

## РАСЧЕТ ЯРКОСТИ КАУСТИЧЕСКОГО ОСВЕЩЕНИЯ НА ОСНОВЕ DIRECTX RAYTRACING

© 2022 г. Е. А. Комаров<sup>a,\*</sup> (ORCID: 0000-0003-3719-995X),  
Д. Д. Жданов<sup>a,\*\*</sup> (ORCID: 0000-0001-7346-8155), А. Д. Жданов<sup>a,\*\*\*</sup> (ORCID: 0000-0002-2569-1982)

<sup>a</sup> Национальный исследовательский университет ИТМО  
197101 Санкт-Петербург, Кронверкский пр., д. 49, лит. А, Россия

\*E-mail: komarov.99@yandex.ru

\*\*E-mail: ddzhdanov@mail.ru

\*\*\*E-mail: adzhdanov@itmo.ru

Поступила в редакцию 15.12.2021 г.

После доработки 16.01.2022 г.

Принята к публикации 22.01.2022 г.

В данной работе была исследована возможность реализации физически корректного рендеринга сцен с каустическим освещением средствами API DirectX Raytracing с последующей интеграцией его в конвейер рендеринга реального времени. Метод основан на использовании дополнительных карт видимости прямой каустики освещения и карт видимости обратной каустики, которые создаются для источников света и виртуальной камеры соответственно. В работе рассмотрена возможность использования метода обратных фотонных карт средствами API DirectX Raytracing и представлен алгоритм расчета яркости каустического освещения и результаты тестирования программной реализации данного алгоритма для различных параметров его настройки. Проведено исследование зависимости скорости рендеринга от параметров сцены и характеристик, определяющих качество расчета каустического освещения: количества лучей, испускаемых источником света, максимальной глубины зеркальной трассы луча и количества источников света в сцене. Результаты тестирования показали, что разработанный алгоритм и его программная реализация могут обеспечить высокое качество расчета каустического освещения в режиме реального времени, что позволяет их использовать в игровой индустрии для повышения реализма зрительного восприятия при визуализации сцен со сложными условиями освещения.

DOI: 10.31857/S0132347422030062

### ВВЕДЕНИЕ

Каустическое освещение окружает нас в повседневной жизни [1]. Это освещение возникает при преломлении или зеркальном отражении света от гладких поверхностей, например, свет, проходящий через окна или отраженный от стекла, создает каустическое освещение. Эффективными примерами каустического освещения могут служить солнечные зайчики или, как показано на рисунке 1, каустики, создаваемые светом после его преломления в бокале вина.

Несмотря на то, что каустическое освещение распространено в реальной жизни, его достаточно сложно моделировать средствами компьютерной графики, особенно если необходимо обеспечить рендеринг сцены в реальном времени, например, в компьютерных играх или системах виртуальной реальности. До недавнего времени методы рендеринга, используемые в компьютерных играх, базировались главным образом на методах растеризации, проецирующих объекты сце-

ны в виртуальную камеру [2]. Однако с появлением DirectX Raytracing (DXR) [3] стало возможным использовать API для трассировки лучей и ком-



Рис. 1. Пример каустики, созданной бокалом вина.

бинировать различные методы рендеринга в приложениях реального времени. В то же время NVIDIA представила технологию Turing для своих графических процессоров, позволяющую ускорить процесс поиска точки встречи лучей с поверхностями сцены до десяти раз по сравнению с решениями, используемыми в предыдущих поколениях [4]. Все это делает возможным использовать методы трассировки лучей в системах рендеринга реального времени и усиливать реализм изображений, формируемых данными системами. DirectX Raytracing содержит необходимую функциональность, которая позволяет адаптировать значительную часть существующих алгоритмов реалистичного рендеринга и методов физически корректной визуализации различных оптических эффектов в приложениях реального времени. В рамках данной работы мы представляем метод и описание его программной реализации, построенной на DirectX Raytracing API, для эффективного расчета яркости каустического освещения.

### ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ

Каустическое освещение является частью глобального освещения и для реализации реалистичного рендеринга с каустической компонентой освещения необходимо базироваться на физически корректных законах распространения и преобразования светового излучения. Для расчета яркости глобального освещения Джеймс Каджия в своей работе [5] предложил так называемое уравнение рендеринга (1). Это уравнение позволяет вычислить видимую яркость  $L(\vec{p}, \vec{v}_r)$  для каждого пикселя изображения  $\vec{p}$  в направлении наблюдения  $\vec{v}_r$  виртуальной камеры:

$$L(\vec{p}, \vec{v}_r) = \tau(\vec{p}) \frac{n_r^2}{n_0^2} \times \left( L_0(\vec{p}, \vec{v}_r) + \int_{\omega} L_i(\vec{p}, \vec{v}_i) f(\vec{p}, \vec{v}_i, \vec{v}_r) \cos(\vec{n}, \vec{v}_i) d\omega \right), \quad (1)$$

где  $L_0(\vec{p}, \vec{v}_r)$  – собственная яркость поверхности в точке  $\vec{p}$ ;  $\tau(\vec{p})$  – коэффициент пропускания среды на пути от виртуальной камеры до наблюдаемой поверхности;  $n_r$  – показатель преломления среды виртуальной камеры;  $n_0$  – показатель преломления среды на поверхности в направлении наблюдения;  $L_i(\vec{p}, \vec{v}_i)$  – яркость сцены в точке  $\vec{p}$  в направлении  $\vec{v}_i$ ;  $f(\vec{p}, \vec{v}_i, \vec{v}_r)$  – значение двунаправленной функции рассеивания (ДФР) поверхности для направления падающего света  $\vec{v}_i$  и направления наблюдения  $\vec{v}_r$ ;  $\vec{n}$  – нормаль к поверхности в точке наблюдения.

Поскольку уравнение рендеринга является интегральным уравнением с бесконечной рекурсией, оно традиционно решается методом Монте-Карло. Таким образом, формула (1) преобразуется в следующую формулу:

$$L(\vec{p}, \vec{v}_r) \approx \tau(\vec{p}) \frac{n_r^2}{n_0^2} \times \left( L_0(\vec{p}, \vec{v}_r) + \frac{1}{\pi \cdot N} \sum_{i=1}^N L_i(\vec{p}, \vec{v}_i) f(\vec{p}, \vec{v}_i, \vec{v}_r) \cos(\vec{n}, \vec{v}_i) \right), \quad (2)$$

где  $N$  – количество случайных выборов.

Уравнение рендеринга решается методами стохастической трассировки лучей. Существуют десятки методов решения этого уравнения, каждый из которых имеет свои преимущества и недостатки. Методы решения уравнения рендеринга можно классифицировать по нескольким основным признакам. Например, по методам интегрирования уравнения рендеринга: во-первых, это методы на основе обыкновенного Монте-Карло интегрирования и методы Монте-Карло интегрирования, построенные по схеме Марковских цепей. Во-вторых, эти методы можно подразделить на несмещенные, работающие, как правило, в терминах яркости, и смещенные, работающие, как правило, в терминах светового потока и осуществляющие интегрирование яркости в конечной области поверхности сцены. В данной работе не рассматривались методы интегрирования, построенные по схеме Марковских цепей. Были рассмотрены основные классические методы – это методы обратной стохастической трассировки лучей [6], прямой стохастической трассировки лучей [7] и двунаправленной стохастической трассировки лучей [8]. Эти методы могут служить основой для расчета яркости каустического освещения, однако их непосредственная реализация может быть неэффективна в гибридных решениях, используемых в игровой компьютерной графике. Другой подход, который наиболее удачно подходит для расчета яркости каустического освещения, это метод фотонных карт, основанный на методах двунаправленной стохастической трассировки лучей. Здесь стоит отметить метод прямых фотонных карт [9], обратных [10] фотонных карт и модификации классического метода двунаправленной трассировки лучей с использованием фотонных карт [11]. Существует ряд специализированных методов, позволяющих повысить эффективность расчета каустического освещения. Эти методы основаны как на использовании прямых фотонных карт [12], [13], так и обратных фотонных карт [14], [15]. Хотя метод фотонных карт и является смещенным методом, он очень эффективно решает проблему расчета яркости каустического освещения, а уменьшение радиуса сферы интегрирования позволяет снизить ошибку усредне-

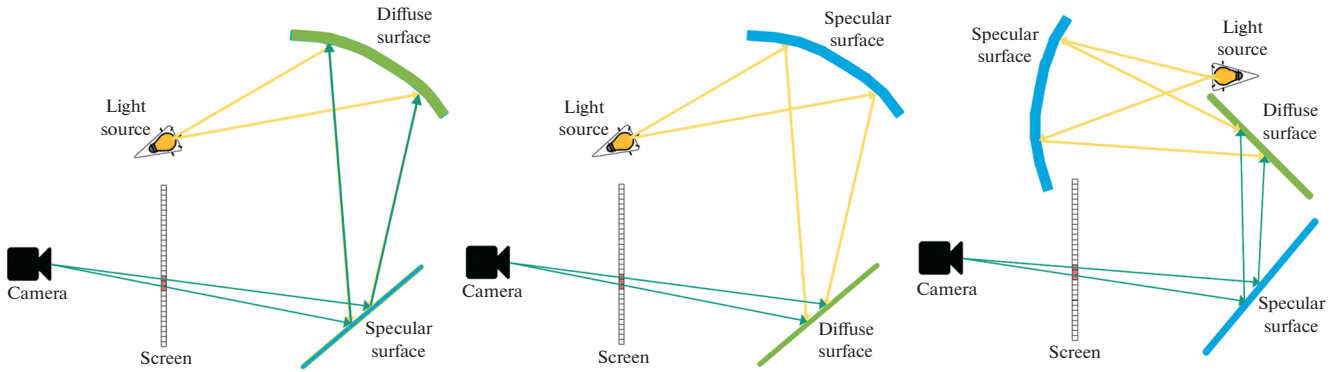


Рис. 2. Каустики наблюдения (слева), освещения (по центру) и смешенного типа (справа).

ния данного метода. Для игровой индустрии наиболее актуальным является визуализация каустики, вызванная отражением и преломлением света от неровной поверхности воды. Моделирование данного эффекта осуществляется путем предварительного расчета анимированной текстуры освещения воды и нормалей, а затем их проецирования на подводные поверхности [16]. Другой метод визуализации каустики в реальном времени — использование карт каустики [17], однако это решение ограничено одним событием зеркального отражения или преломления, например, визуализация солнечных зайчиков. Для ускорения создания каустических карт их также можно построить с использованием отложенного затенения [18].

При разработке реалистичного рендеринга в реальном времени на основе DirectX Raytracing необходимо следовать конвейеру трассировки лучей DXR и использовать соответствующий API [19]. Наиболее многообещающим методом рендеринга на CPU является прогрессивное обратное отображение фотонов [20], поскольку он использует меньше ресурсов при том же качестве рендеринга. Однако этот метод имеет ряд проблем, связанных с каустическим наблюдением, и из-за архитектуры, несовместимой с DXR, и ограниченности ресурсов видеокарты его невозможно использовать напрямую. Поэтому в данной работе предлагается другое решение, объединяющее методы растеризации с трассировкой лучей.

## МЕТОД РЕНДЕРИНГА

При решении уравнения рендеринга можно выделить два основных типа каустик. Это каустика наблюдения и каустика освещения. Кроме того, существует каустика смешанного типа. Первый тип — это каустика наблюдения, возникающая на “зеркальном” пути камерного луча до первого диффузного объекта. Снижение яркости данной каустики определяется параметром  $\tau(\vec{p})$

уравнения (1). Данный тип каустики легко вычисляется методами обратной трассировки лучей. Следующий тип каустики — это каустика освещения, возникающая на “зеркальной” трассе луча от источника света камерного луча до первого диффузного объекта. Снижение яркости данной каустики определяется параметром  $\tau(\vec{p})$ , неявно входящим в подынтегральную часть уравнения (1). Последний тип каустики — это каустика смешанного типа, возникающая при наблюдении и освещении диффузного объекта через “зеркальные” поверхности. Все типы каустик представлены на рисунке 2.

Обратные фотонные карты позволяют наиболее реалистично и эффективно моделировать каустики освещения и каустики смешанного типа. Однако для достижения высокой эффективности требуются значительные вычислительные ресурсы, связанные с формированием, хранением и обработкой фотонных карт. Непосредственная реализация метода обратных фотонных карт, показывающая высокую эффективность на CPU, выглядит малоэффективной на GPU. С другой стороны, идеи, заложенные в методе обратных фотонных карт, могут быть реализованы в гибридной модели рендеринга для каустического освещения (без компоненты каустического наблюдения). Общий алгоритм требует объединения методов трассировки лучей с каустическими картами. В результате разработанный метод рендеринга состоит из четырех основных этапов, интегрированных в единый конвейер рендеринга DirectX: расчет глобального освещения, расчет каустик освещения, расчет каустик смешанного типа и фильтрация изображения.

Данная работа посвящена расчету яркости каустического освещения, поэтому остальные элементы конвейера рендеринга будут представлены схематично.

Для расчета глобального освещения мы используем метод Deferred Shading, описанный в работах [21] и [22]. Основная идея заключается в однократ-

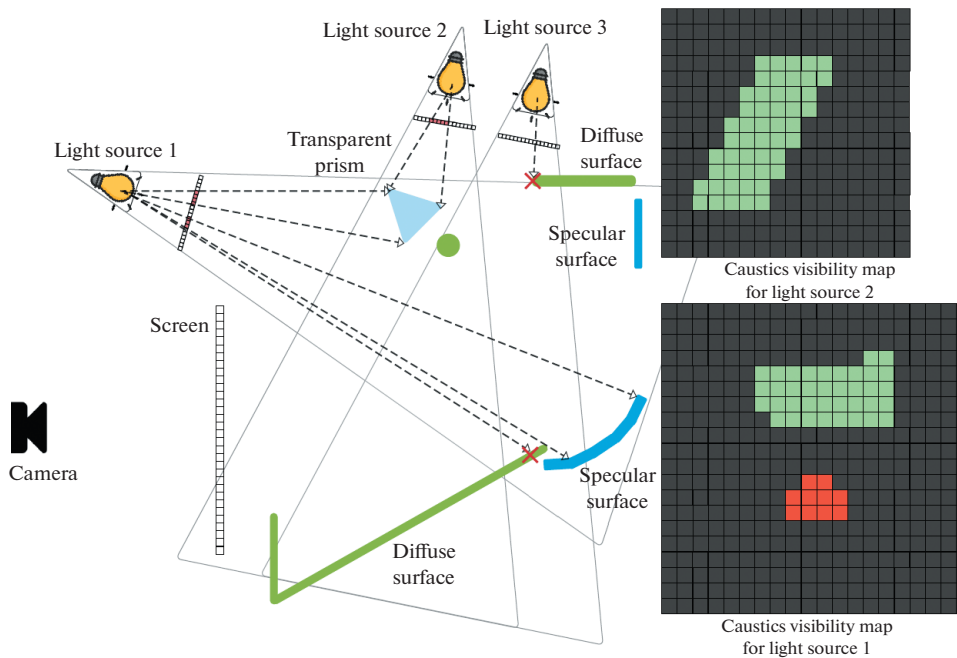


Рис. 3. Создание карт видимости каустики.

ном рендеринге всех видимых объектов сцены в несколько целей рендеринга (RT), которые называются геометрическими буферами (G-буферами). В этих буферах положение отображаемого объекта записывается для каждой точки изображения (обычно в виде расстояния от наблюдателя, нормали к поверхности и параметров материала, который участвует в вычислении яркости). В простейшем случае расчет глобальной освещенности состоит из трех шагов: создание G-буферов, расчет затенения, прозрачности и отражений, окончательный расчет яркости.

### РАСЧЕТ ЯРКОСТИ КАУСТИКИ ОСВЕЩЕНИЯ

Каустики освещения предполагают, что каустическое освещение наблюдается непосредственно и может быть спроецировано на буфер экрана, поэтому для ее расчета требуется вычисленная глобальная освещенность и G-буфер карты глубины экрана.

На следующем этапе необходимо выполнить проекцию источников света на локальные карты экранов (для каждого источника света создается собственный экран) и создать на них G-буферы (или карты видимости) для карт глубины, нормалей поверхности и оптических свойств поверхности света. Информация записывается только для “зеркальных” поверхностей, видимых из источника света, и формирует плотность вероятности каустического преобразования луча. Для точек, в

которых не было найдено пересечение с геометрией сцены или находились поверхности с чисто диффузными свойствами, плотность вероятности продолжения каустической трассировки лучей равна нулю. При наличии зеркальных свойств поверхности плотность вероятности определяется в соответствии с зеркальной двунаправленной функцией рассеивания (ДФР) и разбивается на две части: вероятность отражения и вероятность преломления. Если источник света вообще не видит зеркальных поверхностей, то он исключается из процесса расчетов. Для ускорения процесса рендеринга создание G-буферов источников света может выполняться одновременно с вычислением затенения сцены. На рисунке 3 демонстрируется процесс создания G-буферов. На данном примере для третьего источника света карта видимости не создается, поскольку он не формирует каустическое освещение.

Использование аппаратных средств растеризации значительно сокращает время на создание карт видимости каустики и позволяет быстро сформировать области трассировки каустических лучей и соответствующие функции распределения вероятностей для дальнейшей трассировки каустических лучей. Для ускорения процесса трассировки источники света с одинаковыми разрешениями сетки G-буферов объединяются в массивы текстур, и первичная генерация лучей выполняется по сетке токов карт видимости. Происходит сканирование карты видимости и для точек с не нулевой плотностью вероятности, считанной с G-буфера, выбирается событие отра-

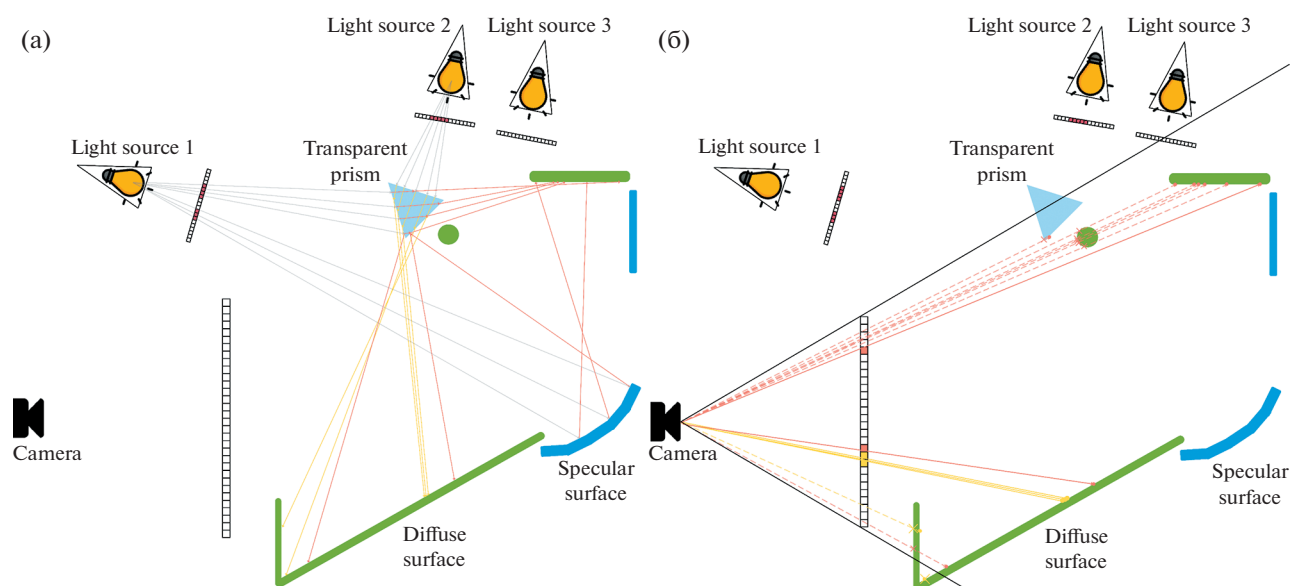


Рис. 4. Прямая трассировка лучей (а) и оценка видимости каустики (б).

жения или пропускания. Для выбора события используется метод выборки по значимости. Направление отраженного или преломленного луча определяется в соответствии с направлением падения луча, оптическими свойствами поверхности и направлением нормали, хранящихся в соответствующих точках G-буферов.

Затем происходит трассировка каустических лучей до следующего пересечения с поверхностью сцены. Далее карты видимости уже не строятся. Происходит анализ оптических параметров объектов сцены в точке попадания луча на его поверхность. Возможны следующие варианты:

1. Луч попал на поверхность с зеркальными свойствами. В данной ситуации луч должен продолжить свое распространение в сцене. Для этого вычисляются оптические и геометрические параметры сцены в точке на поверхности (ДФР и локальная нормаль). В общем случае у луча есть два варианта для дальнейшего распространения: отражение и преломление. По аналогии с первым преобразованием в точке поверхности, видимой источником света, формируются две вероятности на зеркальное отражение и преломление, и в соответствии с алгоритмом выборки по значимости определяется дальнейшая судьба луча (отражение или преломление). И в соответствии с этим выбором определяется направление дальнейшего распространения луча. Поскольку число “зеркальных” поверхностей на трассе каустического луча может быть достаточно большим, имеет смысл ограничить число “зеркальных” преобразований луча. Их число, с одной стороны ограничивается методом русской рулетки и соответствующим событием поглощения луча, а с другой стороны, об-

щим числом актов зеркального взаимодействия с объектами сцены. Алгоритм трассировки лучей показан на рисунке 4 (а).

2. Луч попал на поверхность с диффузными свойствами. В точке пересечения луча с поверхностью происходит оценка ее видимости из камеры. Координаты точки преобразуются из мирового пространства в пространство камеры с помощью соответствующих матриц. В результате получают расстояние до камеры и координаты точки изображения, из которой видно это пересечение. Поскольку карта глубины со стороны камеры уже была предварительно рассчитана и записана в соответствующий G-буфер, то для определения видимости достаточно сравнить эти данные между собой. Если точка пересечения непосредственно видна из камеры, то в этой точке формируется яркость каустического освещения. Таким образом, обратная трассировка лучей заменяется более быстрой растеризацией, как показано на рисунке 4 (б).

3. Луч попал на поверхность со смешанными оптическими свойствами, имеющими как диффузные, так и зеркальные свойства. В этом случае происходит оценка видимости точки пересечения луча с поверхностью и формируется яркость каустического освещения в соответствии с диффузной частью оптических свойств поверхности (смотри пункт 2). Затем, в соответствии с зеркальной частью оптических свойств поверхности, луч продолжает свое распространение в сцене до тех пор, пока критерий завершения его траектории не остановит его движение (смотри пункт 1).

Результирующая яркость каустики, переносимая каждым прямым лучом, вычисляется в соот-

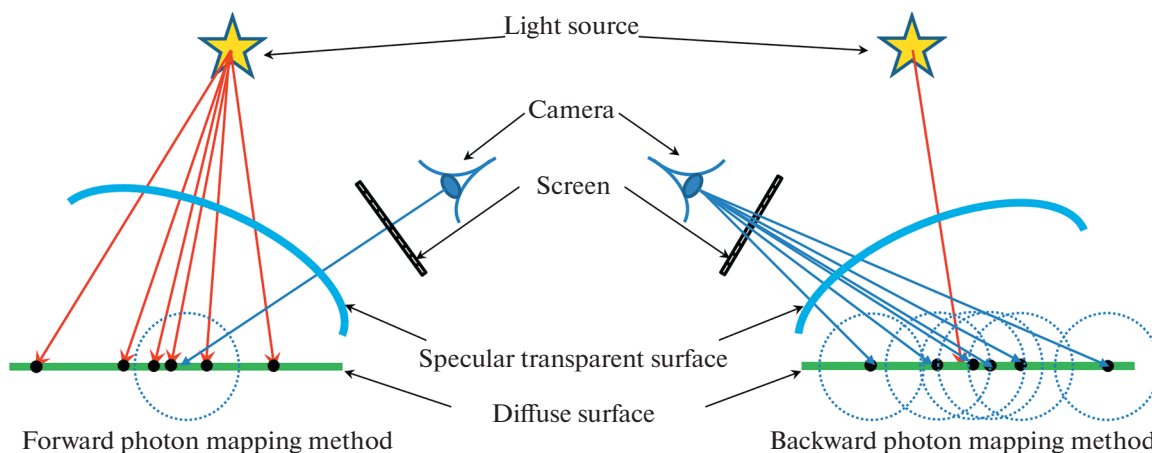


Рис. 5. Принципы формирования прямых и обратных фотонных карт и расчета яркости каустического освещения.

ветствии с формулой (2) и атомарно добавляется к ранее накопленной яркости в соответствующей точке изображения.

#### РАСЧЕТ КАУСТИКИ СМЕШАННОГО ТИПА И ФИНАЛЬНАЯ ОБРАБОТКА ИЗОБРАЖЕНИЯ

Для расчета каустик смешанного типа используется двунаправленная трассировка лучей с фотонными картами. При этом могут использоваться как прямые, так и обратные фотонные карты. Прямые фотонные карты (или карты освещения) формируются на трассах лучей, испущенных источником света, и, как правило, представляют собой набор данных о параметрах луча и поверхности в точке пересечения. При этом основные параметры луча — это его цвет, направление и координаты на поверхности, а параметры поверхности — это направление локальной нормали и оптические свойства поверхности в точке пересечения. Каустические фотонные карты формируются на диффузных объектах сцены до первого события диффузного рассеивания включительно, при условии, что произошел, как минимум, один акт “зеркального” преобразования луча на трассе до диффузной поверхности. Обратные фотонные карты (или карты видимости) формируются аналогичным образом на трассах лучей, выпущенных из камеры. Основные отличия прямых фотонных карт от обратных это то, что цвет луча является коэффициентом пропускания цвета на трассе луча от камеры до диффузной поверхности, а сама карта имеет дополнительный параметр — радиус сферы интегрирования яркости каустического освещения. Принцип формирования прямых и обратных фотонных карт и расчета яркости каустического освещения представлен на рисунке 5.

Принципиальное отличие расчета яркости методом обратных фотонных карт от расчета методом прямых фотонных карт заключается в:

- 1). методе сбора каустического освещения. В методе прямых фотонных карт точка экрана собирает освещенность со всех фотонов карты, попавших сферу интегрирования луча, выпущенного из камеры, а в методе обратных фотонных карт световой луч распределяет свою освещенность по точкам экрана, в сферы интегрирования которых он попал;
- 2). наличии сфер интегрирования, используемых в методе обратных фотонных карт, что позволяет ускорить процесс поиска пересечения луча с фотонной картой;
- 3). дополнительной синхронизации для многопоточковых приложений при записи значений яркости в одинаковые точки изображения (т.к. в методе обратных фотонных карт световой луч передает свою освещенность сразу нескольким точкам экрана).

Выбор реализации прямых или обратных фотонных карт зависит от конкретной сцены (какую часть каустически освещенной сцены видит камера) и архитектуры приложения, в которой будет реализован данный метод.

В рамках данной работы был предложен алгоритм, основанный на методе обратных фотонных карт с использованием G-буферов карт глубин глобального освещения и источников света. Алгоритм расчета каустики смешанного типа базируется на методе трассировки световых лучей, формируемых для расчета яркости каустического освещения и обратных фотонных карт. Для реализации алгоритма на диффузных поверхностях сцены дополнительно создается обратная фотонная карта объектов сцены, видимых камерой через зеркальные поверхности. Для ускорения расчета каустические лучи камеры трассируются



Рис. 6. Результаты рендеринга без учета каустики (слева) и с учетом каустики (справа).

только из тех точек экрана, для которых в соответствующем G-буфере определены зеркальные свойства видимых поверхностей сцены. Кроме того, для ускорения расчета лучи стартуют с точек поверхности, также хранящихся в G-буфере. Метод трассировки обратных лучей, выпущенных из камеры, полностью соответствует методу трассировки прямых лучей, выпущенных источниками света. Критерий остановки процесса трассировки обратных лучей также аналогичен методу прямой трассировки лучей (достигнута максимальная глубина трассы луча, выбрано событие поглощения или поверхность не имеет зеркальных свойств). Начиная со второй поверхности трассы обратного луча создаются карты видимости на диффузных поверхностях сцены, которые включают в себя (помимо информации о свойствах поверхности и параметрах луча в точке встречи его с поверхностью) радиусы сфер интегрирования каустического освещения. Данные радиусы определяются по угловому размеру пикселя на расстоянии от камеры до поверхности.

После того, как для всех точек экрана была сформирована обратная фотонная карта, создается двухуровневая ускоряющая структура, где структура нижнего уровня (BLAS) группирует фотоны в пределах одного объекта сцены, а структура ускорения верхнего уровня (TLAS) строится над набором BLAS. Каждый BLAS представлен одним и тем же типом сферического объекта и матрицей преобразования, содержащей 12 чисел с плавающей точкой. Последним шагом выполняется трассировка световых лучей с, испускаемых с G-буферов источников света, расчет каустики освещения и одновременно проверка пересечения трасс лучей со сферами интегрирования фотонных карт. При попадании световых лучей в сферы интегрирования освещенность, созданная в сфере интегрирования, передается в соответствии с ДРФ на точки экрана, связанные с данными сферами.

Использование стохастических методов синтеза изображения неизбежно приводит к его зашумлению. Основные методы для подавления шума это, во-первых, увеличение количества семплов на пиксель камеры и источника света,

во-вторых, анализ сцены и использование методов выборки по значимости, ориентированных не на локальное значение ДФР, а на распределение яркости результирующего изображения, и, в-третьих, фильтрация изображения. Поскольку первые два решения достаточно трудоемкие, в данной работе было отдано предпочтение методам фильтрации изображения [23]. Наиболее многообещающим фильтром является NVIDIA Real-Time Denoiser (NRD) [24], который использует историю из 32 кадров вместе с дополнительными данными изображения и требует всего около 5 мс для фильтрации одного кадра  $1920 \times 1080$ .

#### РЕЗУЛЬТАТЫ РЕНДЕРИНГА С УЧЕТОМ ЯРКОСТИ КАУСТИЧЕСКОГО ОСВЕЩЕНИЯ

Описанные выше алгоритмы были реализованы на базе рендерера Microsoft Mini-Engine [25]. Этот рендерер основан на DirectX Raytracing и распространяется под лицензией MIT. Кроме реализации алгоритма расчета яркости каустического освещения следующие основные изменения были внесены в рендерер Microsoft Mini-Engine:

- реализована реалистичная модель поверхности на основе ДФР [26], и физически корректный метод расчета яркости, включая вероятностную модель распространения лучей.
- реализован импорт файлов формата GLTF,
- реализован метод deferred shading.

Тестирование проводилось на мобильном графическом процессоре NVIDIA GeForce RTX 2060. Фильтрация изображения была отключена. Тестовая сцена, которая использовалась в данной работе, – это модифицированная сцена Srytek Sponza. На рисунке 6 показано сравнение результатов рендеринга без учета каустики и с ее учетом, рассчитанной предложенным методом.

Метод расчета яркости каустического освещения полностью масштабируем, и при разрешении каустического G-буфера  $512 \times 512$  точек, единичной глубине трассы каустического луча и одном луче с G-буфера каждый источник света, видимый с виртуальной камеры, добавляет 1.3 мс к

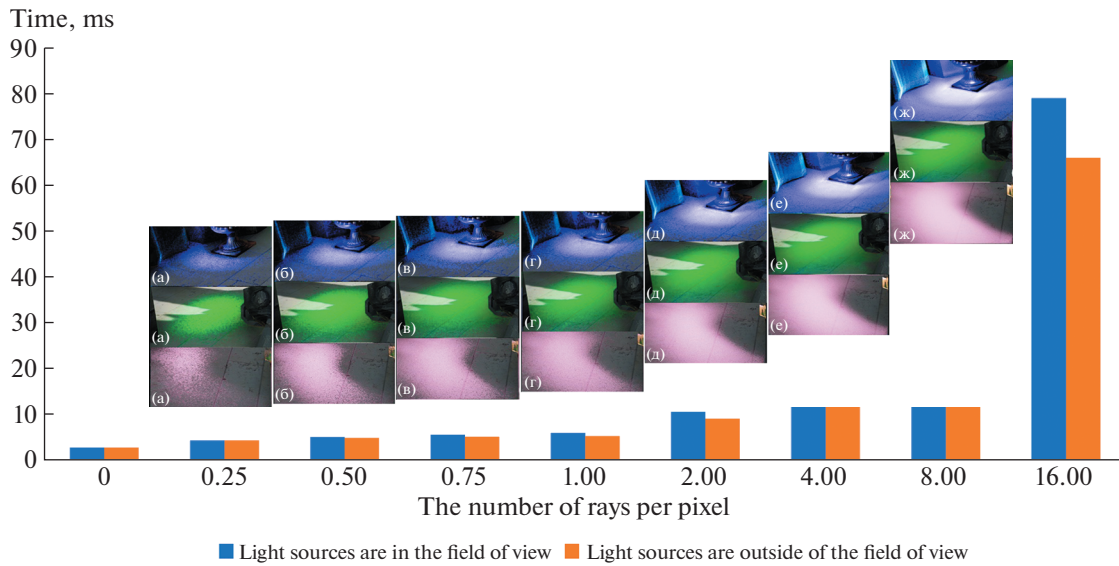


Рис. 7. Качество рендеринга каустического освещения в зависимости от количества лучей на пиксель.

времени рендеринга, а если он не виден, то добавляет 1.05 мс.

На рисунке 7 представлена зависимость качества каустического рендеринга от числа лучей, испускаемых с каустического G-буфера, и соответствующее время, необходимое для этого рендеринга. Видно, что одного луча, испускаемого с пикселя G-буфера, достаточно для качественной визуализации каустического освещения при приемлемом времени рендеринга. Кроме того, видимый шум изображения можно уменьшить с помо-

щью существующих алгоритмов фильтрации в реальном времени.

На рисунке 8 представлены зависимости времени и качества рендеринга каустического освещения от максимальной глубины зеркальной трассы луча. Видно, что увеличение глубины зеркальной трассы до значений более четырех не дает существенного повышения качества изображения. Нелинейный рост времени рендеринга на графике показывает, что в большинстве случаев трассировка лучей останавливается до достижения максимальной глубины.

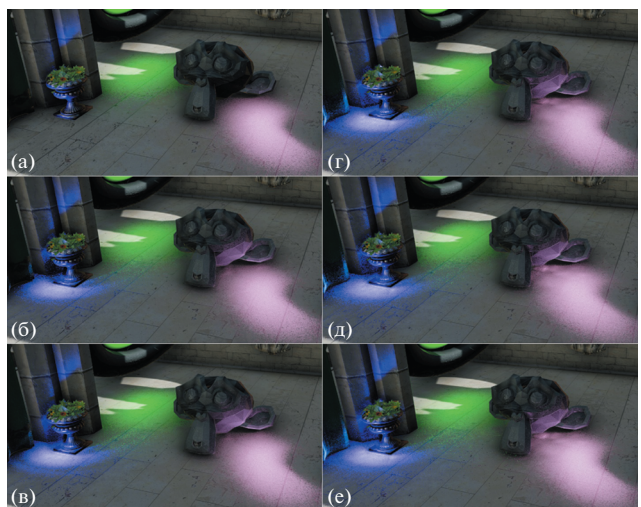
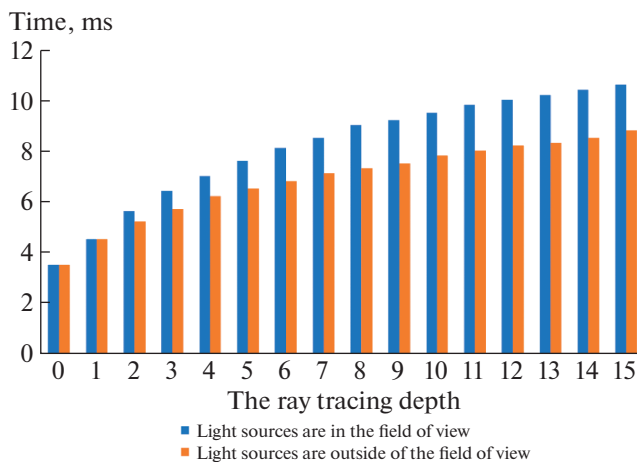


Рис. 8. Время рендеринга и его качество в зависимости от максимальной глубины трассировки каустических лучей: одно зеркальное событие (а), два зеркальных события (б), три зеркальных события (в), четыре зеркальных события (г), восемь зеркальных событий (д) и 16 зеркальных событий (е).



Исследование показало, что разработанный метод рендеринга может обеспечить реалистичный рендеринг сцен с каустическим освещением в режиме реального времени на графическом процессоре NVIDIA RTX среднего уровня. Были определены оптимальные параметры рендеринга: глубина трассировки лучей составляет три “зеркальных” события при испускании одного луча с G-буфера. Количество лучей на пиксель G-буфера можно снизить до 0.5 при использовании методов фильтрации изображения.

### ЗАКЛЮЧЕНИЕ

В рамках данного исследования был разработан метод реалистичной визуализации яркости каустического освещения в реальном времени на основе API DirectX Raytracing и его возможная интеграция в конвейер визуализации DirectX Raytracing. Основным параметром, линейно ограничивающим скорость визуализации, является количество источников света в сцене. Результаты тестирования показали, что при небольшом количестве источников в сцене, предложенное программное решение может работать в режиме реального времени со скоростью до 60 кадров в секунду и обеспечивать высокое качество рендеринга. Разработанные алгоритмы могут быть использованы при создании современных игр и графических приложений систем виртуальной реальности, что должно создать естественные условия погружения в игру или в виртуальный мир.

В рамках дальнейшего развития алгоритмов расчета яркости каустического освещения будет рассмотрена возможность ее расчета в режиме реального времени с использованием метода обратных фотонных карт на основе API DirectX Raytracing.

### СПИСОК ЛИТЕРАТУРЫ

1. *Born M., Wolf E.* Foundations of geometrical optics // Principles of optics: electromagnetic theory of propagation, interference and diffraction of light. Cambridge University Press, Cambridge. 1999. p. 134–135.
2. Microsoft, Rasterization Rules (Direct3D 9), 2018. URL: <https://docs.microsoft.com/en-us/windows/win32/direct3d9/rasterization-rules>.
3. Microsoft, D3D Team. Announcing Microsoft DirectX Raytracing! Microsoft Developer Blogs. 2018. URL: <https://devblogs.microsoft.com/directx/announcing-microsoft-directx-raytracing/> (дата обращения: 12.01.2022).
4. NVIDIA Corporation. NVIDIA Turing GPU Architecture // WP-09183-001\_v01. 2018.
5. *Kajiya J.T.* The rendering equation // SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques. August 1986. P. 143–150.
6. *Kay T.L., Kajiya J.T.* Ray tracing complex scenes // ACM SIGGRAPH computer graphics. 1986. Vol. 20. No. 4. P. 269–278.
7. *Chattopadhyay S., Fujimoto A.* Bi-directional ray tracing // Computer Graphics 1987. Springer, Tokyo. 1987. P. 335–343.
8. *Lafortune E.P., Willems Y.* Bi-directional path tracing // Compugraphics' 93. 1993. p. 145–153.
9. *Jensen H.W., Christensen P.* High quality rendering using ray tracing and photon mapping // ACM SIGGRAPH 2007 courses. 2007.
10. *Zhdanov D.D., Zhdanov A.D.* The backward photon mapping for the realistic image rendering // CEUR Workshop Proceedings. 2020. Vol. 2744. P. 1–12.
11. *Georgiev I., Krivánek J., Davidovic T., Slusallek P.* Light transport simulation with vertex connection and merging // ACM Trans. Graph. 2012. Vol. 31. No. 6. p. 192:1–192:10.
12. *Zhdanov A., Zhdanov D., Galaktionov V.A.* Bidirectional Ray Tracing with Caustic Photon and Indirect Imphoton Maps // Proceedings of the 14th IADIS International Conference Computer Graphics, Visualization, Computer Vision and Image Processing 2020, CGVCVIP 2020 and Proceedings of the 5th IADIS International Conference Big Data Analytics, Data Mining and Computational Intelligence 2020, BigDaCI 2020 and Proceedings of the 9th IADIS International Conference Theory and Practice in Modern Computing 2020, TPMC 2020 – Part of the 14th Multi Conference on Computer Science and Information Systems, MCCSIS 2020. 2020. P. 212–216
13. *Zhdanov A., Zhdanov D., Birukov E.* The realistic rendering with forward and backward photon mapping // Keldysh Institute Preprints. 2020. № 77. 22 с.
14. *Jendersie J., Rohmer K., Brüll F., Grosch T.* Pixel cache light tracing // Proceedings of the conference on Vision, Modeling and Visualization (VMV '17). Eurographics Association, Goslar, DEU. 2017. P. 137–144.
15. *Zhdanov A.D., Zhdanov D.D., Sorokin M.I.* The virtual prototyping of complex optical systems on multiprocessor workstations // Proceedings of SPIE. 2021. Vol. 11875. P. 118750C
16. *Iwasaki K., Dobashi Y., Nishita T.* A fast rendering method for refractive and reflective caustics due to water surfaces // Eurographics. 2003. P. 283–291.
17. *Shah M.A., Konttinen J., Pattanaik S.* Caustics Mapping: An Image-Space Technique for Real-Time Caustics // Transactions on Visualization and Computer Graphics. 2007. Vol. 13. P. 272–280.
18. *Wyman C., Nichols G.* Adaptive caustic maps using deferred shading // Computer Graphics Forum. 2009. Vol. 28. No. 2.
19. Microsoft. DirectX Raytracing (DXR) Functional Spec. 2021. URL: <https://microsoft.github.io/DirectX-Specs/d3d/Raytracing.html> (дата обращения: 12.01.2022).
20. *Potemin I.S., Zhdanov D.D., Zhdanov A.D., Bogdanov N.N., Voloboy A.G.* Hybrid ray tracing method for photoreal-

- istic image synthesis in head-up displays // Proceedings of SPIE, Optical Design and Engineering VII. International Society for Optics and Photonics. 2018. Vol. 10690. P. 106900I.
21. *Akenine-Moller T., Haines E., Hoffman N., Pesce A., Iwanicki M., Hillaire S.* Real-Time Rendering, 4th ed. // Boca Raton: A K Peters/CRC Press. 2018.
  22. *Haines E., Akenine-Moller T.* Ray Tracing Gems High-Quality and Real-Time Rendering with DXR and Other APIs // Apress. 2019.
  23. *Hasselgren J., Munkberg J., Salvi M., Patney A., Lefohn A.* Neural Temporal Adaptive Sampling and Denoising // Computer Graphics Forum. May 2020. No. 39(2). P. 147–155.
  24. *Zhdan D.* Fast Denoising with Self Stabilizing Recurrent Blurs // ГТС. 2020.
  25. Microsoft. DirectX-Graphics-Samples, GitHub. URL: <https://github.com/microsoft/DirectX-Graphics-Samples> (дата обращения: 12.01.2022).
  26. *Pharr M., Jakob W., Humphreys G.* Physically Based Rendering: From Theory to Implementation // San Francisco, Morgan Kaufmann Publishers Inc. 2016.