

Объемно-ориентированная визуализация каустик и динамических преломляющих объектов со сложными свойствами материала в реальном времени

С. И. Вяткин, Б. С. Долговесов

Институт автоматики и электрометрии СО РАН, Новосибирск, Россия

Аннотация. Преломление и рассеяние света при прохождении через различные материалы остается сложной задачей, особенно когда целью является рендеринг в реальном времени динамичных сцен, в которых освещение, материалы и геометрия могут меняться. При визуализации сцен с глобальным освещением важно точно рассчитать освещение, чтобы получить детальные каустические характеристики. И в то же время достаточно быстро, чтобы можно было изменять освещение, геометрию и материалы в режиме реального времени. Наиболее интересные явления зависят от непрямого освещения, поэтому необходимо решить глобальную проблему освещения. Предлагается объемно-ориентированный подход к визуализации каустик и динамических преломляющих объектов со сложными свойствами материала в реальном масштабе времени. В конвейере рендеринга преобразуются поверхности в объемные данные, с отслеживанием кривых траекторий фотонов, преломляющихся в объеме, с отображением произвольных видов результирующего распределения яркости. В результате предложенный метод позволяет визуализировать каустики, поглощение и рассеяние света при выполнении физических симуляций и при манипулировании поверхностями в режиме реального времени.

Ключевые слова: объемно-ориентированная визуализация, вокселизация, вокセル, воксельное восьмеричное дерево, глобальное освещение, преломление, каустика

Volume-oriented visualization of caustics and dynamic refractive objects with complex material properties in real time

S.I. Vyatkin, B.S. Dolgovesov

Institute of Automation and Electrometry, Siberian Branch of the Russian Academy of Sciences,
Novosibirsk, Russia

Abstract. Refraction and scattering of light as it passes through different materials remains a challenging task, especially when the goal is to render dynamic scenes in real time, in which lighting, materials, and geometry can change. When visualizing scenes with global illumination, it is important to calculate accurately the lighting in order to obtain detailed caustic characteristics. In addition, at the same time, it is fast enough to be able to change lighting, geometry, and materials in real time. The most interesting phenomena depend on indirect lighting, so it is necessary to solve the global lighting problem. A volume-oriented approach to visualization of caustics and dynamic refractive objects with complex material properties in real time is proposed. In the rendering pipeline, surfaces are transformed into volumetric data, tracking the curves of the trajectories of photons refracted in the volume, displaying arbitrary views of the resulting brightness distribution. As a result, the proposed method makes it possible to visualize caustics, absorption and scattering of light when performing physical simulations and manipulating surfaces in real time.

Keywords: volume-oriented visualization, voxelization, voxel, voxel octal tree, global illumination, refraction, caustics

Введение

Реалистичные эффекты освещения важны для восприятия сложных пространственных структур [1-4]. Однако они также значительно и более затратные в вычислительном отношении, чем обычные модели локального освещения. Это еще более актуально для рендеринга объемных данных с улучшенными эффектами освещения ввиду отсутствия дискретных однородных объектов, так как вклад освещенности должен оцениваться и распространяться в каждой точке пространства. Одним из аспектов, которым часто пренебрегают, является преломление света. Эффекты преломления обычно игнорируются, и все лучи рассматриваются как прямые. Из-за высоких вычислительных затрат на физический рендеринг интерактивные методы обычно используют упрощенные оптические модели. Физически корректная визуализация скалярных полей выполняется с помощью объемных фотонных карт [5]. Однако этот метод не достигает даже интерактивной скорости вычислений. Большинство работ по фотонным картам посвящены рендерингу, при котором каждый пиксель в сцене вычисляется заранее перед отображением конечного результата, где качество и точность результата ценятся выше интерактивности. Однако в [6] описывается отображение фотонов, реализованное на графическом процессоре. Методы преломления

света в пространстве изображения реалистично отображают рефракцию, но многие из этих методов не отображают каустику [7, 8], или могут обрабатывать поверхностную каустику, но не объемную каустику [9]. Кроме того, в этих методах любое изменение освещения, материалов или геометрии сцены требует повторного расчета распределения яркости, что занимает несколько секунд. В работе [10] представлен физически корректный рендеринг в реальном времени эффектов однократного рассеяния, но без отображения преломления. Метод [11] позволяет отображать однократное рассеяние и преломление, однако он не учитывает поглощение. Кроме того, как и все методы обработки в пространстве изображения, он не учитывает преломления за пределами поля зрения, что делает каустику зависимой от обзора. В статье [12] представлена основанная на экране техника рендеринга объемных каустик, подход обеспечивает высокую производительность для сцен, в которых каустики влияют на небольшое количество пикселей. Этот подход также поддерживает неоднородные среды, но требует очень низкой частоты дискретизации для интерактивности. Что касается объемной визуализации, то в [13] был предложен фильтр на основе сплайнов для высококачественной реконструкции градиентов, с целью улучшения внешнего вида эффектов преломления. В [14] описано восьмеричное дерево для ускорения вычислений. Но эти методы также работают с отдельными выборками на поверхностях. В нашей работе [15] представлены как преломления, так и объемные каустики, которые визуализируются в интерактивном режиме. Используется прямой рендеринг функционально заданных объектов.

В данной работе мы представляем метод, позволяющий визуализировать эффекты, связанные с преломлением, поглощением и анизотропным одиночным рассеянием в динамических сценах в реальном масштабе времени. Предлагается конвейер рендеринга, который состоит из нескольких этапов, каждый из которых реализован на графическом процессоре. На вход конвейера поступает поверхностное описание сцены, которое вокселизируется, а также набор точечных и направленных источников света, а на выходе получаем физически корректное визуализированное изображение.

Постановка задачи

Поставлена задача разработки конвейера рендеринга таким образом, чтобы он начался с вокселизации "на лету", чтобы можно было поддерживать объемное представление даже при изменении входной геометрии. Поскольку воксельное представление позволит в полной мере использовать параллелизм графического процессора на каждом из последующих этапов конвейера, необходимо учесть непрерывное изменение показателя преломления по всему объему. Для этого фотоны и лучи просмотра должны следовать по изогнутым траекториям, определяемым уравнением луча геометрической оптики. Благодаря воксельному представлению необходимо построить восьмеричное дерево для ускорения алгоритма. То есть восьмеричное дерево нужно использовать для выбора переменного размера шага при распространении фотонов по их криволинейным траекториям. Реализация трассировки фотонов на базе графического процессора должна делить работу на этап перемещения фотонов, который вычисляет новое положение, направление и излучение каждого фотона после одного шага, и этап хранения фотонов, который накапливает излучение в объеме. При рендеринге преломляющих материалов необходимо иметь возможность изменять не только параметры просмотра в реальном времени, но и изменять параметры освещения и материала, сохраняя при этом скорость обновлений. Так, например, пользователи системы должны иметь возможность свободно изменять расположение и цвета источников света при просмотре эффектов каустики, поглощения, одиночного рассеяния и теней. Также конвейер должен поддерживать динамическое обновление свойств материала, а также освещение. Необходимо предусмотреть интеграцию в систему алгоритмов моделирования поверхностей, сохраняя при этом эффекты преломления. Например, пользователь должен иметь возможность в реальном времени деформировать преломляющий объект и наблюдать за влиянием деформации на освещение сцены.

Целью представленной работы является создание объемно-ориентированного метода визуализации каустик и динамических преломляющих объектов со сложными свойствами материала в реальном масштабе времени.

Теория

1. Функциональные объекты

Для компактного хранения базы данных используются функциональные объекты [16]. Функциональные объекты описываются с помощью квадрик и функций отклонения от квадрик:

$$q'(x, y, z) = q(x, y, z) + \sum_{i=1}^N d_i(x, y, z), \quad (1)$$

где $q'(x, y, z)$ – функциональная поверхность; $q(x, y, z)$ – квадрик; $i=1\dots N$ – количество функций отклонения; $d_i(x, y, z)$ – функция отклонения,

$$d_i(x, y, z) = \begin{cases} q_i^3(x, y, z), & \text{if } q_i(x, y, z) \geq 0 \\ 0, & \text{if } q_i(x, y, z) < 0 \end{cases} \quad (2)$$

где $q_i(x, y, z)$ – возмущающая функция второго порядка.

Для формирования моделей сложных объектов используются теоретико-множественные операции. Перед визуализацией в режиме реального времени функциональные объекты триангулируются [17]. Далее алгоритм работает в интерактивном режиме.

2. Вокселизация 3D-объектов

С целью плавного изменения значений показателя преломления в объеме необходимо при вокселизации присваивать дробные значения покрытия тем вокселям, через которые проходит поверхность. Для этого сначала выполняется дискретизация объема с субвоксельным разрешением. Это означает, что преобразуется сетка треугольников в объемную текстуру, которая в каждом измерении в четыре раза больше, чем на выходе. Однако при уменьшении выборки результирующей текстуры стоимость считывания выборок текстуры для каждого из вокселов на выходе является очень высокой. Поэтому используется понижающая дискретизация только для вокселов вблизи поверхности. На рисунке 1 показаны основные этапы вычисления вокселизации.

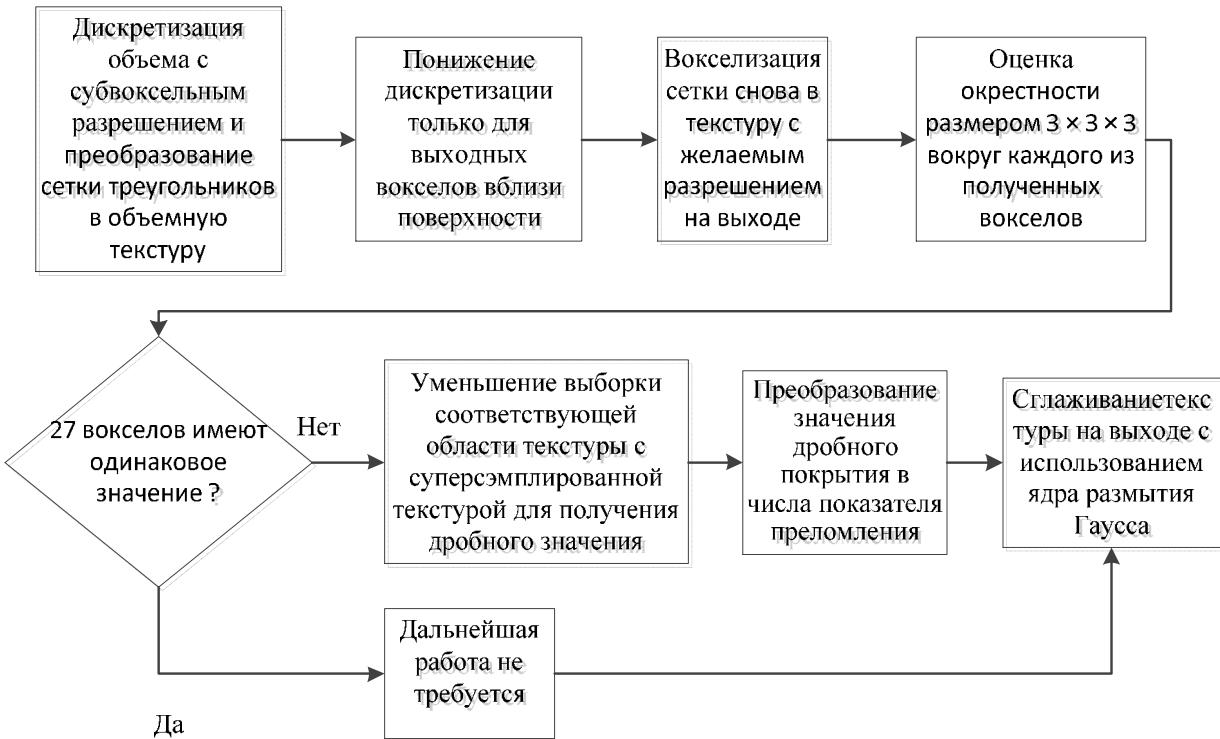


Рисунок 1. Структура вычислительного процесса вокселизации объекта

Вокселизируется сетка треугольников снова в текстуру с желаемым разрешением на выходе, затем проверяется окрестность размером $3 \times 3 \times 3$ вокруг каждого из полученных вокселов. Если значения 27 вокселов отличаются в этой окрестности, то уменьшаем выборку соответствующей области

текстуры с суперсэмплированной текстурой для получения дробного значения. Далее преобразуются значения дробного покрытия в числа показателя преломления и сглаживается текстура на выходе с использованием ядра размытия Гаусса. Используется супервыборка $4 \times 4 \times 4$ и ядро $9 \times 9 \times 9$. Суперсэмплирование эффективно повышает точность нормалей к поверхности, представленных градиентом показателя преломления, в то время как размытие размывает граничную область на более широкий диапазон вокселов.

Были определены разрешение сверхдискретизации и ядро размытия по Гауссу на основе экспериментальных изображений. Было отмечено, что изображения улучшаются по мере увеличения суперсэмплирования, независимо от того, какое ядро размытия используется. Как только было выбрано суперсэмплирование $4 \times 4 \times 4$ (самое большое, которое сочли целесообразным), увидели небольшое улучшение, когда размер ядра blur увеличился с 5 до 7–9. Однако существенных изменений не произошло, когда размер ядра увеличился до 11. Поэтому были использованы суперсэмплинг $4 \times 4 \times 4$ и ядро $9 \times 9 \times 9$.

Далее строится восьмеричное дерево (рис. 2). Построение представления в виде восьмеричного дерева аналогично построению mip-map пирамиды двумерного аналога. Пирамидастоится из трехмерных массивов, в которых записаны минимальные и максимальные показатели преломления, присутствующие в каждой объемной области. Затем анализируется пирамида от самого грубого уровня до самого точного, записывая номер текущего уровня в восьмеричное дерево всякий раз, когда диапазон значений показателя преломления находится в пределах входного допуска.

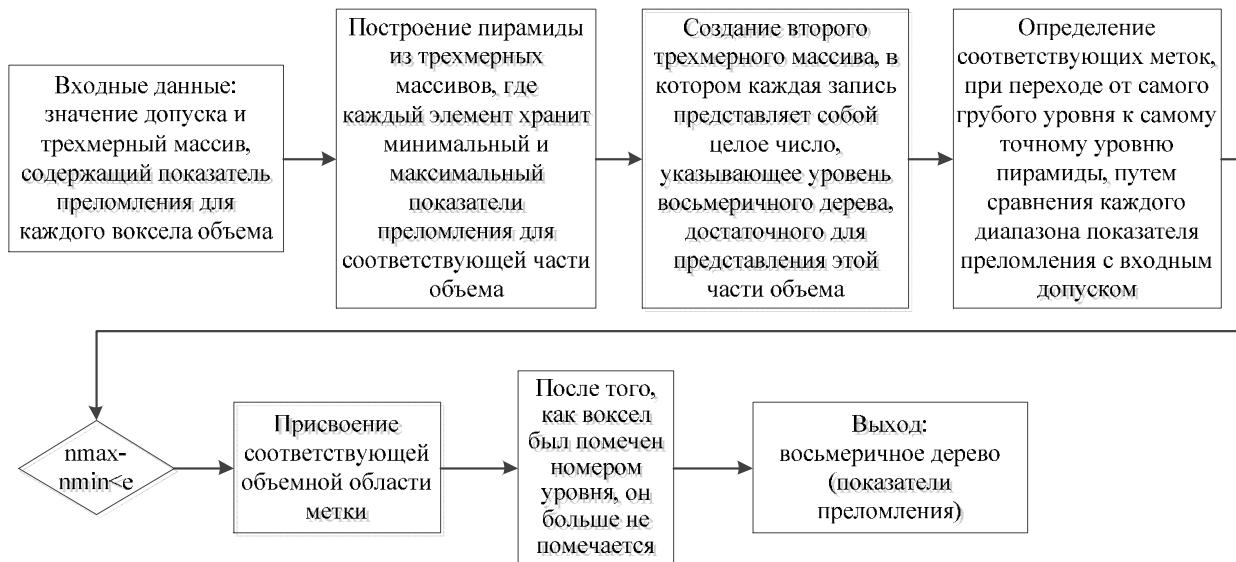


Рисунок 2. Алгоритм построения восьмеричного дерева

На вход (см. рис. 2) поступает кубический объем элементов показателя преломления n размером 2^M в каждом измерении. Начинаем с построения пирамиды из M трехмерных массивов, где каждый элемент хранит минимальный и максимальный показатели преломления n_{\min} и n_{\max} для соответствующей части объема. Далее создается второй трехмерный массив размером 2^M в каждом измерении, где каждая запись представляет собой целое число, указывающее уровень восьмеричного дерева, достаточный для представления этой части объема. Используется ноль в качестве метки для самого точного уровня дерева и m в качестве метки для самого грубого (одновоксельного) уровня. Затем определяем соответствующие метки, переходя от самого грубого уровня к самому точному уровню пирамиды, сравнивая каждый диапазон показателя преломления с входным допуском. При просмотре уровня пирамиды m , если встретится запись, удовлетворяющая условию $n_{\max} - n_{\min} < \epsilon$, происходит присваивание соответствующей объемной области метки m . После того как воксель был помечен номером уровня, он больше не помечается. Построение восьмеричного дерева происходит таким образом, чтобы оба показателя преломления n и коэффициент затухания β были

приблизительно постоянны для каждого конечного узла, поскольку на алгоритм отслеживания фотонов влияют оба этих свойства материала.

3. Алгоритм отслеживания фотонов

Сначала настраиваем графический конвейер таким образом, чтобы положение камеры совпадало с источником света и было ориентировано на интересующий объем. Затем визуализируем грани куба, ограничивающие объем. Далее отображаем текстуру с использование шейдера, который записывает альфа-значение равное единице вместе с позицией в 3D для каждого пикселя, представляющего обращенную вперед поверхность. Пустые пиксели имеют нулевые альфа-значения. Затем преобразуем текстуру в список точечных примитивов, представляющих фотоны, используя геометрический шейдер. Каждый пиксель с ненулевым значением альфа генерирует один фотон, положение фотона определяется по значению пикселя. Направление фотона определяется по координатам текстуры и положению источника света. Яркость фотона определяется характеристиками излучения света.

Уравнение луча геометрической оптики [18]

$$\frac{d}{ds} \left(n \frac{d\vec{x}}{ds} \right) = \nabla n, \quad (3)$$

где $\vec{x}(s)$ – криволинейный световой путь, n – показатель преломления.

Запишем уравнение (3) в виде системы дифференциальных уравнений первого порядка:

$$\frac{d\vec{x}}{ds} = \frac{\vec{v}}{n}, \quad (4)$$

$$\frac{d\vec{v}}{ds} = \nabla n, \quad (5)$$

где $\vec{v} = n \frac{d\vec{x}}{ds}$.

Кусочно-линейные аппроксимации кривых выглядят следующим образом:

$$\vec{x}_{i+1} = \vec{x}_i + \frac{\Delta s}{n} \vec{v}_i, \quad (6)$$

$$\vec{v}_{i+1} = \vec{v}_i + \Delta s \nabla n, \quad (7)$$

где Δs – размер шага.

По мере движения фотонов вдоль этих кривых меняется размер шага Δs , для этого используется структура данных восьмеричного дерева. Входные данные состоят из значений показателя преломления восьмеричного дерева, трехмерного массива коэффициентов ослабления RGB и списка фотонов. Каждый фотон имеет начальное положение \vec{x}_0 , направление \vec{v}_0 и значение яркости RGB \tilde{L}_0 .

Результатом является трехмерный массив распределений яркости RGB, описывающий освещенность, которая поступает в каждый воксель. Каждая итерация отслеживания фотонов продвигает фотоны на один шаг вперед в соответствии с уравнениями 6 и 7. Для каждого фотона используется восьмеричное дерево для определения наибольшего размера шага Δs_{octree} , чтобы удерживать фотон в области с приблизительно постоянным показателем преломления. То есть положение фотона используется для определения уровня окружающего его конечного узла восьмеричного дерева. А размер шага Δs_{octree} вычисляется для движения фотона к границе узла восьмеричного дерева. Затем выбираем Δs , который будет больше Δs_{octree} , и минимальный размер шага Δs_{min} , указанный пользователем. Минимальный размер шага обычно равен ширине одного или двух вокселов и может регулироваться пользователем в зависимости от точности и производительности.

В каждом вокселе аппроксимируется распределение входящего излучения с сохранением только средневзвешенного значения направления приходящих фотонов и значения излучения для RGB-длин волн. При прохождении фотоном пути от \vec{x}_i до \vec{x}_{i+1} должно происходить излучение в каждом из вокселов, через которые он проходит. Поэтому, когда движется фотон на один шаг вперед,

генерируются две вершины в буфере вершин для записи старого и нового значений положения, направления и яркости фотона. После продвижения всех фотонов на один шаг вперед обрабатывается буфер вершин в виде списка линейных сегментов. Сегменты растираются в выходной массив распределения яркости. В графическом конвейере используется интерполяция положения, направления и яркости фотона между двумя конечными точками каждого сегмента линии. Применяется пиксельный шейдер, в котором добавляется излучение фотона к распределению, сохраненному в каждом вокселе. Взвешивается направление движения фотона по сумме значений его RGB-излучения, прежде чем добавляется к направлению, сохраненному в каждом вокселе. После выделения всех фотонов и сохранения их вклада в распределение излучения исключаем каждый фотон, объем или излучение которого упало ниже порогового значения, и затем повторяется весь процесс. Продолжаются эти итерации до тех пор, пока количество активных фотонов не составит лишь малую долю от первоначального количества фотонов. Далее происходит сглаживание объемного распределения излучения с использованием приближения гауссова ядра $3 \times 3 \times 3$ для уменьшения шума. Алгоритм отслеживания фотонов показан на рисунке 3.

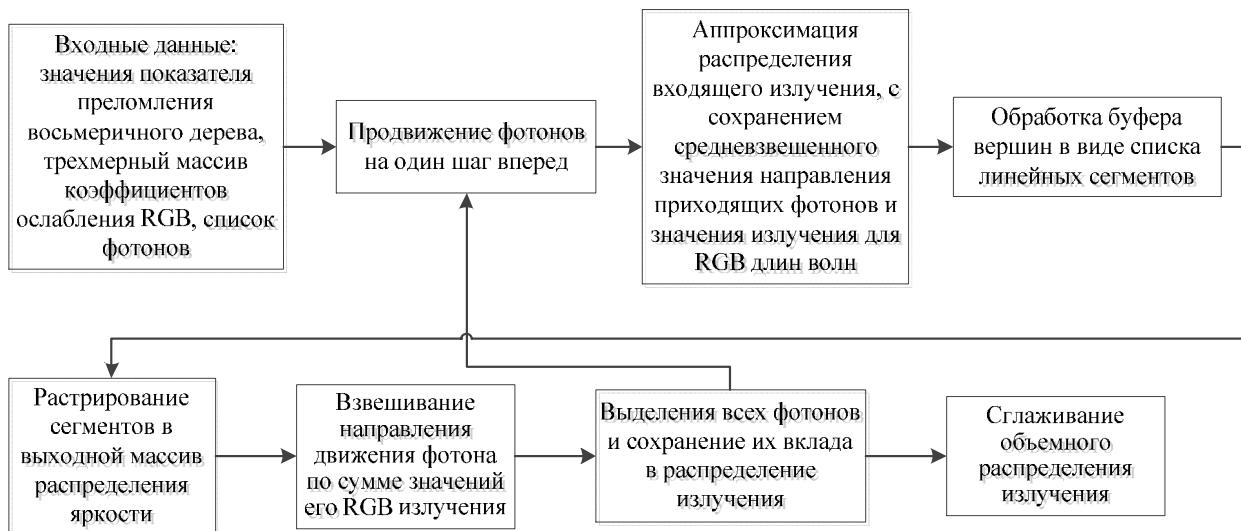


Рисунок 3. Алгоритм отслеживания фотонов

Отметим, что внешний вид каустик чувствителен к размеру шага перемещения фотонов. Размер шага, в свою очередь, зависит от допуска, используемого в алгоритме построения октодерева. Для некоторых объемных данных сложно выбрать допуск, который обеспечивает точную визуализацию с разумным количеством шагов перемещения фотонов. В частности, иногда приемлемое качество рендеринга требует такого низкого допуска, что размеры шагов слишком малы для достижения частоты кадров в реальном времени. Небольшое увеличение допуска может привести к объединению множества узлов октодерева, что приводит к значительно большему размеру шага и значительному снижению качества рендеринга. Поэтому создаем узлы октодерева с промежуточными размерами шага. В каждом вокселе хранится максимальный размер шага в дополнение к номеру уровня окружающего узла восьмеричного дерева конечного уровня. Выбирается размер шага между заданным пользователем минимальным размером шага и максимальным размером шага узла восьмеричного дерева. Выбираем максимальный размер шага для каждого вокселя на этапе построения октодерева. Когда вокселю присваивается номер уровня октодерева, если изменение показателя преломления меньше чем $\epsilon = 0,005$, устанавливается ограничение на максимальный размер шага вокселя равное бесконечности. Если изменения показателя преломления больше чем $\epsilon_1 = 0,005$, но меньше второго заданного пользователем допуска $\epsilon_2 = 0,02$, то устанавливается значение Δs_{\max} вокселя на конечный размер шага, выбранный пользователем. Эта схема гарантирует, что в узлах с практически постоянным показателем преломления продвигаются фотоны до самой границы узла, в то время как в узлах с некоторым изменением показателя преломления ограничивается размер шага.

4. Визуализация

Отслеживание лучей происходит в обратном направлении траектории, по которым световые лучи достигают камеры. Процесс начинается с инициализации начала и направления отдельного луча для каждого пикселя выходного изображения. Если луч пересекает треугольную сетку (после триангуляции функционально заданного объекта), то происходит шаг за шагом продвижение вдоль луча, пока он не выйдет из объема. Используются уравнения (6) и (7) для определения траекторий, по которым проходят лучи, точно так же, как это делается для фотонов. Однако при трассировке лучей обзора используется фиксированный размер шага, равный ширине одного вокселя, а не переменный размер шага. На каждом шаге по лучу обзора просматриваются значение и направление излучения, сохраненные в соответствующем вокселе. Затем оценивается функция фазы рассеяния для определения части излучения, рассеиваемой в направлении камеры. Для этого умножаем полученный результат на локальный коэффициент рассеяния и общее значение затухания вдоль луча из-за поглощения и рассеяния. Это произведение дает вклад в яркость одного вокселя, которое затем добавляется к общей яркости текущего луча. Как только луч выходит за пределы объема, включается фон для завершения вычисления цвета выходного пикселя. На рисунке 4 показана общая схема конвейера вычислений.



Рисунок 4. Общая схема конвейера

Результаты тестирования

Тестирование производилось на процессорах Intel Core i7 3820 и GPU NVIDIA GeForce GTX 1080 Ti. Для тестирования использовалось несколько сцен, пять из них показаны на рисунках 5–9. Разрешение изображения для всех примеров рендеринга составляет 1024 x 768. Все сцены полностью динамичные.

Можно отметить, что по мере того, как показатель преломления объекта увеличивается по сравнению с единицей, волновые фронты пересекаются раньше, и артефакты распространения становятся все более заметными. С другой стороны, отслеживание фотонов дает приемлемые результаты независимо от показателя преломления. Мы можем повторно высвечивать сцены, подобные изображенному на рисунке 5, примерно со скоростью восемьдесят кадров в секунду. На рисунке 6 показан эффект, как изменение освещения влияет на визуализацию преломляющего объекта, при этом видны каустические изменения. Пользователь в интерактивном режиме может деформировать преломляющий объект и наблюдать за влиянием деформации на освещение сцены (рис. 7 и 8). Была применена динамическая структура для обработки больших деформаций модели, в частности для обнаружения самопересечения для явного контроля топологии, создания гладких переходов между двумя объединенными частями поверхности и т.д.

Сочетание вокселизации и рендеринга требует использования современных графических процессоров. Центральный процессор используется для выполнения деформации. В результате можно интегрировать в систему множество алгоритмов моделирования поверхностей, сохраняя при этом эффекты преломления. Метод позволяет генерировать сцены, содержащие неоднородные преломляющие объекты. На рисунке 9 показана модель аморфного полимера (с фрагментами внутри) с низкой степенью кристалличности и низкой интенсивностью рассеяния.

Производительность метода зависит от нескольких параметров, включая количество треугольников во входной сетке, количество занятых вокселов и количество исходных фотонов. Производительность метода показана в таблице.



Рисунок 5. Объект с показателем преломления 1,1



Рисунок 6. Каустика с изменением положения источника света в реальном времени



Рисунок 7. Объемная и поверхностная каустика, создаваемая преломляющим объектом, с деформацией поверхности в реальном времени (рис. 8)



Рисунок 8. Деформация поверхности в реальном времени



Рисунок 9. Слабоанизотропный материал

Результаты тестирования

Сцена	Рис. 5	Рис. 6	Рис. 7	Рис. 9
Кол-во функционально заданных объектов	1	3	8	9
Кол-во треугольников	53K	55K	100K	118K
Разрешение объема	256 ³	256 ³	256 ³	256 ³
Занятые вокселя	90 ³	80 ³	77 ³	103 ³
Исходные фотоны	1024 ²	1024 ²	1024 ²	1024 ²
Количество шагов трассировки фотонов, млн	2,7	2,0	1,7	3,81
Количество фотонов, прошедших трассировку по объему	36K	41K	35K	143K
Генерация градиента после вокселизации, построение октодерева, объединение освещенности после отслеживания фотонов, время просмотра и пр., мс	3,79	6,25	9,38	9,75
Вокселизация, мс	2,65	3,38	5,9	6,45
Отслеживание фотонов, мс	10,8	9,6	8,52	13,8
Общее время, мс	17,24	19,23	23,8	30
Кол-во кадров в секунду	58	52	42	33

Примечания:

- 1) время вокселизации зависит от количества треугольников и занятых вокселов (см. таблицу);
- 2) время отслеживания фотонов зависит от количества занятых вокселов и исходных фотонов;
- 3) количество шагов трассировки фотонов зависит от количества занятых вокселов и исходных фотонов;
- 4) разрешение объема, количество треугольников, занятых вокселов и исходных фотонов можно менять.

Метод поддерживает динамическое обновление свойств материала, а также освещение. Материалы, находящиеся в объеме, характеризуются тремя свойствами: показателем преломления, коэффициентом затухания и коэффициентом рассеяния. Изменения показателя преломления определяют, как лучи света изгибаются при прохождении через объем. Коэффициент рассеяния влияет на то, сколько света распространяется по объему от источников света к камере. Коэффициент рассеяния определяет, насколько сильно излучение внутри каждого вокселя влияет на конечное изображение.

Поскольку работы [1] и [4] посвящены сложным зеркальным поверхностным отражениям с каустиками, а не объемам, рассмотрим наиболее близкие методы в сравнении с предлагаемым подходом.

Для визуализации скалярных полей с преломляющими средами использовались такие методы, как объемное фотонное отображение [5]. Однако эти методы являются непомерно дорогими для интерактивных приложений, поскольку количество фотонов, необходимое для того, чтобы результат казался непрерывным, слишком велико.

Метод «эйкональной визуализации», предложенный в [8], наиболее близок к визуализации всех эффектов преломления и участвующих в нем сред в режиме реального времени. Несмотря на то что визуализация с помощью этого метода позволяет создавать новые изображения в интерактивном режиме, любое изменение освещения, материалов или геометрии сцены требует повторного расчета распределения яркости, что занимает несколько секунд.

Наш метод позволяет более точно воспроизвести размер и форму каустик, чем метод [8], поскольку в нашем методе накапливается вклад излучения всех входящих фотонов в вокселе, в то время как при визуализации в методе [8] сохраняется вклад только участка волнового фронта с наибольшей энергией, который проходит через воксель. В результате при рендеринге в методе [8] не улавливаются каустические волны, которые должны генерироваться несколькими волновыми фронтами низкой интенсивности, сходящимися с разных направлений.

Также в нашем методе можно беспрепятственно распространять фотоны по объему, когда они проходят через каустику, в то время как изображения в методе [8] содержат искаженные четырехугольники волнового фронта. Это происходит, когда четыре угла участка проходят через каустические сингулярности в разное время. Некоторые из этих дефектных участков обнаруживаются и устраняются, однако это приводит к искусственной потере яркости, в то время как те, которые остаются, вносят числовую ошибку, видимую в виде шумных, неровных краев каустики. По мере того как показатель преломления объекта увеличивается по сравнению с единицей, волновые фронты пересекаются раньше, и артефакты распространения становятся все более заметными.

Кроме того, наш подход намного быстрее, чем рендеринг [8], и позволяет создавать изображения после изменения параметров освещения или материала. Можно также повторно высвечивать сцены в реальном времени, в то время как визуализация в [8] занимает порядка семи секунд всего для одного кадра. Высокая производительность заключается в использовании аддитивных размеров шагов при отслеживании фотонов. Делая меньше шагов по объему, генерируется меньше примитивов для рендеринга и используется скорость графического процессора для растеризации сегментов линий вместо точек.

Скорость нашего подхода открывает возможности для многих новых применений. Если предыдущие методы рендеринга преломляющих материалов позволяли изменять только параметры просмотра в интерактивном режиме, то теперь можно также изменять параметры освещения и материала, сохраняя при этом интерактивность обновлений. Пользователи нашей системы могут свободно изменять расположение и цвета источников света при просмотре эффектов каустики, поглощения, одиночного рассеяния и теней. Предлагаемый метод можно использовать для физического моделирования реальных процессов и анимации воображаемых процессов, например для

моделирования динамики твердого тела и жидкости, а также турбулентного перемешивания неоднородных материалов.

Весь объём обрабатываемых данных распределён между CPU и GPU. На CPU данные обрабатываются с использованием технологии для многопоточного программирования OpenMP. На GPU данные обрабатываются с использованием технологии NVIDIA CUDA.

Весь объём обрабатываемых данных делится на две неравные части: зону, обрабатываемую CPU, и зону, обрабатываемую GPU.

Граница между зоной CPU и зоной GPU в случае статической балансировки является фиксированной. В случае динамической балансировки вычислительная нагрузка изменяется в процессе выполнения программы.

Для оценки необходимости внесения корректировки в положение границы зон, на каждом n -м шаге итерации запускается процесс определения общей эффективности вычислений при текущем распределении вычислительной нагрузки. В результате этого процесса принимается решение о сохранении текущего положения границы зон или о сдвиге его на определённую величину в ту или иную сторону. В случае сдвига организовывается перераспределение данных, которые в результате сдвига начинают обрабатываться другим вычислителем.

Поскольку программа гибридная и наряду с вычислениями на CPU используются вычисления и на GPU, то в случае, когда GPU обрабатывает выделенные ему данные медленнее, чем CPU – выделенные ему, программа будет простаивать в точках синхронизации в ожидании окончания работы GPU. Вследствие этого загрузка программой CPU будет меньше 100 %. Получение программой данных о том, насколько она потребляет выделенный ей ресурс CPU, делается с помощью системного вызова.

Допускается, чтобы CPU простаивал в ожидании GPU не более 5 %. Если значение загрузки CPU больше нормального, значит, уменьшается его зона и, соответственно, увеличивается зона GPU. Если наоборот, меньше нормального, то увеличивается зона CPU.

Однако скачки в виде постоянного то уменьшения, то увеличения размеров зон неприемлемы. Для этого нормальной загрузкой CPU считается не конкретное значение, а диапазон от 85 до 95 %.

Заключение

Разработан объемно-ориентированный метод визуализации преломляющих объектов в режиме реального времени. В методе используется воксельное восьмеричное дерево для определения наибольшего размера шага при отслеживании фотонов. Также применение воксельной сетки и конвейера растеризации графического процессора для хранения данных о влиянии излучения фотонов во время прохождения процесса освещения позволяет избежать значительных затрат на поиск близлежащих фотонов на каждом этапе просмотра.

Представлена методика, позволяющая воспроизводить эффекты преломления, однократного рассеяния и поглощения даже при изменении освещения, свойств материала, геометрии и параметров обзора. Это позволяет одновременно изменять освещение, материалы и геометрию в реальном времени. Конвейер рендеринга использует графический процессор для преобразования поверхностных сеток в объемные, отслеживания фотонов в сцене и рендеринга изображения результирующего распределения яркости. Центральный процессор применяется для таких приложений, как деформация поверхности и моделирование физических процессов. Метод позволяет учитывать непрерывное изменение показателя преломления по всему объему.

Имеется несколько направлений для будущей работы. К таким работам относятся реалистичное отображение тумана, дыма и пара, физическое моделирование таких явлений, как жидкости и миражи, где эффекты преломления имеют большое значение. Для этого необходимо будет учесть многократное рассеяние во время адаптивного процесса отслеживания фотонов. Имеется также интерес к работе по включению карт окружения и других сложных источников света в конвейер рендеринга.

Источник финансирования

Работа выполнена при поддержке Министерства науки и высшего образования в рамках выполнения работ по Государственному заданию № 124041700102-4 в ИАиЭ СО РАН.

Список литературы

1. Looking Through the Glass: Neural Surface Reconstruction Against High Specular Reflections / J. Qiu [and others] // IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2023. Vancouver, BC, Canada. P. 1–20.
2. Loube G., Zeltner T., Holzschuch N. Slope-space integrals for specular next event estimation// ACM Transactions on Graphics. 2020. Vol. 39, no. 6. P. 1-13.
3. Glossy Probe Reprojection for Interactive Global Illumination / S. Rodriguez [and others] // December 2020 ACM Transactions on Graphics. 2020. Vol. 39, no 6. P. 1-16.
4. Neural Point Catacaustics for Novel-View Synthesis of Reflections / G. Kopanas [and others] // ACM Transactions on Graphics. 2022. Vol. 41, no. 6. P. 1-15.
5. Non-linear Volume Photon Mapping / D. Gutierrez [and others] // Proceedings of the Eurographics Symposium on Rendering. 2005. P. 291–300
6. Kern R., Brull F., Grosch T. Accelerating Photon Mapping for Hardware-based Ray Tracing // Journal of Computer Graphics Techniques. 2023. Vol. 12, no. 1. P. 1-28.
7. Davis S. T., Wyman C. Interactive refractions with total internal reflection // Proceedings of Graphics Interface. 2007. P. 185–190.
8. Eikonal Fields for Refractive Novel-View Synthesis / M. Bemana [and others] // ACM SIGGRAPH Conference Proceedings. 2022. P. 1 – 9.
9. Wyman C., Davis S.T. Interactive image-space techniques for approximating caustics // Proceedings of the Symposium on Interactive 3D Graphics and Games. 2006. P. 153–160.
10. A Practical Analytic Single Scattering Model for Real Time Rendering / B. Sun [and others] // ACM Transactions on Graphics. 2005. Vol. 24, no. 3. P. 1040–1049.
11. Kruger J., Bugruber K., Westerman R. Interactive Screen-Space Accurate Photon Tracing on GPUs // Rendering Techniques. 2006. P. 319–329.
12. Li S., Mueller K. Spline-based gradient filters for high-quality refraction computations in discrete datasets // Proceedings of EuroVis. 2005. P. 215–222.
13. Li S., Mueller K. Accelerated, high-quality refraction computations for volume graphics // Proceedings of Volume Graphics. 2005. P. 73–81.
14. Interactive volume caustics in single-scattering media/ W. Hu [and others] // I3D '10: Proceedings of the ACM SIGGRAPH symposium on Interactive 3D Graphics and Games. 2010. P. 109 – 117.
15. Вяткин С. И., Долговесов Б. С. Интерактивное вычисление преломления света и каустик с применением графического процессора // Программирование. 2024. № 1. С. 100-112.
16. Вяткин С. И., Долговесов Б. С. Метод сжатия геометрических данных с применением функций возмущения // Автометрия. 2018. Т. 54, № 4. С. 18- 25.
17. Vyatkin S. I. Polygonization method for functionally defined objects // International Journal of Automation, Control and Intelligent Systems. 2015. Vol. 1, no. 1. P. 1–8.
18. Principles of Optics / M. Born [and others]. (7-th expended edition). Cambridge University Press, 1999. 99 p.