

Использование технологий Embree для трассировки лучей в оптических системах с поверхностями свободной формы

Д.Д. Жданов¹, И.С. Потемин¹, А.Д. Жданов¹

¹ Университет ИТМО, Кронверкский пр. 49, лит. А, Санкт-Петербург, 197101, Россия

Аннотация

В работе рассматривается возможность использования двухуровневого представления геометрии поверхностей свободной формы для ускорения трассировки лучей и повышение устойчивости результатов трассировки для случаев «неестественного» распространения лучей, например, при анализе рассеянного света в оптических системах. В качестве поверхностей свободной формы рассматривались поверхности, заданные полиномами высокого порядка (до 34-го порядка), а также полиномами Якоби. Был проведен анализ и выявлены недостатки представления данной геометрии в явном виде и в виде треугольной сетки. Был предложен метод двойного представления геометрии в виде грубой треугольной сетки и исходного аналитического выражения. На базе библиотеки Embree были разработаны методы трассировки лучей для геометрических поверхностей, имеющих двойное представление. Разработанные методы показали значительное ускорение процесса трассировки лучей при сохранении точности и высокой стабильности вычислений. В работе приводятся результаты расчета функции рассеивания точки и бликов для объективов с асферическими поверхностями, заданными полиномами Якоби.

Ключевые слова

Поверхность свободной формы, Embree, трассировка лучей, SIMD, рассеянный свет.

Using Embree Technologies for Ray Tracing in Optical Systems with Free-form Surfaces

Dmitry Zhdanov¹, Igor Potemin¹, Andrey Zhdanov¹

¹ ITMO University, Kronverksky Pr. 49, bldg. A, St.-Petersburg, 197101, Russia

Abstract

The paper considers the possibility of using a two-level representation of the geometry of free-form surfaces to speed up the ray tracing process and increase the stability of the tracing results for cases of "unnatural" ray propagation, for example, in the analysis of scattered light in optical systems. Surfaces defined by high-order polynomials (up to the 34th order), as well as Jacobi polynomials, were considered as free-form surfaces. The analysis was carried out and the shortcomings of the representation of this geometry in an explicit form and in the form of a triangular grid were revealed. A method was proposed for the double representation of geometry in the form of a coarse triangular grid and the original analytical expression. Based on the Embree library, ray tracing methods were developed for geometric surfaces that have a dual representation. The developed methods have shown a significant acceleration of the ray tracing while maintaining the accuracy and high stability of calculations. The paper presents the results of calculating the point spread function and flare for lenses with aspherical surfaces given by Jacobi polynomials.

Keywords

Free Form Surface, Embree, Ray Tracing, SIMD, Stray Light.

ГрафиКон 2023: 33-я Международная конференция по компьютерной графике и машинному зрению, 19-21 сентября 2023 г., Институт проблем управления им. В.А. Трапезникова Российской академии наук, г. Москва, Россия

EMAIL: ddzhdanov@mail.ru (Д.Д. Жданов); ipotemin@yandex.ru (И.С. Потемин); andrew.gtx@gmail.com (А.Д. Жданов)

ORCID: 0000-0001-7346-8155 (Д.Д. Жданов); 0000-0002-5785-7465 (И.С. Потемин); 0000-0002-2569-1982 (А.Д. Жданов)



© 2023 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

1. Введение

Средства компьютерной графики позволяют не только выполнять реалистичный рендеринг трехмерных сцен, используемый в игровой индустрии, системах виртуальной и смешанной реальности, но и решать задачи виртуального прототипирования оптических систем освещения и наблюдения. Последнее становится особенно важным в связи с появлением новых технологий проектирования и производства элементов оптических устройств. С точки зрения вычислительной оптики это так называемые поверхности свободной формы. К поверхностям свободной формы можно отнести поверхности, не имеющие аналитического представления (как правило, такие поверхности используются в осветительных системах и обеспечивают заданное распределение силы света), и поверхности, заданные полиномами высокого порядка, например, полиномами Якоби [1] (как правило, такие поверхности используются в линзовых объективах, например, в камерах смартфонов или в системах дополненной реальности). На рисунке 1 представлен пример объектива камеры дополненной реальности.

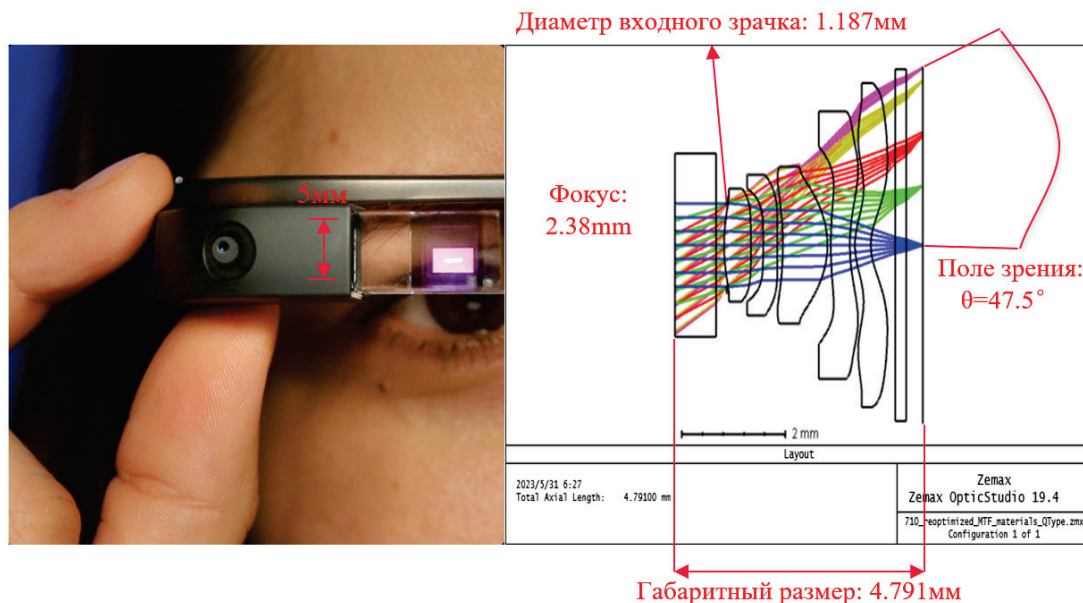


Рисунок 1 – Пример объектива камеры дополненной реальности

Использование полиномов Якоби с параметрами разложения, представленными в работе [2], позволяет избежать ошибок округления при использовании степенных полиномов высокого порядка. Например, оптические поверхности рассмотренного ниже объектива имеют степень полинома, равную 34. Кроме того, для полиномов такого типа можно оценить проблемы, связанные со сложностью их изготовления [3].

До настоящего времени основной метод расчета точки пересечения луча с асферической поверхностью, используемый в вычислительной оптике, – это метод Федера [4]. Он был предложен более 70 лет назад и до настоящего времени не претерпел никаких принципиальных изменений. Суть метода заключается в последовательной подгонке точки встречи луча с касательной плоскостью к асферической поверхности, при этом параметры касательной плоскости последовательно уточняются в точках пересечения луча с некоторой базовой поверхностью, аппроксимирующей асферическую поверхность. Как правило, за начальную базовую поверхность выбирается плоскость:

$$z = 0, \quad (1)$$

где ось Z – оптическая ось, а асферическая поверхность задана в локальной системе координат с вершиной в точке $(0, 0, 0)$. В процессе подгонки луча базовая плоскость смещается в направлении к точке пересечения луча с асферической поверхностью.

Для всех типов асферических поверхностей [1], будь то стандартное полиномиальное представление, описанное выражением (2),

$$z(\rho) = \frac{c\rho^2}{1 + \sqrt{1 - \varepsilon c^2 \rho^2}} + \sum_{m=0}^M a_m \rho^{2m+4}, \quad (2)$$

или так называемая Q-типе асферическая поверхность (3), заданная полиномом Якоби,

$$z(\rho) = \frac{c\rho^2}{1 + \sqrt{1 - \varepsilon c^2 \rho^2}} + D_{con} \left(\frac{\rho}{\rho_{max}} \right), \quad (3)$$

используется одинаковый способ подгонки луча к поверхности.

где c, ε, a_m – коэффициенты асферической поверхности, $\rho^2 = x^2 + y^2$, $D_{con}(u) = u^4 \sum_{m=0}^M a_m Q_m^{con}(u^2)$, Q_m^{con} – задаются полиномами Якоби следующего вида:

$$Q_m^{con}(u) = P_m^{(0,4)}(2u - 1), \quad (4)$$

где $P_0^{(0,4)}(u) = 1$, $P_1^{(0,4)}(u) = -(5 - 6u)$, $P_2^{(0,4)}(u) = 15 - 14u(3 - 2u)$, и т.д.

Незначительные модификации метода, предложенного Федером, касались возможности его применения к нецентрированным поверхностям [5] и дальнейшего его развития для применения к различного рода неосесимметричным поверхностям, включая поверхности дифракционных решеток [6].

Очевидно, что вычисление полиномов является достаточно затратной процедурой, тем более, что для определения точки встречи луча с поверхностью, описанной полиномом, требуется многократное повторение данных вычислений особенно в случае, если точка встречи луча с полиномиальной поверхностью может сильно отличаться от начальной базовой плоскости. При высокой степени полинома вычисление его значений может привести к потере точности, а в ряде случаев и к переполнению данных. Это особенно проявляется при «нестандартном» ходе луча, например, при расчете бликов, возникающих между линзовыми поверхностями.

Для ускорения расчета точки встречи луча с асферической поверхностью можно использовать подходы, применяемые в компьютерной графике. Простейшим решением служит задание асферической поверхности в виде треугольной сетки, вершины которой совпадают с точками на асферической поверхности, а нормали направлены в соответствии с истинными нормальными в точках этой поверхности. Однако, для достижения высокой точности треугольная сетка должна быть очень высокого разрешения, что отрицательно сказывается на производительности вычислений и объеме памяти, необходимого для хранения этой сетки. Поэтому в данной работе предлагается использование двухуровневого представления геометрии: на первом уровне треугольная сетка низкого разрешения для определения начальной точки встречи луча с поверхностью, а на втором уровне – собственно асферическая поверхность.

2. Требуемая точность представления геометрии

Для оценки точности представления геометрии в виде треугольной сетки было проведено исследование зависимости точности трассировки лучей на примере расчета функции рассеивания точки (построение изображения точечного объекта методом прямой трассировки лучей) от разрешения треугольной сетки (разбивки асферической поверхности по азимутальному углу и высоте на поверхности линзы ρ). Исследование проводилось для двух видов асферических поверхностей, представленных формулами (2) и (3) соответственно. Для моделирования оптической системы с асферической поверхностью, представленной формулой (2), использовался четырехлинзовый объектив, показанный на рисунке 2(а), а для моделирования оптической системы с асферической поверхностью, представленной формулой (3), использовался десятилинзовый объектив, показанный на рисунке 2(б). Диаметр входных зрачков обоих объективов составлял порядка 1 мм. Для вычисления функции рассеивания точки использовалось два решения. Базовое решение было реализовано на алгоритмах Федера [4] и расширении для асферических поверхностей, представленных полиномами Якоби [1]. Второе решение – это представление геометрии в виде треугольной сетки, вершины которой совпадают с асферической поверхностью, и нормали в вершинах совпадают с нормальными к асферической поверхности в этих вершинах. В последнем случае трассировка лучей осуществлялась средствами компьютерной графики [7].

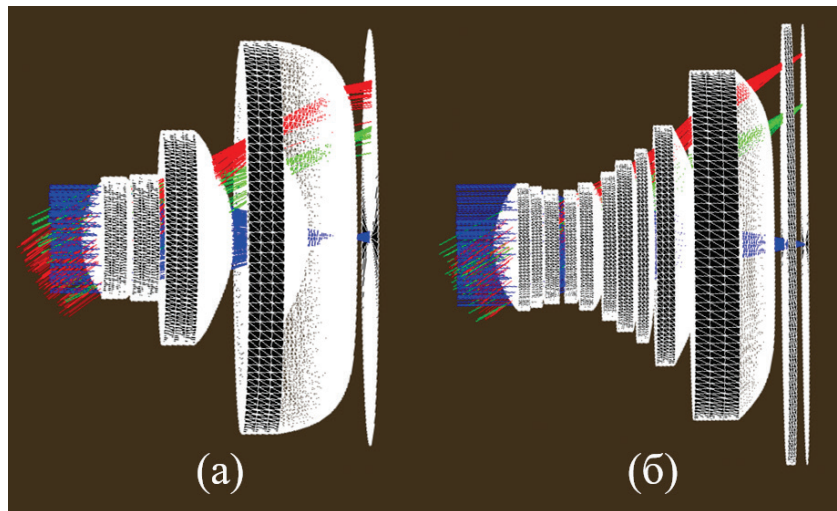


Рисунок 2 – Пример объективов, используемых для расчета функции рассеивания точки, (а) – четырехлинзовый объектив с асферической поверхностью, определяемой уравнением (2), (б) – десятилинзовый объектив с асферической поверхностью, определяемой уравнением (3)

Оценка точности расчета проводилась для бесконечно удаленной точки объекта, находящейся на оси объектива. Пучок лучей, выходящий из точки на объекте, формировал прямоугольную сетку, равномерно распределенную по входному зрачку объектива. Пучок имел разрешение 1024×1024 . Вычисления производились на восьмиядерном компьютере AMD Ryzen 9 с 32 Гб оперативной памяти. Оценка скорости вычислений в данном тесте не проводилась, основной упор делался на определение необходимой точности представления геометрии оптических поверхностей линз. Был рассмотрен ряд вариантов представления в виде треугольной сетки. В случае минимального разбиения асферической поверхности сетка содержала 1441 треугольник, а в максимуме, когда изображения, вычисленные по алгоритмам [4] и [1], визуально совпали с изображениями на треугольной сетке, сетка содержала 4320001 треугольник. Соответствующие изображения функции рассеивания точки представлены на рисунках 3(а) – изображения, вычисленные по алгоритмам [4] и [1], 3(б) – изображения, вычисленные для сетки с 1441 треугольником, 3(в) – изображения, вычисленные для сетки с 48001 треугольником и 3(г) – изображения, вычисленные для сетки с 4320001 треугольником. Размер изображения для четырехлинзового объектива, см. рисунок 2(а), был 100 мкм, для десятилинзового объектива, см. рисунок 2(б), – 25 мкм.

Видно, что использование треугольной сетки позволяет добиться корректного результата, однако использование миллионов треугольников для описания одной поверхности является достаточно накладной операцией как с точки зрения времени вычислений, так и с точки зрения оперативной памяти, необходимой для хранения информации о геометрии объектов. Поэтому, для ускорения расчетов и уменьшения объема данных геометрии предлагается комбинированное решение, заключающееся в двойном представлении геометрии и сочетающее грубое представление геометрии в виде треугольной сетки и итерационные процедуры по алгоритмам [4] и [1]. Это решение напоминает алгоритм “displacement mapping” [8], однако данное решение нацелено не только на снижение объема данных, но и на ускорение расчета.

3. Специализированный метод трассировки лучей в оптических системах

Для решения данной задачи был написан специализированный трассировщик лучей, ориентированный на синтез изображений, формируемых оптическими системами. В качестве базового решения для трассировки лучей была использована библиотека Embree от компании Intel [9], работающая на центральном процессоре. Основные причины выбора API Embree перечислены ниже:

- API имеет богатый программный интерфейс для формирования геометрии, формирования пространственного разбиения сцены, трассировки и обработки результатов трассировки лучей.
- API не содержит избыточных функций, ориентированных на процесс рендеринга.
- API обеспечивает качественное пространственное разбиение геометрии и высокопроизводительные методы поиска пересечения луча с геометрией.
- API поддерживает необходимые способы задания геометрии, включая треугольную сетку и геометрию, заданную пользователем.
- API поддерживает все возможные виды SIMD инструкций.
- API поддерживает параллельные вычисления на базе Intel TBB API [10].
- API поддерживает специальные функции, позволяющие модифицировать геометрию поверхности во время трассировки лучей.

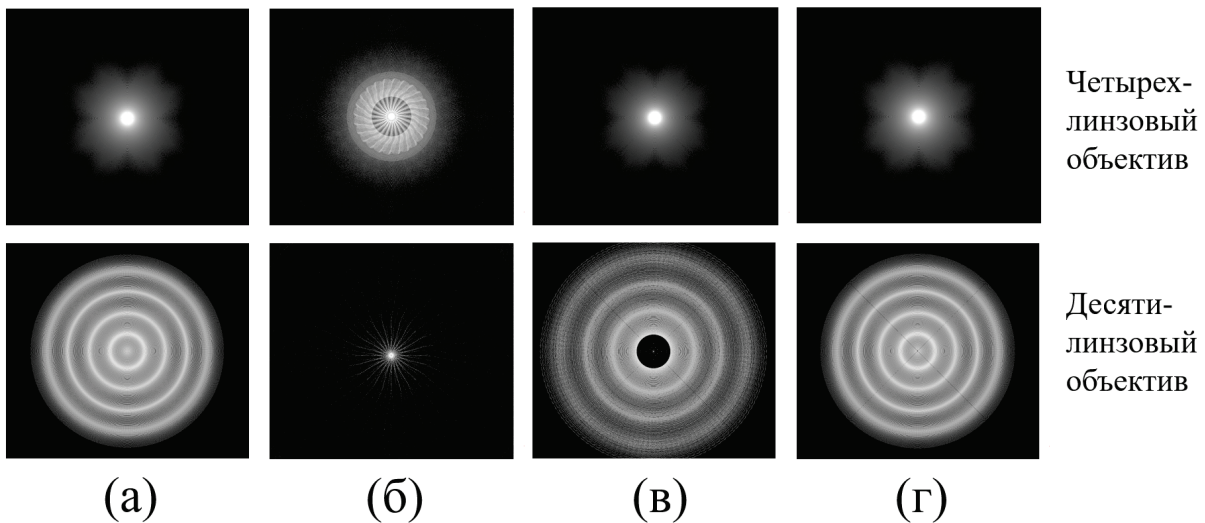


Рисунок 3 – Результаты моделирования функции рассеивания точки десятилинзового объектива (верхний ряд) и четырехлинзового объектива (нижний ряд), (а) – вычисление по алгоритмам [4] и [1], (б) – вычисление по сетке из 1441 треугольника, (в) – вычисление по сетке из 48001 треугольника, (г) – вычисление по сетке из 4320001 треугольника

Учитывая специфику задач прикладной оптики, а именно то, что луч распространяется последовательно от одной поверхности оптической системы к следующей поверхности по порядку их задания в оптической системе, а также возможности API Embree, позволяющие на одном устройстве иметь неограниченное количество сцен, было принято решение о формировании отдельных Embree сцен для каждой поверхности оптической системы. Данное решение позволяет значительно уменьшить число треугольников в сцене, что в итоге ускоряет процесс поиска точки встречи луча с треугольной сеткой. Кроме того, это решение позволяет избежать неоднозначности при поиске точки встречи луча с поверхностью, если расстояние между поверхностями достаточно мало, например, поверхности оптической системы находятся в оптическом контакте или имеют очень малый зазор, возникающий в результате склеивания двух поверхностей. Данное решение проиллюстрировано на рисунке 4.

Таким образом, трассировка лучей в сцене заключается в последовательном переборе Embree сцен, где последовательность перебора задается пользователем. Например, последовательность $\{1, 2, 3, \dots, N\}$ (где N – число поверхностей в оптической системе) говорит о распространении света от объекта до изображения, а последовательность $\{N, N - 1, N - 2, N - 3, \dots, 1\}$ говорит о распространении света от изображения к источнику света. Последовательность перебора поверхностей может быть достаточно произвольной, что позволяет моделировать многоканальные и мультиконфигурационные оптические системы. Кроме того, последовательность может быть и случайной, что позволяет моделировать такой эффект как блики на линзовых поверхностях оптических элементов. В последнем случае последовательность следования сцен определяется вероятностным образом в результате

случайного выбора типа события преобразования луча на поверхности (отражение или преломление). Необходимо отметить, что последовательность поверхностей в оптических системах строго фиксирована. Если луч находится на поверхности K , а следующая для луча поверхность определена как M , то в случае непопадания луча на поверхность M он рассматривается как убитый и трассировка прекращается. Луч также может прекратить свое существование, если реальный тип события преобразования луча на поверхности не соответствует требуемому, например, в случае преломления возникает эффект полного внутреннего отражения. Поскольку распространение луча является детерминистической операцией, луч в процессе своего распространения может менять свою энергию для заданной длины волны. Кроме того, для возможного последующего расчета дифракционных эффектов происходит вычисление оптического пути луча, поэтому на выходе из оптической системы луч содержит не только энергию, но и оптический путь. Трассировка прекращается, если луч удачно покинул последнюю поверхность из списка.

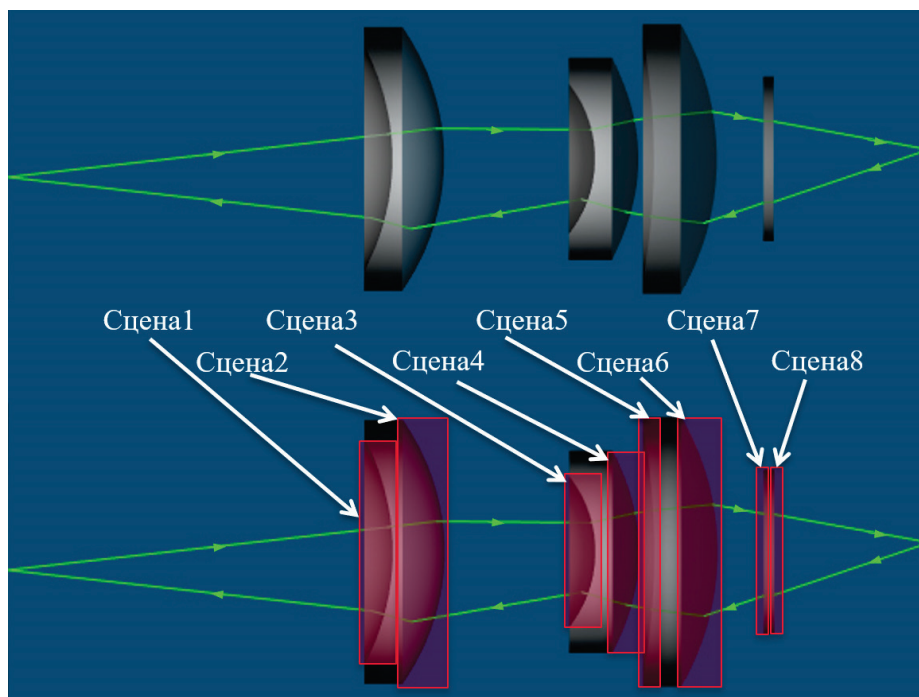


Рисунок 4 – Представление оптической системы в виде набора Embree сцен, сверху – исходная оптическая система, внизу – оптическая система, разбитая на ряд Embree сцен (каждой поверхности соответствует одна сцена) с граничными объемами для каждой поверхности оптической системы

Данное решение может использоваться не только в задачах прикладной оптики. Это решение может найти применение как в задачах компьютерной графики, так и в задачах виртуального прототипирования. Основное преимущество данного решения – внедрение детерминистической модели трассировки лучей в задачи стохастического моделирования. И это может быть не только модель линзовой камеры, формирующей пучки лучей, идущих от точек изображения к сцене в методах, использующих обратную трассировку лучей. Это может быть и модель сложного источника освещения, например, проекционный объектив или просто модель оптической системы, находящейся внутри сцены. Конечно, данная модель имеет ограничения, поскольку не содержит замкнутую конструкцию оптической системы, что не дает возможности моделировать сложные эффекты рассеивания света на нерабочих поверхностях оптического устройства. Однако, если моделирование не ставит задачу расчета рассеянного света на нерабочих поверхностях оптического устройства, данное решение может на порядки ускорить процесс распространения света в оптической системе, сохраняя при этом физическую корректность модели.

Второй момент, который был решен с помощью API Embree, – это двойное представление геометрии. Основная проблема треугольной сетки – это ее большой объем, а для аналитической

формы, решаемой алгоритмами [4] и [1], – это большое число итераций, необходимых для достижения заданной точности поиска точки встречи луча с поверхностью. Двойное представление геометрии позволяет решить обе эти проблемы. Геометрия линзы задается треугольной сеткой низкого разрешения и служит для определения начальной точки встречи луча с поверхностью. Далее от этой начальной точки начинается процесс подгонки луча по алгоритмам [4] и [1]. Алгоритм этого подхода представлен на рисунке 5.

На рисунке 5 представлены два подхода к определению начальной точки встречи луча с асферической поверхностью. Нижняя часть рисунка демонстрирует положение первого приближения точки встречи луча с поверхностью в случае использования алгоритмов [4] и [1], верхняя часть – в случае двойного представления. Для реализации двойного представления геометрии была использована специальная функциональность API Embree, а именно, возможность задания метода фильтрации, вызываемого функцией RTCIntersect после вычисления точки встречи луча с поверхностью. Метод фильтрации вызывается независимо от того, было ли найденное пересечение ближайшим или нет. Метод фильтрации позволяет переопределить точку встречи луча с поверхностью, включая нормаль в точке встречи, и вернуть новые параметры пересечения функции RTCIntersect, которая рассматривает полученный результат, как результат своего поиска.

В результате алгоритм работает следующим образом: функция RTCIntersect находит пересечение луча с треугольной сеткой и вызывает метод фильтрации с найденными результатами пересечения, далее метод фильтрации запускает алгоритм подгонки [4], [1], начиная с точки пересечения с треугольной сеткой, и находит пересечение с асферической поверхностью, полученные параметры (дистанция и нормаль) возвращаются функции RTCIntersect, как результат поиска точки встречи луча с асферической поверхностью. Полученное решение позволяет как ускорить процесс поиска точки встречи с асферической поверхностью, так и незначительно увеличить память за счет использования дополнительной, сильно разреженной, треугольной сетки.

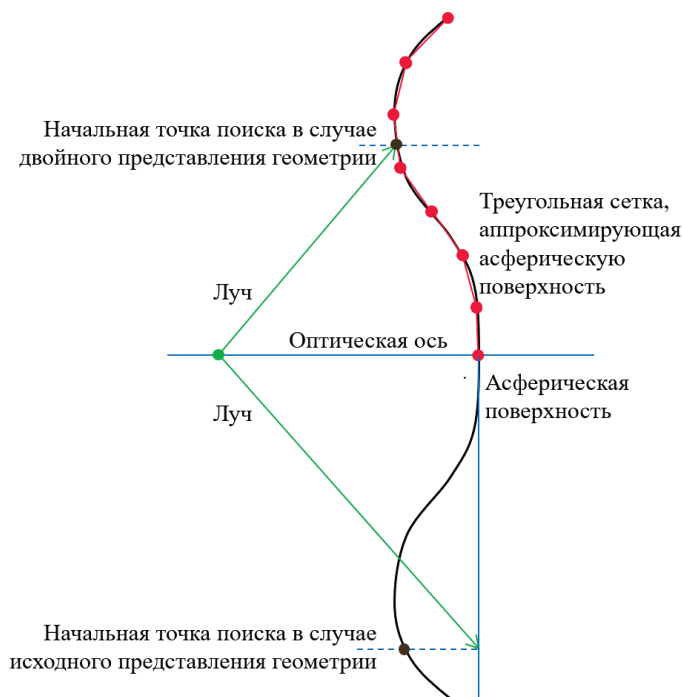


Рисунок 5 – Определение начальной точки встречи луча с асферической поверхностью, верхняя часть рисунка демонстрирует случай двойного представления геометрии (треугольная сетка и аналитическая формула), нижняя часть рисунка демонстрирует случай работы с одной аналитической формулой

Следующими решениями, использованными для ускорения поиска точки встречи луча с асферической поверхностью, были распараллеливание расчетов с помощью TBB API от

компании Intel и использование векторных операций с помощью SIMD инструкций. Необходимо отметить, что Embree API обеспечивает интерфейсы, дающие возможность эффективно использовать SIMD инструкции.

Трассировка лучей осуществлялась пучками в параллельном режиме. TBB запускает ряд параллельных потоков, и в каждом потоке функции RTCIntersect передается пучок из десятков тысяч лучей, для которых выполняется операция поиска точки встречи луча с поверхностью. Как было отмечено выше, поиск точки пересечения выполняется в два этапа. На первом этапе выполняется поиск пересечения с геометрией сцены, а на втором – фильтрация, осуществляющая подгонку к асферической поверхности. Поскольку Embree работает с SIMD инструкциями, то результат метода фильтрации передается в виде вектора (в случае AVX2 это восемь чисел в формате int или float). Поскольку точность float может быть недостаточной для выполнения операции подгонки, эта операция выполняется на числах двойной точности. В результате вектор из восьми float разбивается на два вектора из четырех double. Эта процедура показана на рисунке 6. Хотя данная процедура замедляет процесс вычислений, она позволяет повысить точность, что может оказаться важным при трассировке лучей в оптических системах с асферическими поверхностями высокого порядка (в данных примерах рассматривались поверхности с асферическими поверхностями, заданными полиномами, включая полиномы Якоби, 34-й степени).

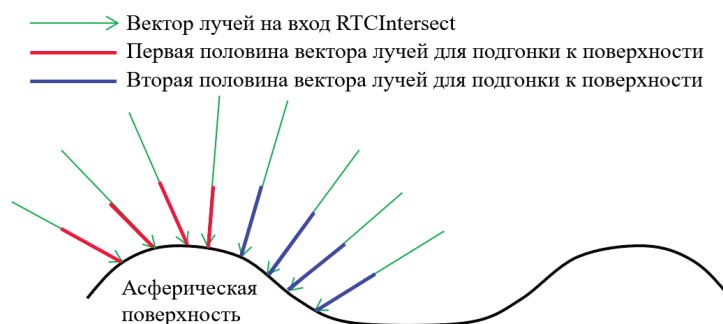


Рисунок 6 – Разбиение вектора из восьми лучей на выходе RTCIntersect для работы с числами двойной точности в методе подгонки лучей

Общий алгоритм расчета функции рассеивания точки (PSF – point spread function) представлен на рисунке 7.

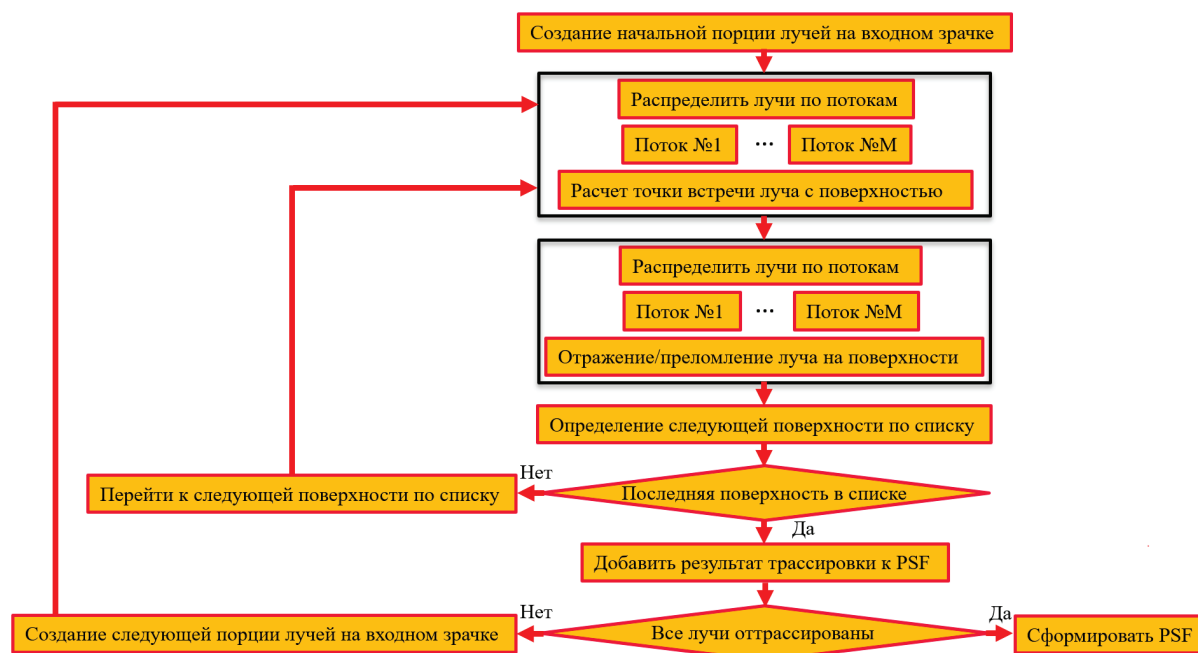


Рисунок 7 – Алгоритм расчета функции рассеивания точки

Данный алгоритм может работать не только для расчета функции рассеивания точки, но и для синтеза изображений или расчета бликов. Различие заключается только в методе формирования начальных лучей на входном (или выходном) зрачке и в выходной информации, например, для расчета бликов оптический путь не сохраняется. Поскольку линзовые элементы обладают дисперсией, весь расчет выполняется для одной длины волны. Если необходимо построить «цветное» изображение, то расчет выполняется для заданного набора длин волн и результаты всех монохроматических расчетов сливаются в один спектральный результат. Данный алгоритм поддерживает не только преломление на линзовых элементах. Оптическая система может содержать зеркала, что позволяет моделировать отражение. Кроме того, для анализа бликов преобразование луча на поверхности может иметь случайный характер и определяться формулами Френеля или коэффициентами зеркального отражения, учитывающими, например, эффект просветления линзовых поверхностей.

4. Тестирование разработанных алгоритмов

В соответствии с разработанными алгоритмами была реализована программа расчета функции рассеивания точки и бликов в зеркально-линзовых объективах. Было проведено тестирование времени расчета функции рассеивания точки по разработанным алгоритмам и исходным решениям, описанным в [4] и [1]. Для тестирования использовались объективы, представленные на рисунке 2. Тестирование производилось на восьмиядерном компьютере AMD Ryzen 9 с 32 Гб оперативной памяти. В модели геометрии поверхности линзы использовались 1279 треугольников, число трассируемых лучей задавалось исходя из сетки на входном зрачке 1024x1024, т.е. 1048576 лучей. Поля зрения для четырехлинзового объектива составляли 0° , 26° и 37° , для десятилинзового объектива поля зрения составляли 0° , 28° и 38° . Изображения функции рассеивания точки, полученные в результате тестирования, представлены на рисунке 8. Время, затраченное на трассировку данного числа лучей для разработанного алгоритма и алгоритма с подгонкой, описанного в работах [4] и [1], представлено в таблице 1. Изображения, полученные с использованием новых и старых алгоритмов совпадают и не приводятся на рисунке 8.

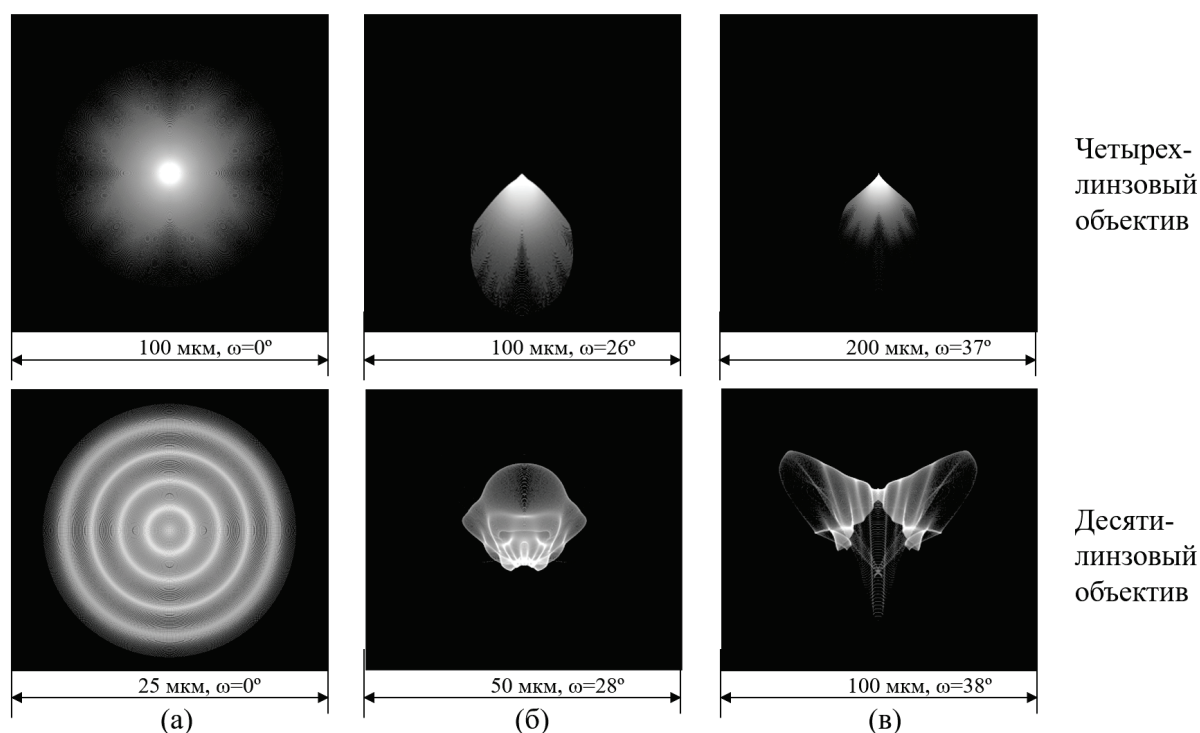


Рисунок 8 – Изображения, формируемые объективами, представленными на рисунке 2, для трех точек поля зрения

Таблица 1 – Время расчета трех полей зрения для двух объективов

Четырехлинзовый объектив	$\omega=0^\circ$	$\omega=26^\circ$	$\omega=37^\circ$
	время расчета в мс	время расчета в мс	время расчета в мс
Двухуровневая модель	43	36	32
Модель с подгонкой	39	48	51
Десятилинзовый объектив	$\omega=0^\circ$	$\omega=28^\circ$	$\omega=38^\circ$
	время расчета в мс	время расчета в мс	время расчета в мс
Двухуровневая модель	160	89	46
Модель с подгонкой	162	120	110

Данное исследование показало, что для осевых пучков предложенный метод не дает ускорения расчета (что вполне естественно), однако для внеосевых точек наблюдается значительное ускорение (в ряде случаев более чем в два раза), связанное со значительным уменьшением числа итераций в процессе подгонки луча к асферической поверхности, см. рисунок 5. Очевидно, что процесс поиска точки встречи луча с треугольной сеткой требует значительно меньше времени, чем подгонка луча к асферической поверхности.

Поскольку для моделирования и проектирования оптических систем осевой пучок не является основополагающим, то предложенная двухуровневая модель позволит значительно ускорить процесс трассировки лучей и расчет функций качества изображения. Другим примером использования данного решения является расчет бликов в линзовых системах. Для десятилинзового объектива была задана модель оптических свойств поверхностей, соответствующая непросветленной оптике. Факт отражения или пропускания определялся вероятностным образом по коэффициенту отражения, вычисляемого по формуле Френеля. В соответствии с выбором события отражения или преломления формировался список чередования поверхностей оптической системы для трассировки лучей. Результат моделирования бликов для двух внеосевых полей зрения $\omega=40^\circ$ и $\omega=45^\circ$ представлен на рисунке 9.

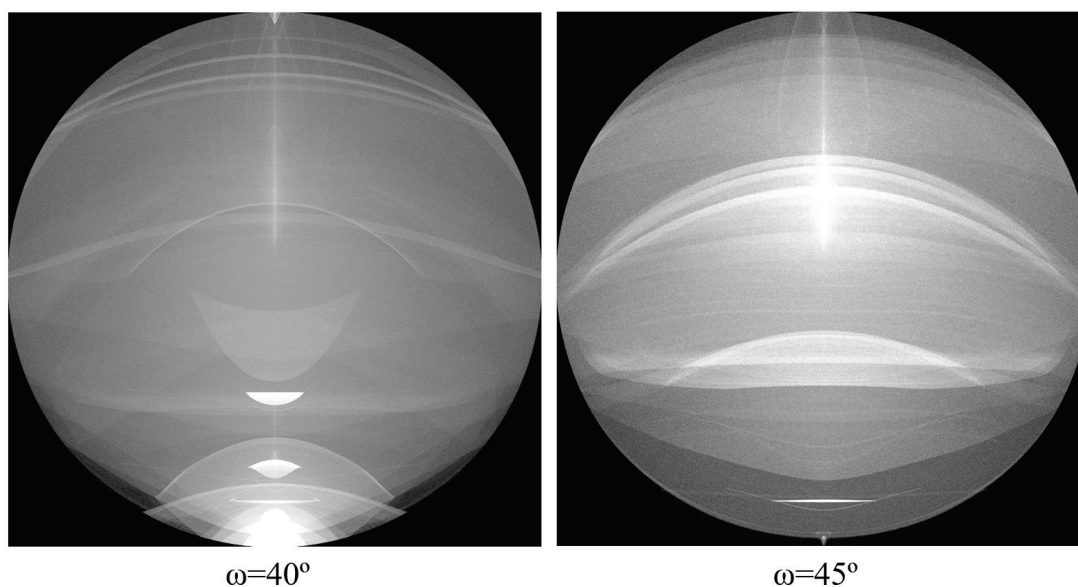


Рисунок 9 – моделирование бликов в десятилинзовом объективе для внеосевых полей зрения $\omega=40^\circ$ (левое изображение) и $\omega=45^\circ$ (правое изображение)

Кроме того, разработанное решение может стать частью системы виртуального прототипирования и рендеринга, если модель сцены содержит оптические системы, например, фото или проекционные объективы.

5. Заключение

Проведенное исследование показало эффективность использования двухуровневого представления геометрии асферических поверхностей оптических систем для задач прикладной оптики и компьютерной графики. Основным критерием эффективности было ускорение процедуры поиска точки встречи луча с поверхностью при незначительном увеличении объема данных, необходимых для хранения дополнительной информации об упрощенной форме поверхности. Кроме повышения эффективности было отмечено повышение стабильности вычислений точки встречи луча с асферической поверхностью, особенно для лучей, распространяющихся «неестественным» образом, например, при расчете бликов, когда появляются лучи, распространяющиеся вдоль оптической поверхности.

В качестве дальнейшего развития рассматривается возможность внедрения модуля трассировки лучей в оптических системах в программы реалистичной визуализации, для моделирования оптически сложных сцен, содержащих зеркально-линзовые системы, включая системы освещения и наблюдения.

6. Благодарности

Работа была выполнена при финансовой поддержке Российского Научного Фонда, грант № 22-11-00145.

7. Список источников

- [1] G. W. Forbes. Shape specification for axially symmetric optical surfaces // Opt. Express. 2007. № 15. С. 5218–5226.
- [2] G. W. Forbes. Robust, efficient computational methods for axially symmetric optical aspheres // Opt. Express 2010. № 18. С. 19700–19712.
- [3] K. P. Thompson, F. Fournier, J. P. Rolland, and G. W. Forbes. The Forbes Polynomial: A more predictable surface for fabricators // International Optical Design Conference and Optical Fabrication and Testing / OSA Technical Digest (CD) (Optica Publishing Group, 2010), paper OTuA6.
- [4] D. Feder. Optical Calculations with Automatic Computing Machinery // J. Opt. Soc. Am. 1951. № 41. С. 630.
- [5] W. Allen, J. Snyder, Ray Tracing through Uncentered and Aspheric Surfaces // J. Opt. Soc. Am. 1952. С. № 42, 243
- [6] G. H. Spencer M., V. R. K. Murty. General Ray-Tracing Procedure // J. Opt. Soc. Am. 1962. № 52. С. 672-678.
- [7] Physically Based Rendering: From Theory to Implementation / M. Pharr, W. Jakob, G. Humphreys. М.: Morgan Kaufmann 2016. 1266 p.
- [8] Kimmo Karhu. Displacement Mapping // Tik-111.500 Seminar on computer graphics / Telecommunications Software and Multimedia Laboratory. Spring 2002: Rendering high-quality 3D graphics.
- [9] GitHub - embree/embree: Embree ray tracing kernels repository, URL: <https://github.com/embree/embree>.
- [10] Intel® Threading Building Blocks Tutorial (ed.ac.uk), URL: <https://www.inf.ed.ac.uk/teaching/courses/ppls/TBBtutorial.pdf>.