ФОТОЭЛЕКТРОНИКА
PHOTOELECTRONICS

УДК 621.384.3 PACS: 07.05.Tp
DOI: 10.51368/2307-4469-2023-11-5-433-445
EDN: TLFFBA

Моделирование взаимодействия излучения со средой в программном комплексе симуляции трехмерных динамических сцен

А. В. Зверев, Д. Е. Ипатов

Реализована поддержка трассировки пути в разработанном симуляторе трехмерных динамических сцен. Данный метод позволяет получить физически корректные изображения с учетом многократных взаимодействий электромагнитных волн с телами. Представлены результаты расчета нескольких сцен и оценка быстродействия аппаратно-ускоренной трассировки пути в видимом диапазоне длин волн.

Ключевые слова: моделирование, ФПУ, 3D, трассировка лучей, трассировка пути, Vulkan.

Введение

На сегодня робототехника широко применяется в различных областях человеческой деятельности [1–5], а поставленные перед ней задачи предполагают обработку данных, получаемых из окружающей нас действительности. Интегрированный в аппаратно-программный комплекс алгоритм использует множество сенсоров для формирования картины мира и последующего отклика на основе заготовленных шаблонов поведения [2, 3].

Традиционно одним из наиболее информационно мощных каналов данных о мире считается его визуальное представление [6]. Научная дисциплина – компьютерное зрение – основана на теории создания машин, обладающих способностью анализировать наблюдаемые образы [7]. Системы машинного зрения решают целый ряд задач, в которых входные данные имеют характерные особенности. Это позволяет сформировать суждения о наблюдаемой сцене, пригодные для совершения действий с допустимым уровнем принятия ошибочного решения.

В большинстве актуальных задач робот фактически не способен выполнять необходимый анализ поступающих сенсорных данных. Высокие запросы к самостоятельности и длительности автономной работы требуют от вычислительных средств способности адекватно оценивать окружающий мир и принимать решение, соответствующее поставленной задаче.

При разработке искусственных органов восприятия мы невольно делаем акцент на собственные, человеческие способности по анализу результата их работы. Прошедший множество часов обучения оператор способен выполнить максимально допустимый анализ зашумленного изображения, непрерывно корректируя выходной сигнал с помощью регулируемых параметров сенсора, поскольку он понимает, что именно он наблюдает.

Отсутствие таких фундаментальных понятий как жизненный опыт и целеполагание в полной мере раскрывают «блеск и нищету» современных автоматизированных систем. Текущие достижения микроэлектроники позволяет обеспечить необходимую глубину обработки данных, но при этом с трудом позволяет достичь требуемого быстродействия в

Зверев Алексей Викторович, рук. отдела, к.ф.-м.н.
Ипатов Дмитрий Евгеньевич, разработчик.
E-mail: dipatov@motivnt.ru
ООО «Мотив-НТ».
Россия, 121205, Москва, б-р Большой (Сколково инновационного центра тер), 42, стр. 1.

Статья поступила в редакцию 6.09.2023
После доработки 21.09.2023
Принята к публикации 27.09.2023

условиях ограниченных энергетических ресурсов, поэтому уровень аналитических способностей остаётся низким [7]. В этих условиях достижение требуемых показателей производительности возможно за счет переноса задачи глубокой обработки информации непосредственно на сенсор.

Поэтому к используемым в робототехнике сенсорам изображения предъявляются гораздо более строгие требования. На данный момент существует целый ряд различных технических приемов, призванных изменить текущее положение дел в видимом спектре [8–15]. При этом все перспективные решения, показавшие свои колоссальные преимущества в ряде задач, применяются только к сенсорам видимого диапазона длин волн. Фотоприемные устройства, чувствительные к средне- и длинноволновому участку инфракрасного (ИК) диапазона электромагнитных длин волн также представляют большой интерес для их тесной интеграции в современные интеллектуальные решения.

В литературе представлено множество работ, показывающих значительные преимущества при переходе от видимой части спектра в инфракрасную, особенно в задачах со сложными климатическими условиями [6, 16, 17]. Отдельное внимание заслуживает пример адаптации измерения временного контраста на изображении на основе болометрического сенсора [18], но очень низкое быстродействие чувствительных элементов теплового сенсора не способны в полной мере проявить преимущества данного подхода.

Вопрос разработки принципиально новых устройств чувствительных к ИК-области спектра требует проведения серии модельных экспериментов с целью поиска оптимального варианта реализации конструкций и архитектур. Но подобную работу невозможно выполнить на основе существующих данных, поскольку они уже содержат архитектурные особенности прибора в виде ограниченного динамического диапазона и утерянной информации о динамике наблюдаемой сцены.

Для этого необходимо подготовить колоссальный набор физически корректных данных с широким динамическим диапазоном величин. Эти данные могут быть использованы в качестве эталонных данных для оценки качества работы как самого фоточувствительного сенсора, так и для систем на их основе, которые будут способны выполнять анализ его отклика с целью принятия решения, соответствующего поставленной задаче. В зарубежной литературе подобное направление получило общее название гиперспектральная отрисовка изображений или Hyperspectral Imaging, HSI [19–31]. В данном направлении решается задача моделирования физически корректного взаимодействия электромагнитного излучения в широком диапазоне длин волн, обычно от ультрафиолета до дальнего ИК или в области от 0,2 до 20 мкм.

В рамках обзора общедоступных научных публикаций не удалось обнаружить информации о разрабатываемых либо уже разработанных отечественных программных комплексах, выполняющих задачу гиперспектрального моделирования сцен как для метеорологических и космических задач, так и для поведеческого моделирования современных видов фотоприемных устройств на основе синтетических данных.

Поэтому необходимо разработать собственный программный комплекс для моделирования трёхмерных сцен, способный сформировать набор физически корректных гиперспектральных входных данных как для исследования моделей сенсора, так и для формирования базы гиперспектральных сигнатур различных объектов для задач компьютерного зрения в сфере машинного обучения. В данной работе представлено применение в разработанном программном комплексе техники трассировки пути для получения наиболее физически корректного результата отрисовки в моделировании сцен видимого диапазона длин волн.

Симулятор сцены

В предыдущих работах [32, 33] нами представлен программный комплекс для моделирования сложных динамических ИК сцен в широком диапазоне длин волн. *Визуализатор*, как основной вычислительный блок программного комплекса, последовательно выполняет расчет модулей, каждый из которых содержит в себе определённую математическую модель:

- «Сцена» – трёхмерная модель сцены, на которой находятся тела с тепловыми характеристиками, источники излучения, модель атмосферы и модель камеры с оптической системой, определяющая поле зрения и спектральные характеристики оптического тракта;
- «Фотоприемное устройство» (ФПУ) – модель фотоприемного устройства заданной топологии элементов, пространственного разрешения, характеристик, механизмов регистрации и считывания сигнала излучения, падающего на фокальную плоскость сенсора;
- «Модуль электронной обработки» (МЭО) – модель обработчика данных, получаемых с ФПУ. Включает в себя математическую обработку поступающего от ИК ФПУ сигнала, для последующего представления результатов в требуемом виде.

На рис. . ***1*** представлена диаграмма активности основного цикла работы визуализатора при расчете одного кадра сцены до получения изображения с широким динамическим диапазоном.

Рис. 1. Диаграмма активности основного цикла работы визуализатора с подробным описанием действий в рамках модуля «Сцена»

Каждый модуль выполняет набор математических операций над входными данными с применением вычислительных мощностей графического процессора (ГП). Для этого в модуле задаётся последовательность графических преобразований данных, называемая *графическим конвейером*, который обязательно загружается в ГП перед началом исполнения инструкций, записанных в буферы команд. Непосредственно инструкции для выполнения различных операций на ГП представлены специальными программами, называемыми *шейдерами.*

В случае модуля «Сцена» входными данными является трехмерная модель сцены в формате glTF [34], в которой хранятся *вершины* (координаты точек, определяющих сложную геометрическую форму тела), их свойства, материалы, анимация, задающая динамическую составляющую, и другие компоненты, необходимые для создания качественного виртуального театра. Графический конвейер модуля «Сцена» для отрисовки методом растеризации состоит из следующих этапов:

1) *Входная сборка*: на этом этапе происходит формирование вершинных и индексных буферов, размещённых в памяти ГП.

2) *Вершинный шейдер*: как и любой шейдер, представлен программным кодом на языке GLSL [35]. В этом шейдере выполняются геометрические преобразования для размещения всех вершин тел в едином пространстве мира и расчет анимации;

3) *Сборка примитивов и их отбраковка*: выполняется объединение вершин в треугольники и отбрасываются те части треугольников, которые не вносят вклад в итоговое изображение в заданном поле обзора;

4) *Растеризация*: двумерная проекция трехмерной сцены разбивается на заданную сетку *фрагментов*, которая соответствует пространственному разрешению фокальной плоскости моделируемого ФПУ. Затем для каждого фрагмента выполняется тест глубины, что позволяет оставить только видимые из них. Примитивы, скрытые более близкими примитивами к наблюдателю, отбрасываются;

5) *Фрагментный шейдер*: вызывается для каждого фрагмента и выполняет расчет их свойств. В случае моделирования ИК сцен, выполняется расчет числа излучаемых фотонов с учетом спектральных характеристик пропускания атмосферы, пропускания опти-

ческого тракта и свойств тел в заданном диапазоне длин волн. Моделирование видимого спектра выполняется наложением текстур материалов на фрагменты для получения реалистичного изображения.

Такая последовательность выполняемых процедур соответствует широко распространённому методу отрисовки, называемому *растеризацией,* который является самым быстрым методом получения картины сцены. В результате однократного прохода по последовательности графического конвейера мы получаем следующие выходные данные:

1) Набор двумерных картин распределения числа падающих фотонов на фокальную плоскость сенсора, причем точки каждой картины – *пиксели* – имеют формат чисел с плавающей запятой одинарной точности. Пиксели картин соответствует пикселям на ФПУ, а число этих картин определяется требуемой глубиной разбиения интересующего ИК-диапазона на малые участки длин волн.

2) Изображение сцены в видимом участке спектра, обеспечивающая возможность итеративного процесса разработки сцен в привычном виде. Каждый пиксель изображения хранит три цветовых компоненты в виде чисел с плавающей запятой одинарной точности, которые затем могут быть каким-либо образом преобразованы с целью вывода информации на пользовательский экран.

Следует отметить тот факт, что каждый этап графического конвейера при отрисовке методом растеризации проходится *единожды*. Повторный вызов функций произвольных этапов на каком-либо шаге конвейера недопустим и представляет значительные трудности в моделировании физических эффектов, имеющих высокую значимость в исследуемой области спектра.

Примером может служить механика непрямого освещения и затенения – когда первый объект подсвечивается отражённым светом от второго объекта или формируемое затенение между зависит от близости их взаимного расположения. В таком случае цветовые компоненты фрагмента первого объекта нельзя определить до того, как будет вычислен цвет второго, или требуется их одновременное изменение, что вынуждает делать множество графических подпроходов и использовать различные методы *отложенной отрисовки*, в какой-то мере качественно имитирующих вышеназванные механизмы.

Другими примерами трудновоспроизводимых эффектов можно назвать глубину резкости, каустику, преломление и отражение электромагнитных волн, геометрические и хроматические аберрации, неоднородные полупрозрачные среды, представленные облаком частиц или нагретым газом и так далее.

Переход в инфракрасную область спектра качественно изменяет получаемую информацию, ведь помимо отраженного от тел сигнала мы исследуем сигнал, источником которого являются сами тела. А поскольку каждое твердое тело, жидкость или газ с температурой выше абсолютного нуля является таким источником, то мы сталкиваемся с задачей моделирования *фоновой засветки*, в которую вышеназванные эффект вносят значительный вклад.

Ключом к решению задачи физически корректного моделирования является механизм *трассировки лучей*, в зарубежной литературе известный как *raytracing* [36]. Расчет распространения луча, представляющего электромагнитную волну с ограниченным числом взаимодействий со средой, позволяет корректно учитывать элементы теплового взаимодействия излучающих тел.

Трассировка лучей

Моделирование физических взаимодействий на визуализаторе, например, столкновения двух тел или переотражения света, в общем случае представлено в виде проверки пересечения двух объектов в пространстве. При описании взаимодействия излучения с объектами на сцене используется трассировка луча для оценки расстояния до ближайшего объекта [37]. Луч $\mathbf{r}(t)$ определяется точкой отсчета $\mathbf{o}$ и вектором направления $\mathbf{d}$, который, в общем случае, нормализован $||\mathbf{d}|| = 1$. Математическую форму луча можно представить как:

$$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d} \qquad (1)$$

где t – скалярная величина, задающая точку вдоль луча. Поскольку $\mathbf{d}$ нормализован, то t можно также называть расстоянием от точки отсчета в единицах измерения длины в системе. Обычно она используется в качестве численного значения расстояния до точки пересечения луча.

В поставленной задаче мы хотим обнаруживать ближайшее пересечение луча со сценой, все пересечения луча со сценой и пересечение луча с определённым условием. В современных программных интерфейсах доступа к графическим ресурсам, таких как DirectX, [38] OpenGL, [39] Vulkan [40] и других, трехмерные объекты на сцене обычно представлены в виде поверхностей, состоящих из большого набора *вершин*, каждые три соседних из которых формируют плоскость в виде *треугольника*.

Пусть нам требуется найти точку пересечения лучом тела на сцене и очевидный способ решения этой задачи является последовательный перебор всех треугольников поверхности тела на факт пересечения с ним. Для сцены, обычно состоящей приблизительно из 1 миллиона треугольников, проверка пересечения одного луча, испущенного из каждого пикселя области отрисовки (например, экрана с разрешением 1280×720), приведёт к необходимости выполнения почти триллиона операций определения столкновения луча с телом ради получения одного кадра. Очевидно, такой подход совершенно неприемлем для создания интерактивных виртуальных миров с малым временем отклика на воздействие пользователя.

Одим из способов преодоления такого ограничения предложен в виде описания трехмерной сцены как иерархии ограниченных объемов (*Bounding Volume Hierarchy, BVH*), [41] формальное описание которой называется *ускоряющими структурами.* Применение ограниченных объёмов позволяет представить произвольную сцену из множества объектов в виде иерархической структуры групп выпуклых многоугольников [42, 43]. Многие современные графические процессоры обладают аппаратной поддержкой ускоряющих структур, содержащих в своей основе иерархию ограниченных объёмов.

Многоугольник, ограничивающий собственным объёмом сцену, заключает в себе иерархию многоугольников, ограничивающих каждый объект и его составные части соответственно. Подобный подход в трассировке лучей позволяет значительно уменьшить вычислительную сложность алгоритма с $O(n)$ до $O(\log(n))$, где n – число полигонов модели тела.

Самым простым выпуклым многоугольником является параллелепипед, у которого нормали к граням совпадают с осями базиса мировых координат. Такой параллелепипед имеет название *ограничивающий параллелепипед, выровненный по координатным осям*, в литературе часто обозначается как *Axis-Aligned Bounding Box или AABB*. Математически, AABB задаётся двумя диагональными точками a_{min} и a_{max} , причем $a_{imin} \leq a_{imax}$, $\forall i \in \{x, y, z\}$. Существует множество различных методов поиска пересечения луча с ограниченным объёмом является, наиболее популярным среди которых является *метод плит (slabs method)*, а его математическое описание подробно представлено в [37, 41].

После определения целевого полигона вышеназванным методом необходимо определить конкретную точку пересечение луча с полигоном. Один из наиболее быстрых способов определения точки пересечения луча с треугольником предложен в [44]. Пусть луч (Формула 1) падает на треугольник, вершины которого представлены тремя точками $\mathbf{p_0}$, $\mathbf{p_1}$ и $\mathbf{p_2}$. Точка пересечения луча на площади треугольника, $\mathbf{f}(u,v)$ задается явно:

$$\mathbf{f}(u,v) = (1-u-v)\mathbf{p}_0 + u\mathbf{p}_1 + v\mathbf{p}_2 \qquad (2)$$

где u,v – две барицентрические координаты, удовлетворяющие условиям $u \geq 0$, $v \geq 0$ и $u + v \leq 1$ (рис. . ***2***). Таким образом, u и v являются величинами, по которым оценивается вклад каждой вершины в определенную координату точки на площади треугольника. Третья координата $w = 1 + u + v$. Поиск пересечения луча, $\mathbf{r}(t)$, и треугольника, $\mathbf{f}(u,v)$ эквивалентно:

$$\mathbf{o} + t\mathbf{d} = (1-u-v)\mathbf{p}_0 + u\mathbf{p}_1 + v\mathbf{p}_2 \qquad (3)$$

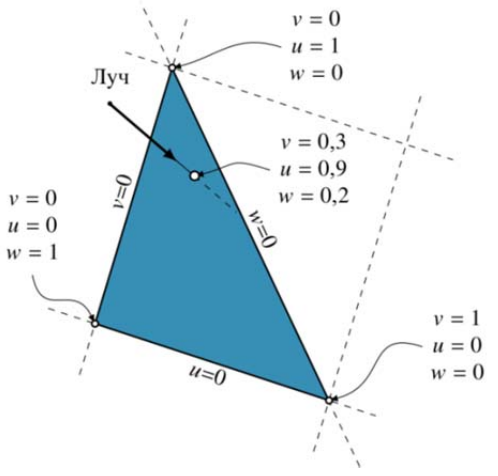

Рис. 2. Барицентрические координаты для треугольника, а так же точка пересечения с лучом. При этом u + v + w = 1

После перестановки членов получаем следующую СЛАУ, решение которой позволит определить барицентрические координаты точки пересечения (u,v) и расстояние до неё от точки испускания луча t:

$$\left(-dp_1 - p_0 p_2 - p_0\right)\begin{pmatrix} t \\ u \\ v \end{pmatrix} = o - p_0. \qquad (4)$$

Эта информация позволяет учесть свойства тела в данной точке его поверхности и выполнить корректный расчет взаимодействия с электромагнитной волной.

Трассировка пути в визуализаторе

Физически корректный вариант отрисовки видимой сцены с применением трассировки лучей обычно проводится методом *трассировки пути (path tracing)* [36, 45]. Поддержка трассировки пути и моделирования распространения лучей в разработанном визуализаторе требует внесение изменений в ряд процедур, которые отмечены на рис. 1 цветными блоками. В данной работе представлен процесс отрисовки примитивов сцены методом трассировки пути (см. рис. 1, зелёный блок) с применением аппаратного ускорения на основе интерфейса прикладного программирования *Vulkan*.

Формирование графического конвейера трассировки пути напоминает растеризацию – мы загружаем заранее подготовленную трехмерную сцену и получаем из неё массив вершин, индексов, материалов с их свойствами и изображениями текстур. Все эти ресурсы переносятся в выделенную локальную память на ГП, а полученные адреса в памяти данных ресурсов затем используются в шейдерных программах для расчетов.

На следующем шаге формируется иерархия ограниченных объемов. Для пользовательского доступа эта иерархия представлена из двух структур, *низко- (BLAS) и верхнеуровневыми (TLAS) ускоряющими структурами (bottom and top level acceleration structures)* [40].

BLAS это фактическое описание примитивов на основании которых будет строиться иерархия ограниченных объемов для быстрого доступа в графическом конвейере визуализатора. Такая структура содержит в себе информацию о том, как выглядят исходные примитивы – непрозрачные треугольники, каким форматом они описываются, как правильно интерпретировать их свойства и так далее. Подобное описание предназначено для каждого уникального тела на сцене, но может быть подготовлено одно для нескольких тел сразу.

TLAS является описанием иерархии объектов на сцене, заполняемая экземплярами BLAS, каждый из которых обладает уникальным набором матриц преобразования и другими свойствами. Это позволяет разделить ресурс описания объекта и допустимые над этим объектом преобразования, что крайне удобно для создания анимации или формирования сцен из большого числа объектов.

Далее описывается набор шейдеров, моделирующих различные виды взаимодействия луча со сценой и, поскольку мы не можем заранее предположить все варианты направлений распространения лучей, мы подготавливаем *таблицу связывания шейдеров*. Структура наподобие *vtable* для языка *C++* содержит в себе список идентификаторов шейдеров, таким образом, предоставляя возможность вызывать любую функцию на любом этапе отрисовки изображения. Диаграмма последовательности вызовов шейдеров представлена на рис. . ***3***.

Каждый шейдер трассировки пути описывает определенный вид взаимодействия луча с элементами сцены. *Шейдер формирования лучей*, вызывается для каждого пикселя фокальной плоскости сенсора, на которую будет падать итоговое излучение от тел на сцене. В данном шейдере реализован механизм обратного распространения луча [46]. Каждый кадр отрисовки изображения сцены данный шейдер определяет центр пикселя на фокальной плоскости сенсора и испускает через него луч, который ограниченное число раз взаимодействует с элементами окружающей среды. Максимальное число взаимодействий луча со сценой называется глубиной трассировки. В данном визуализаторе глубина является управляемым параметром с ограничением до 100 взаимодействий, а вклад каждого последующего шага обратно пропорционален его номеру.

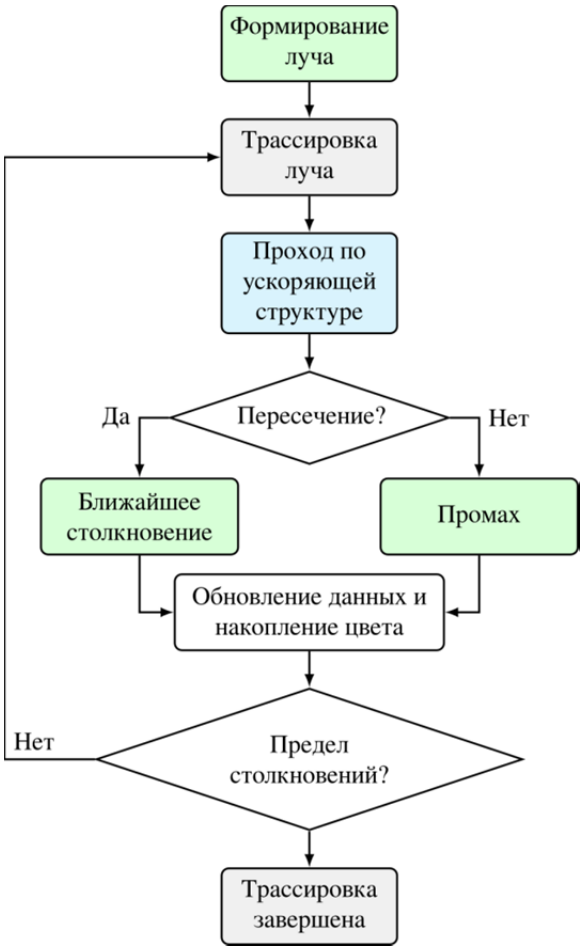

Рис. 3. Диаграмма активности цикла трассировки пути. Зелёным оттенком обозначены блоки, представленные шейдерами. После формирования исходного луча начинается процесс трассировки ускоряющей структуры на предмет столкновения с ограниченным объемом. Достижение предела столкновений луча завершает его трассировку

Разбиение изображения на ограниченное число фрагментов, из центров которых происходит генерация лучей, неизбежно приведёт к проблеме алиасинга изображения. На итоговой картине это проявляется как ступенчатость различных цветовых граней, идущих под небольшим углом к строкам или колонкам пикселей. Одни из методов ослабления алиасинга в случае трассировки лучей обеспечивается небольшим колебанием точки прохождения луча относительно заданного центра пикселя. Такое небольшое смещение приводит к аккумуляции результата расчета в данном фрагменте от нескольких близких друг к другу фрагментов, что обеспечивает плавные цветовые переходы на резких границах. В качестве источника случайной добавки к координате центра используется псевдослучайный генератор чисел с равномерным распределением на основе быстрого алгоритма *Tiny Encryption Algorithm*, TEA [47] (рис. . ***4***).

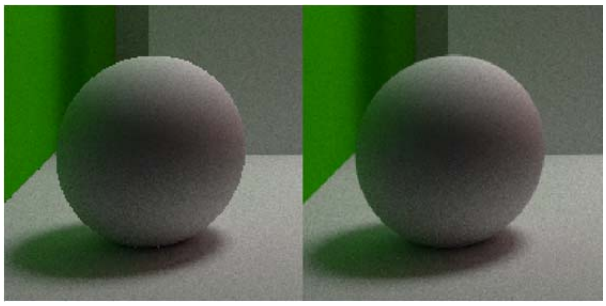

Рис. 4. Эффект алиасинга на изображении при отрисовке фигуры (слева), проявляющийся в резких ступенькообразных гранях сферы и результат применения метода компенсация алиасинга с помощью колебания точки прохождения луча (справа).
Наличие алиасинга на изображении обусловлено ограниченной плотностью разбиения трехмерной сцены примитивов на фрагменты, что в свою очередь определяет недостаточную пространственную частоту дискретизации сцены

При столкновении испущенного луча с поверхностью вызывается *шейдер ближайшего столкновения*. В этом шейдере выполняется расчет цвета поверхности в точке столкновения луча и формирование нового луча с уменьшением числа оставшихся взаимодействий на единицу. На текущий момент все материалы сцены считаются идеально диффузными, поэтому для них применяется закон Ламберта, определяющий изменение величины силы света от угла к нормали отражающей поверхности.

Таким образом, отраженный от поверхности луч приобретает новое направление на основе выбора случайной точки на полусфере с центром в точке столкновения с поверхностью. В качестве величины, передаваемой обратно в вызывающий шейдер, передаётся величина силы света материала (если материал поверхности светоизлучающий) и коэффициент поглощения, вносящий поправку в отраженный от поверхности свет.

Шейдер промаха обычно возвращает результат трассировки луча, если он не достиг никакой поверхности либо превысил максимально допустимую глубину трассировки. В данном случае все промахнувшиеся лучи возвращают предварительно заданный цвет и трассировка завершается.

Выходными данными описанного выше графического конвейера является цветное

изображение формата RGBA, где на каждый канал приходится 32 бита данных о цвете в виде числа с плавающей точкой. Это изображение содержит высокую степень шума, потому что один стохастически проложенный путь распространения луча не может учесть вклад всех источников излучения в сигнал на данном пикселе. В результате один кадр фотореалистичного изображения получается накоплением множества выборок – *семплов* и количество таких семплов может достигать несколько тысяч, если в поставленной задаче требуется достичь как можно меньшего уровня шума.

После накопления достаточного количества семплов для данного кадра он передаётся в модуль МЭО, где над ним производится нелинейная гамма-коррекция с целью представления изображения в корректном для интерпретации виде, а затем выполняется операция понижения разрядности изображения для вывода на экран.

Результаты

Для исследования разработанного трассировщика лучей была поставлена серия модельных экспериментов на компьютере под управлением ОС Debian 11, с процессором Intel Core i9 12900K и ГП Nvidia RTX 3080 12G. Интерфейс взаимодействия симулятора сцен с ГП представлен аппаратно-программным интерфейсом Vulkan версии 1.3.224, а поддержка ГП обеспечивается драйверами Nvidia версии 535.98.

Апробация трассировщика пути была проведена на основе двух сцен: модель человека на фоне кирпичной стены, ранее представленная в тестах различных моделей ФПУ, а также традиционная сцена для различных визуализаторов с применением трассировки пути – Корнеллская коробка [48].

Демонстрация трассировки пути

На рис. 5 представлены результаты поэтапного перехода от растеризатора к трассировщику пути с большой глубиной трассировки луча. Растеризатор в данном комплексе (рис. 5*а*) не предоставляет механизмов моделирования света и тени, поэтому наблюдаемое затенение тела на сцене может быть обеспечено только благодаря карте теней, добавленных к изображениям текстур, что можно увидеть, например, на кирпичной кладке.

Рис. 5. Изображения сцены с моделью человека на фоне стены в видимой части спектра: а) – изображение растеризатора; б) – трассировщик лучей с единичной глубиной трассировки; в) – трассировщик пути с одним источником освещения; г) – трассировщик пути с тремя источниками освещения. Для трассировщика пути глубина трассировки составила 10 шагов, а кадр накоплен из 2 тысяч семплов. Время отрисовки составило ~ 2 секунды

Примитивный способ моделирования светотени можно увидеть на рис. 5*б*. Данный вариант трассировки содержит единичный случай столкновения испущенного луча из фокальной плоскости сцены, поэтому является самым быстрым способом моделирования трассировки лучей. Затенение областей на лице человека обеспечено учётом ламбертовского отражения света, а тень на стене обеспечена проверкой доступности источником света указанной точки на поверхности – если источника света нет в прямой видимости, следует уменьшить яркость точки на заданную величину.

Переход к трассировке пути позволило получить изображение, представленное на рис. 5*в*. Трассировка пути обеспечивает наличие мягких теней на деталях лица и кирпичной стены. Многократное переотражение луча от стены можно заметить по небольшой освещённой области лица у левого виска и вдоль края левой стороны тела, куда свет не попадает явно.

На рис. 5*г*, представлена трассировка пути с несколькими источниками освещения, что позволило получить изображение картину близкую к естественному освещению сцену без необходимости последующей обработки.

Корнеллская коробка

Корнеллская коробка представляет собой классическую сцену для оценки качества и производительности трассировщиков лучей. Она состоит из кубического помещения, в котором левая и правая грани раскрашены в яркие зеленые и красные цвета, соответственно, а остальные грани имеют белый цвет. На верхней грани размещается источник освещения, на полу располагается ряд тел произвольной формы, каким-либо образом размещённых друг относительно друга. Исходный файл сцены был получен из открытых примеров разработки визуализаторов Nvidia [49].

Результат отрисовки Корнеллской коробки показан на рис. . ***6***. Наличие яркого источника освещения в совокупности с контрастными оттенками противоположеных стен позволяют продемонстрировать качественное преимущество метода трассировки пути по сравнению с подходом растеризации сцены. На представленном изображении можно отметить корректную обработку непрямого освещения и теней.

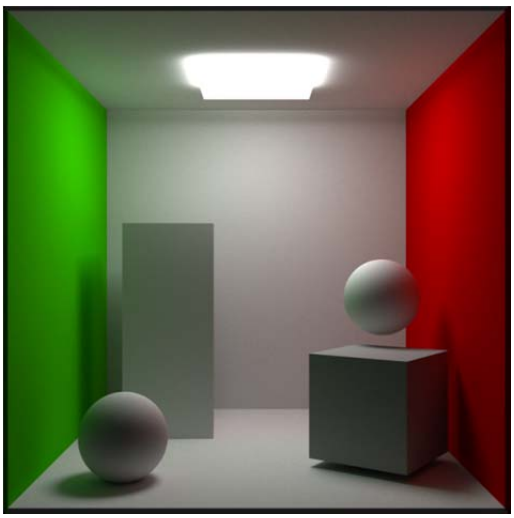

Рис. 6. Фрагмент изображения Корнеллской коробки, полученного с помощью трассировщика пути. Исходное изображение с разрешением 2560×1440 содержит 5000 семплов, глубина взаимодействий луча с окружением составляет 10 столкновений. Время отрисовки изображения с указанными параметрами составило 14 секунд

Оценка производительности

Трассировка лучей всегда идёт вместе с проблемой высокой нагрузки на вычислительные ресурсы. С целью исследования влияния параметров отрисовки изображения на скорость выполнения расчетов была проведена серия модельных экспериментов с применением сцены Корнеллской коробки. В качестве параметров использовались пространственное разрешение, определяющее возможности комплекса моделировать сцены с высокой детализацией окружения потребителю и глубина трассировки, определяющая максимальное число взаимодействий луча с моделируемой сценой.

На рис. . *7* представлен график зависимости времени отрисовки кадра от глубины трассировки для ряда пространственных разрешений с соотношением сторон 16:9. Минимальная глубина трассировки определяется минимальным числом взаимодействий луча с поверхностями для получения освещенности в указанной точке – каждый луч должен достигнуть источник света для получения отклика в заданной точке.

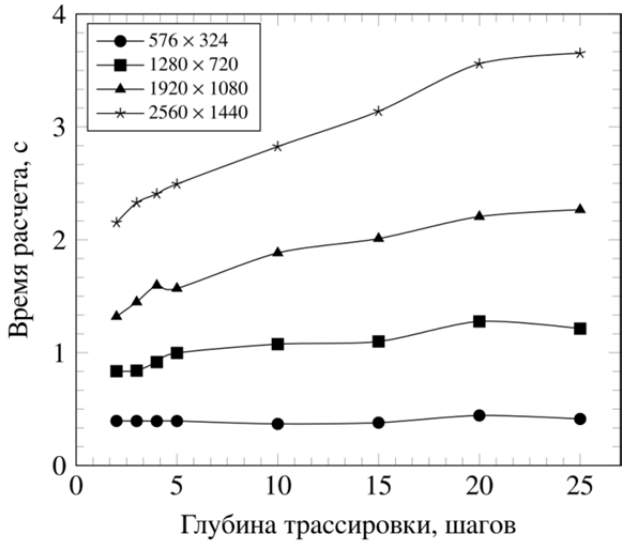

Рис. 7. Зависимость скорости отрисовки кадра из 1000 семплов от максимального числа взаимодействий испущенного луча со сценой. Каждая точка зависимости является средней арифметической величиной по выборке из 10 модельных экспериментов, а величина разброса результатов составляет менее 0,1 % от среднего значения

Каждый кадр изображения в данном эксперименте содержит 1000 семплов. Минимальное время трассировки одного кадра было

достигнуто при наименьшем размере изображения и составляет порядка 390 микросекунд на один семпл. Поскольку данная величина не зависит от глубины трассировки, то она характеризует минимальную задержку обработки кадра изображения всеми графическими конвейерами модулей. В случае большого пространственного разрешения изображения мы получаем практически линейную зависимость отрисовки кадра от числа шагов.

Чтобы определить достаточное число шагов на отрисовку одного кадра сцены для получения правдоподобного изображения был проведён ряд экспериментов, результаты которого представлены на рис. . ***8***. На данном примере можно отметить постепенное уточнение областей непрямого освещения при изменении глубины трассировки с 2 до 5 столкновений – к примеру, осветление области тени между ребром куба и красной стеной или нижней гранью куба и поверхностью пола, а также более выразительные красные и зелёные пятна отраженного света на объектах. При этом дальнейшее увеличения глубины трассировки до 10 и до 25 столкновений не вносит заметных качественных изменений в итоговую картину сцены.

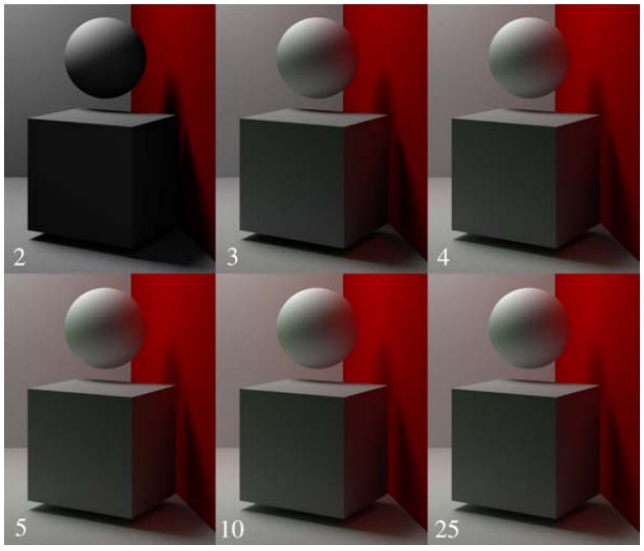

Рис. 8. Качество итогового изображения от максимальной глубины трассировки сцены. Каждое изображение получено накоплением 5000 семплов, а в левом нижнем углу фрагментов сцены указана разрешенная глубина трассировки. Важно отметить тот факт, что учет взаимодействий больше чем 5 раз не вносит заметного вклада в общую картину сцены, но при этом значительно увеличивает нагрузку на ГП в случае больших пространственных разрешений

Заключение

В данной работе была представлена реализация поддержки трассировщика пути в разработанный визуализатор трехмерных динамических сцен. Трассировщик пути основан на циклическом графическом конвейере с произвольным доступом к шейдерам и позволяет выполнять физически корректный расчет взаимодействия электромагнитной волны с телами, представленными на сцене в видимом диапазоне длин волн. Полученные в результате расчета сцен изображения обладают высокой степенью соответствия реально наблюдаемому окружению.

Поскольку исходный визуализатор симулятора трехмерных сцен был подготовлен на основе аппаратно-программного интерфейса *Vulkan,* то трассировщик пути получил аппаратное ускорение с применением вычислительных ядер ГП с поддержкой ускоряющих структур, что позволяет получить правдоподобные изображения различных сцен с несколькими источниками света с высокой частотой кадров.

В дальнейшей работе над визуализатором сцен планируется внедрение элементов геометрической оптики, полупрозрачных и зеркальных материалов, спектральных характеристик атмосферы и непосредственная модификация трассировщика пути для отрисовки изображений в инфракрасной области спектра, представляющую интерес для подготовки набора физически корректных исходных данных с высоким динамическим диапазоном величин.

ЛИТЕРАТУРА

1. *Hossein Motlagh N., Taleb T., Arouk O.* / IEEE Internet of Things Journal. 2016. Vol. 3. № 6. P. 899–922.

2. *Horgan J. et al.* / 2015 IEEE 18th International Conference on Intelligent Transportation Systems. 2015. P. 2032–2039.

3. *Loquercio A. et al.* / IEEE Robotics and Automation Letters. 2018. Vol. 3. № 2. P. 1088–1095.

4. *Yamashita R. et al.* / Insights into Imaging. 2018. Vol. 9. № 4. P. 611–629.

5. *Bojarski M. et al.* / CoRR. URL: http://arxiv.org/abs/1604.07316 (дата обращения: 20.08.2023).

6. *Mohammed A. S.* / Sensors. 2020. Vol. 20. № 22. P. 6532.

7. *Janai J. et al.* // CoRR. URL: http://arxiv.org/abs/1704.05519 (дата обращения: 20.08.2023).

8. *Posch C. et al.* / Proceedings of the IEEE. 2014. Vol. 102. № 10. P. 1470–1484.

9. *Chen D. G. et al.* / IEEE Transactions on Biomedical Circuits and Systems. 2011. Vol. 5. № 1. P. 64–82.

10. *Culurciello E., Etienne-Cummings R., Boahen K.* / IEEE Journal of Solid-State Circuits. 2003. Vol. 38. № 2. P. 281–294.

11. *Lichtsteiner P., Posch C., Delbruck T.* / IEEE Journal of Solid-State Circuits. 2008. Vol. 43. № 2. P. 566–576.

12. *Lichtsteiner P., Posch C., Delbruck T.* / 2006 IEEE Internati-onal Solid State Circuits Conference – Digest of Technical Papers. 2006. P. 2060–2069.

13. *Mueggler E., Huber B., Scaramuzza D.* / 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2014. P. 2761–2768.

14. *Camuñas-Mesa L. A. et al.* / IEEE Transactions on Neural Networks and Learning Systems. 2018. Vol. 29. № 9. P. 4223–4237.

15. *Posch C., Matolin D., Wohlgenannt R.* / IEEE Journal of Solid-State Circuits. 2011. Vol. 46. № 1. P. 259–275.

16. *Zang S. et al.* / IEEE Vehicular Technology Magazine. 2019. Vol. 14. № 2. P. 103–111.

17. *Vargas J. et al.* / Sensors (Basel). 2021. Vol. 21. № 16.

18. *Posch C. et al.* / IEEE Sensors Journal. 2009. Vol. 9. № 6. P. 654–664.

19. *Zhu H. et al.* / Nature Communications. 2021. Vol. 12.

20. *Zahidi U. A. et al.* / Remote Sensing. 2020. Vol. 12. № 1.

21. FASSP – Dr. John Kerekes. 2022. URL: https://www.cis.rit.edu/people/faculty/kerekes/fassp.html (дата обращения: 20.11.2022).

22. *Moorhead I. et al.* / Optical Engineering – OPT ENG. 2001. Vol. 40. P. 1896–1905.

23. *Goodenough A. A., Brown* / IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2017. Vol. 10. № 11. P. 4818–4833.

24. *Lefebvre S. et al.* / Proceedings of SPIE – The International Society for Optical Engineering. 2011. Vol. 8050.

25. *Willers C., Willers M., Lapierre F.* / Proc SPIE. 2011. Vol. 8187.

26. *Sundberg R. L. et al.* / Sensors, Systems, and Next-Generation Satellites VII. 2004. Vol. 5234. P. 252–261.

27. *Curran A. R., Gonda T. G.* Applications of the MuSES Infrared Signature Code. Тех. отч. DTIC ADA457152, 2005.

28. *Cota S. et al.* / Journal of Applied Remote Sensing. 2010. Vol. 4.

29. *Vaitekunas D. A., Lawrence O. E.* / SPIE. Targets and Backgrounds: Characterization and Representation. 1999. Vol. 3699. P. 92–102.

30. *Lapierre F., Archer C., Marc A.* New research results and validation on real data of the osmosis opensource infrared ship signature modeling software. 2009. URL:https://api.semanticscholar.org/CorpusID:17535855 (дата обращения: 20.11.2022).

31. *Servera J. et al.* / IEEE Transactions on Geoscience and Remote Sensing. 2021. Vol. 60.

32. *Зверев А. В., Ипатов Д. Е.* / Наноиндустрия. 2023. Т. 16. № 119. С. 234–242.

33. *Zverev A. V., Ipatov D. E.* / 2022 IEEE 23rd International Conference of Young Professionals in Electron Devices and Materials (EDM). 2022. P. 15–19.

34. glTF™ 2.0 Specification. 2022. URL: https://www.khronos.org/registry/glTF/specs/2.0/glTF-2.0.html (дата обращения: 20.04.2022).

35. *Kessenich J., Baldwin D., Rost R.* The OpenGL Shading Language. Version 4.60.7. 2019. URL: https://registry.khronos.org/OpenGL/specs/gl/GLSLangSpec.4.60.pdf (дата обращения: 10.08.2022).

36. *Kajiya J. T.* / SIGGRAPH Comput. Graph. 1986. Vol. 20. № 4. P. 143–150.

37. *Akenine-Möller T. et al.* Real-Time Rendering 4th Edition. – Boca Raton, FL, USA : A K Peters/CRC Press, 2018.

38. DirectX graphics and gaming. 2023. URL: https://docs.microsoft.com/en-us/windows/win32/directx (дата обращения: 20.04.2023).

39. OpenGL – The Industrys Foundation for High Performance Graphics. 2023. URL: https://www.opengl.org/ (дата обращения: 20.04.2023).

40. Vulkan – Industry Forged. 2023. URL: https://www.khronos.org/vulkan/ (дата обращения: 20.04.2023).

41. *Kay T. L., Kajiya J. T.* / SIGGRAPH Comput. Graph. 1986. Vol. 20. № 4. P. 269–278.

42. *Meister D. et al.* / Computer Graphics Forum. 2021. Vol. 40. № 2. P. 683–712.

43. NVIDIA Turing GPU Architecture: Graphics Reinvented: тех. отч. / NVIDIA Corporation. 2018. P. 86.

44. *Möller T., Trumbore B.* / ACM SIGGRAPH 2005 Courses, 2005.

45. *Veach E., Guibas L. J.* / ACM SIGGRAPH Press/Addison-Wesley Publishing Co., 1997. P. 65–76.

46. *Whitted T.* / Commun. ACM. 1980. Vol. 23. P. 343–349.

47. *Zafar F., Olano M., Curtis A.* / Proceedings of the Conference on High Performance Graphics. Saarbrucken, Germany : Eurographics Association, 2010. P. 133–141.

48. The Cornell Box – Cornell University Program of Computer Graphics. 2023. URL: http://www.graphics.cornell.edu/online/box/ (дата обращения: 10.08.2023).

49. NVIDIA DesignWorks Samples. 2023. URL: https://github.com/nvpro-samples (дата обращения: 10.08.2023).

PACS: 07.05.Tp

An environmental radiation interaction modeling in dynamic scene simulation software

A. V. Zverev and D. E. Ipatov

Motiv NT, LLC
Bd. 1 42, b-rd Bolshoy (ter. Skolkovo Innovationny Centra), Moscow, 121205, Russia
E-mail: dipatov@motivnt.ru

Received 6.09.2023; revised 21.09.2023; accepted 27.09.2023

A path tracing method support to the existing dynamic 3D scenes simulator has been developed. This method allows obtaining physically correct images taking into account multiple interactions of electromagnetic waves with bodies. The results of several scenes calculations and a performance assessment of hardware-accelerated path tracing in the visible wavelength range are presented.

Keywords: modelling, photodetector, 3D, scene, ray tracing, path tracing, GPU, Vulkan.

DOI: 10.51368/2307-4469-2023-11-5-433-445

REFERENCES

1. Hossein Motlagh N., Taleb T. and Arouk O., IEEE Internet of Things Journal **3** (6), 899–922 (2016).
2. Horgan J. et al., 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pp. 2032–2039 (2015).
3. Loquercio A. et al., IEEE Robotics and Automation Letters **3** (2), 1088–1095 (2018).
4. Yamashita R. et al., Insights into Imaging **9** (4), 611–629 (2018).
5. Bojarski M. et al., CoRR **abs/1604.07316** (http://arxiv.org/abs/1604.07316) (2016).
6. Mohammed A. S., Sensors **20** (22), 6532 (2020).
7. Janai J. et al., CoRR **abs/1704.05519** (http://arxiv.org/abs/1704.05519) (2017).
8. Posch C. et al., Proceedings of the IEEE **102** (10), 1470–1484 (2014).
9. Chen D. G. et al., IEEE Transactions on Biomedical Circuits and Systems **5** (1), 64–82 (2011).
10. Culurciello E., Etienne-Cummings R. and Boahen K., IEEE Journal of Solid-State Circuits **38** (2), 281–294 (2003).
11. Lichtsteiner P., Posch C. and Delbruck T., IEEE Journal of Solid-State Circuits **43** (2), 566–576 (2008).
12. Lichtsteiner P., Posch C. and Delbruck T., 2006 IEEE International Solid State Circuits Conference – Digest of Technical Papers. 2006, pp. 2060–2069.
13. Mueggler E., Huber B. and Scaramuzza D., 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp. 2761–2768.
14. Camuñas-Mesa L. A. et al., IEEE Transactions on Neural Networks and Learning Systems **29** (9), 4223–4237 (2018).
15. Posch C., Matolin D. and Wohlgenannt R., IEEE Journal of Solid-State Circuits **46** (1), 259–275 (2011).
16. Zang S. et al., IEEE Vehicular Technology Magazine **14** (2), 103–111 (2019).
17. Vargas J. et al., Sensors (Basel) **21** (16), (2021).
18. Posch C. et al., IEEE Sensors Journal **9** (6), 654–664 (2009).
19. Zhu H. et al., Nature Communications **12** (2021).
20. Zahidi U. A. et al., Remote Sensing **12** (1), (2020).
21. FASSP – Dr. John Kerekes. (https://www.cis.rit.edu/people/faculty/kerekes/fassp.html) (2022).
22. Moorhead I. et al., Optical Engineering – OPT ENG **40**, 1896–1905 (2001).
23. Goodenough A. A., Brown, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **10** (11), 4818–4833 (2017).
24. Lefebvre S. et al., Proceedings of SPIE – The International Society for Optical Engineering **8050** (2011).
25. Willers C., Willers M. and Lapierre F., Proc SPIE **8187** (2011).
26. Sundberg R. L. et al., Sensors, Systems, and Next-Generation Satellites VII. **5234**, 252–261 (2004).
27. Curran A. R. and Gonda T. G. Applications of the MuSES Infrared Signature Code, Tech Report DTIC ADA457152 (2005).

28. Cota S. et al., Journal of Applied Remote Sensing **4** (2010).

29. Vaitekunas D. A. and Lawrence O. E., SPIE, Targets and Backgrounds: Characterization and Representation **3699**, 92–102 (1999).

30. Lapierre F., Archer C. and Marc A., New research results and validation on real data of the osmosis opensource infrared ship signature modeling software, (https://api.semanticscholar.org/CorpusID:17535855) (2009).

31. Servera J. et al., IEEE Transactions on Geoscience and Remote Sensing **60** (2021).

32. Zverev A. V. and Ipatov D. E., Nanoindustry **16** (119), 234–242 (2023) [in Russian].

33. Zverev A. V. and Ipatov D. E., 2022 IEEE 23rd International Conference of Young Professionals in Electron Devices and Materials (EDM), 2022, pp. 15–19.

34. glTF™ 2.0 Specification, (https://www.khronos.org/registry/glTF/specs/2.0/glTF-2.0.html) (2022).

35. Kessenich J., Baldwin D. and Rost R. The OpenGL Shading Language, Version 4.60.7 (https://registry.khronos.org/OpenGL/specs/gl/GLSLangSpec.4.60.pdf) (2019).

36. Kajiya J. T., SIGGRAPH Comput. Graph. **20** (4), 143–150 (1986).

37. Akenine-Möller T. et al., Real-Time Rendering 4th Edition, Boca Raton, FL, USA : AK Peters/CRC Press, 2018.

38. DirectX graphics and gaming, (https://docs.microsoft.com/en-us/windows/win32/directx) (2023).

39. OpenGL – The Industrys Foundation for High Performance Graphics. (https://www.opengl.org/) (2023).

40. Vulkan – Industry Forged. (https://www.khronos.org/vulkan/) (2023).

41. Kay T. L. and Kajiya J. T., SIGGRAPH Comput. Graph **20** (4), 269–278 (1986).

42. Meister D. et al., Computer Graphics Forum **40** (2), 683–712 (2021).

43. NVIDIA Turing GPU Architecture: Graphics Reinvented : Tech Report, NVIDIA Corporation, 2018, p. 86.

44. Möller T. and Trumbore B., ACM SIGGRAPH 2005 Courses (2005).

45. Veach E. and Guibas L. J., ACM SIGGRAPH Press/Addison-Wesley Publishing Co., 1997, pp. 65–76.

46. Whitted T., Commun. ACM **23**, 343–349 (1980).

47. Zafar F., Olano M. and Curtis A., Proceedings of the Conference on High Performance Graphics. Saarbrucken, Germany : Eurographics Association, 2010, pp. 133–141.

48. The Cornell Box – Cornell University Program of Computer Graphics. (http://www.graphics.cornell.edu/online/box/) (2023).

49. NVIDIA DesignWorks Samples. (https://github.com/nvpro-samples) (2023).