq 1, \quad c \in \{R, G, B\} \quad (12)$$

Весовые коэффициенты  $\alpha^i$  в общем случае являются векторными и определяют свойства поверхности отдельно для каждого цветового канала. Функция BSDF является физически корректной, если справедливо неравенство (12). Данная функция используется в системе для описания материалов и передается для обработки на ГПУ.

Для каждой  $BxDF$  компоненты система позволяет задавать текстурную карту, которая корректирует весовой коэффициент в каждой точке  $x$  поверхности:

$$\tilde{\alpha}^i(x) = \alpha^i T^i(x), \quad i = 1..N \quad (13)$$

Здесь через  $T^i(x)$  обозначена функция выборки из текстуры. В текущей версии системы текстурные данные передаются на графический процессор в виде обычных массивов CUDA. Отказ от использования текстурной памяти ведет к некоторой потере производительности, но позволяет снять ограничения на число обрабатываемых текстур, их размер и формат пикселя. При таком подходе методы фильтрации должны быть реализованы программно. На данный момент система поддерживает интерполяцию по ближайшему элементу и билинейную интерполяцию.

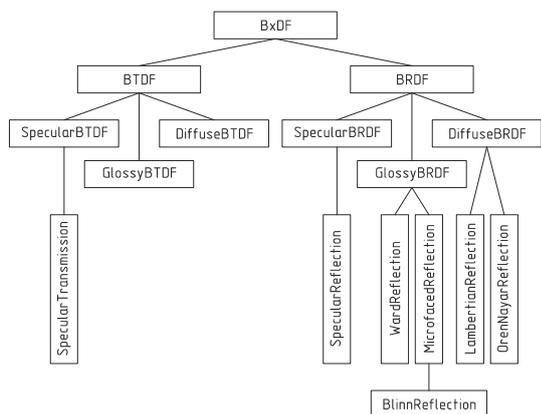


Рис. 4. Диаграмма классов для описания материалов.

Для эффективного расчета вторичного освещения используется выборка по значимости: для генерации направлений вторичных лучей плотность вероятности строится пропорционально BSDF поверхности. Для этого интеграл вторичного освещения разбивается на части, соответствующие отдельным BxDF компонентам:

$$L_{indirect}(x \rightarrow \Theta) = \sum_{i=1}^N \tilde{\alpha}^i(x) \int_{H_x} f_x^i(x, \Psi \rightarrow \Theta) L_r(r(x, \Psi) \rightarrow -\Psi) |\cos(N_x, \Psi)| d\omega_\Psi \quad (14)$$

Для оценки данного выражения используется следующий двухшаговый алгоритм:

- Конструируется дискретное распределение для  $N + 1$  события, вероятности которых равны  $q_1, q_2, \dots, q_{N+1}$ . Для  $i \in [1, N]$   $i$ -ое событие состоит в выборе  $i$ -ой компоненты интеграла освещенности. Последнее событие состоит в поглощении пути.
- Если не принято решение о поглощении пути, то случайное направление  $\Psi$  генерируется с помощью плотности вероятности  $p_i(\Psi)$ , пропорциональной  $i$ -ой BxDF компоненте материала.

Оценка энергетической яркости для выбранного направления  $\Psi$  будет иметь вид:

$$\langle L_{indirect}(x \rightarrow \Theta) \rangle = \begin{cases} \frac{f_x^i(x, \Theta \leftrightarrow \Psi) L_r(r(x, \Psi) \rightarrow -\Psi) |\cos(N_x, \Psi)|}{q_i p_i(\Psi)} & i = 1 \dots N \\ 0 & i = N + 1 \end{cases} \quad (15)$$

В настоящей работе вероятности  $q_i$  вычисляются на основе коэффициентов  $\tilde{\alpha}^i$  и текущего веса пути  $w(x)$ , который определяет вклад луча в яркость обрабатываемого пикселя:

$$q_i = \begin{cases} \frac{\tilde{\alpha}_R^i w_R(x) + \tilde{\alpha}_G^i w_G(x) + \tilde{\alpha}_B^i w_B(x)}{w_R(x) + w_G(x) + w_B(x)} & i = 1 \dots N \\ 1 - \sum_{i=1}^N q_i & i = N + 1 \end{cases} \quad (16)$$

Данный подход обеспечивает корректный баланс BxDF составляющих, учитывает потенциальный вклад текущего луча и влияние текстурных карт.

## 4. АНАЛИЗ РЕЗУЛЬТАТОВ

### 4.1 Проверка корректности

Для проверки корректности использовались сцены для тестирования алгоритмов глобального освещения [17], которые выявляют различные аспекты переноса света. Все изображения были сгенерированы методом стохастической трассировки путей (рис. 5). Для математической оценки

результатов в каждом случае вычислялась среднеквадратическая ошибка (RMSE, рис. 6).

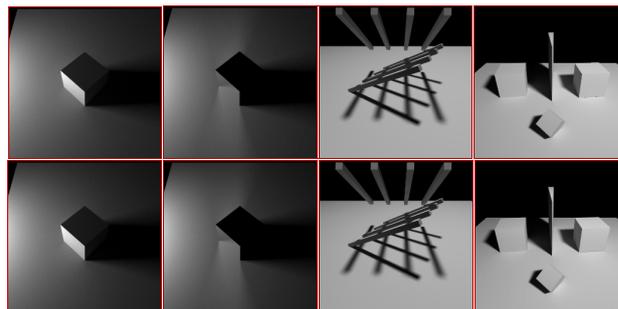


Рис. 5. Наши результаты (сверху) и эталонные (снизу): Secondary, Caustic, Shadow и Geometry.

Небольшие отклонения связаны с особенностями реализации датчика случайных чисел на графическом процессоре.

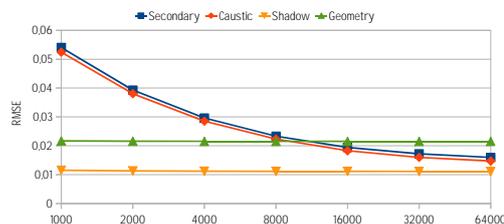


Рис. 6. Зависимость ошибки (RMSE) от числа итераций.

Тем не менее, полученные результаты позволяют говорить о корректности визуализации.

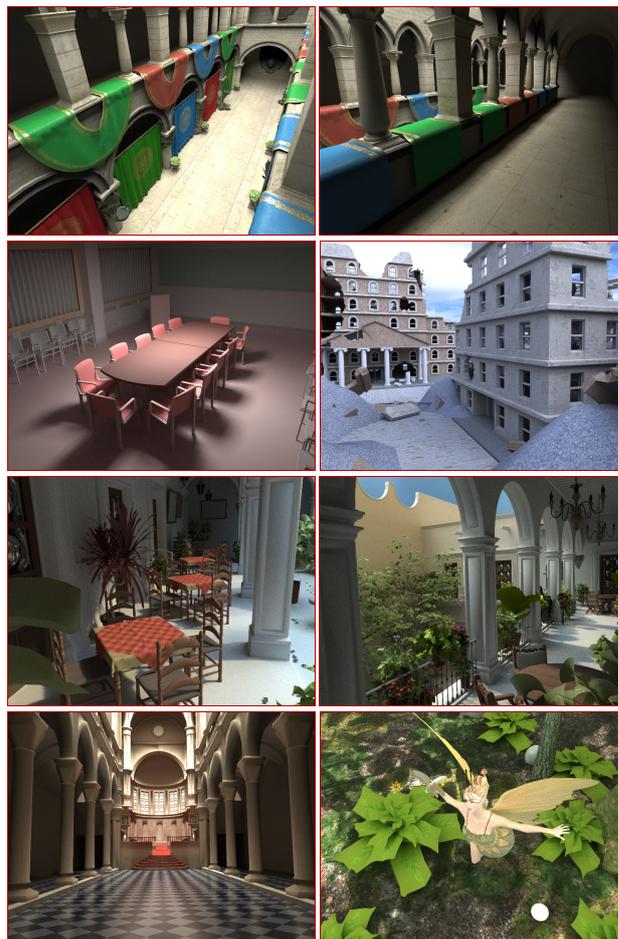


Рис. 7. Тестовые сцены и точки обзора: Crytek Sponza (2), Conference, Ruins, San Miguel (2), Sibenik, Fairy Forest.

## 4.2 Оценка производительности

Для оценки производительности использовались популярные сцены для сравнения алгоритмов глобального освещения (рис. 7). Модели содержат различные типы материалов, а также диффузные и зеркальные текстурные карты высокого разрешения. Изображения визуализировались в разрешении  $800 \times 600$  методом стохастической трассировки путей при максимальной глубине пути 10. Для расчетов использовался графический процессор начального уровня NVIDIA GeForce 560 1 Гб.

Test Scene	Triangles	MRays / sec	FPS
Crytek Sponza ( <i>outside</i> )	262 267	28.1	5.85
Crytek Sponza ( <i>inside</i> )	262 267	38.3	7.9
Conference Room	331 179	37.1	7.8
Ruins	336 967	41.4	8.5
San Miguel ( <i>ground floor</i> )	7 880 512	23.2	4.8
San Miguel ( <i>upper floor</i> )	7 880 512	20.2	4.2
Sibenik Cathedral	75 284	40.5	8.4
Fairy Forest	174 117	111.6	23.4

Таким образом, приемлемое для интерактивных приложений качество изображений (несколько десятков кадров) может быть получено за несколько секунд. Производительность может быть значительно повышена за счет сокращения максимальной длины путей до 2-3 отскоков с сохранением всех эффектов глобального освещения.

## 4.3 Дополнительная функциональность

Система поддерживает визуализацию методом двунаправленной трассировки путей, которая эффективна при расчете вторичного освещения. На рис. 8 показано сравнение обычной и двунаправленной трассировки в ситуациях, когда площадной источник направлен вниз и вверх. Во втором случае объекты получают лишь вторичное освещение.

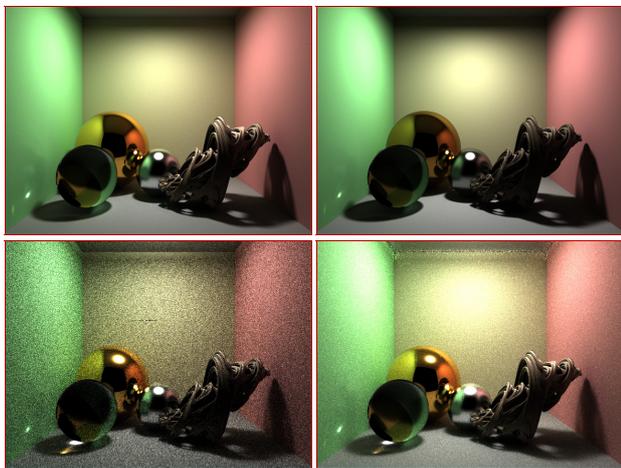


Рис. 8. Обычный (слева) и двунаправленный метод (справа)

Кроме того, данный рисунок демонстрирует возможность обработки таких неполигональных объектов, как сферы и фрактальные множества Джулия.

## 5. ЗАКЛЮЧЕНИЕ

В рамках настоящей работы разработана система интерактивной визуализации для сложных сцен гибридной природы с поддержкой глобального освещения и динамической геометрии. Под гибридностью понимается возможность сочетания традиционных полигональных моделей с объектами других типов, таких как полупрозрачные среды, поверхности Безье, изоуровни скалярных полей и поверхно-

сти второго порядка. В основе системы – гибкий графический конвейер, который допускает конфигурацию всех этапов формирования изображения. Настройка конвейера позволяет получать эффективные решения для различных задач – от расчета глобального освещения с расширенными моделями материалов и источников света до интерактивных систем с базовыми эффектами глобального освещения и динамической геометрии. Реализована межплатформенная версия системы на базе NVIDIA CUDA, которая обеспечивает высокую производительность построения ускоренных структур и визуализации гибридных сцен.

## 6. ЛИТЕРАТУРА

- [1] M. J. Keates, R. J. Hubbard. *Interactive ray tracing on a virtual shared-memory parallel computer* // Computer Graphics Forum 14, 4 (October 1995), 189-202.
- [2] M. Muuss. *Towards real-time ray-tracing of combinatorial solid geometric models* // In Proc. of BRL-CAD Symposium '95, 5-9 June 1995.
- [3] S. Parker, W. Martin, P. J. Sloan, P. Shirley, B. Smits, C. Hansen. *Interactive ray tracing* // In Proc. of Interactive 3D Graphics (1999), 119-126.
- [4] I. Wald, P. Slusallek, C. Benthin, M. Wagner. *Interactive Rendering with Coherent Ray tracing* // Computer Graphics Forum 20, 3 (2001), 153-164.
- [5] A. Reshetov, A. Soupikov, J. Hurley. *Multi-Level Ray Tracing Algorithm* // ACM Trans. Graph. 24, 3, 1176-85.
- [6] C. P. Gribble, T. Ize, A. Kensler, I. Wald, S. G. Parker. *A coherent grid traversal approach to visualizing particle-based simulation data* // IEEE Transactions on Visualization and Computer Graphics 13, 4, 758-768.
- [7] S. Boulos, D. Edwards, J. D. Lacewell, J. Kniss, J. Kautz, P. Shirley, I. Wald. *Packet based Whitted and Distribution Ray Tracing* // In Proc. of Graphics Interface 2007 (GI '07). ACM, New York, NY, USA, 177-184.
- [8] J. M. Singh, P. J. Narayanan. *Real-Time Ray Tracing of Implicit Surfaces on the GPU* // IEEE Transactions on Visualization and Computer Graphics 16, 2, 261-272.
- [9] J. Mensmann, T. Ropinski, K. Hinrichs. *An Advanced Volume Raycasting Technique using GPU Stream Processing* // International Conference on Computer Graphics Theory and Applications 2010, 190-198.
- [10] I. Wald, W. R. Mark, J. Günther, S. Boulos, T. Ize, W. Hunt, S. G. Parker, P. Shirley. *State of the Art in Ray Tracing Animated Scenes* // Computer Graphics Forum 28, 6 (2009), 1691-1722.
- [11] P. Shanmugam, O. Arikan. *Hardware accelerated ambient occlusion techniques on GPUs* // In Proc. of the 2007 symposium on Interactive 3D graphics and games, 73-80.
- [12] C. Dachsbacher, M. Stamminger. *Reflective shadow maps* // In Proc. of the 2005 symposium on Interactive 3D graphics and games (I3D '05), 203-231.
- [13] I. Radax. *Instant Radiosity for Real-Time Global Illumination*. Institute of Computer Graphics and Algorithms. Technical Report (May 2008).
- [14] R. Wang, R. Wang, K. Zhou, M. Pan, H. Bao. *An efficient GPU-based approach for interactive global illumination* // ACM Trans. Graph. 28, 3, Article 91.
- [15] M. McGuire, D. Luebke. *Hardware-Accelerated Global Illumination by Image Space Photon Mapping* // In Proc. of High Performance Graphics 2009 (HPG '09), 77-89.
- [16] D. Sopin, D. Bogolepov, D. Ulyanov. *Real-time SAH BVH construction for ray tracing dynamic scenes* // In Proc. of the GraphiCon'2011, 74-77.
- [17] B. Smits, H. W. Jensen. *Global Illumination Test Scenes* // Tech. Rep. UUCS-00-013, Computer Science Department, University of Utah, June 2000.