



ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 25 за 2025 г.



ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

**Ван Янь, Д.Д. Жданов,
И.С. Потемин, А.Д. Жданов,
Н.Б. Дерябин**

**Метод быстрой оценки
функции рассеяния точки для
анализа и визуализации
характеристик
проектируемой оптической
системы**

Статья доступна по лицензии
[Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/)



Рекомендуемая форма библиографической ссылки: Метод быстрой оценки функции рассеяния точки для анализа и визуализации характеристик проектируемой оптической системы / Ван Янь [и др.] // Препринты ИПМ им. М.В.Келдыша. 2025. № 25. 21 с. EDN: [OUNKDN](https://doi.org/10.26907/2071-2898.2025.25)
<https://library.keldysh.ru/preprint.asp?id=2025-25>

**Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Российской академии наук**

**Ван Янь, Д.Д. Жданов, И.С. Потемин, А.Д. Жданов,
Н.Б. Дерябин**

**Метод быстрой оценки функции
рассеяния точки для анализа
и визуализации характеристик
проектируемой оптической системы**

Москва — 2025

Ван Янь, Жданов Д.Д., Потемин И.С., Жданов А.Д., Дерябин Н.Б.

Метод быстрой оценки функции рассеяния точки для анализа и визуализации характеристик проектируемой оптической системы

В работе рассматривается возможность использования двухуровневого представления геометрии поверхностей свободной формы для ускорения трассировки лучей и повышения 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. Проведен анализ, и выявлены недостатки представления данной геометрии в явном виде и в виде треугольной сетки. 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 рассеяния точки и бликов для объективов с асферическими поверхностями, заданными полиномами Якоби.

Ключевые слова: трассировка лучей, асферическая поверхность высокого порядка, функция рассеяния точки

Wang Yan, Dmitry Dmitrievich Zhdanov, Igor Stanislavovich Potemin, Andrey Dmitrievich Zhdanov, Nikolay Borisovich Deryabin

Method for fast point spread function estimation for the analysis and visualization of the characteristics of the designed optical system

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.

Key words: ray tracing, high order aspherics, point spread function

Работа выполнена при поддержке Российского научного фонда, грант № 22-11-00145.

1. Введение

С развитием технологии изготовления оптических деталей все больше и больше разработчиков оптических систем в последнее время используют поверхности свободной формы и асферические поверхности высокого порядка из-за их широких возможностей по коррекции aberrаций. Поверхности свободной формы используются в основном в осветительных системах и обеспечивают заданное распределение силы света, а асферические поверхности высокого порядка используются главным образом в линзовых объективах, например в камерах смартфонов или в системах дополненной реальности. Полтора десятилетия назад был предложен новый тип описания асферических поверхностей [1], который определяет профиль поверхности с помощью полиномов Якоби. Этот тип получил название «асферика Q-типа». Такое представление сложных асферических поверхностей устойчиво к произвольному порядку и позволяет избежать ошибок округления [2], а также позволяет количественно оценить сложность изготовления асферической поверхности такого рода [3]. Примеры типичных объективов мобильных камер, построенных на базе линз с асферическими поверхностями Q-типа, представлены на рисунке 1.

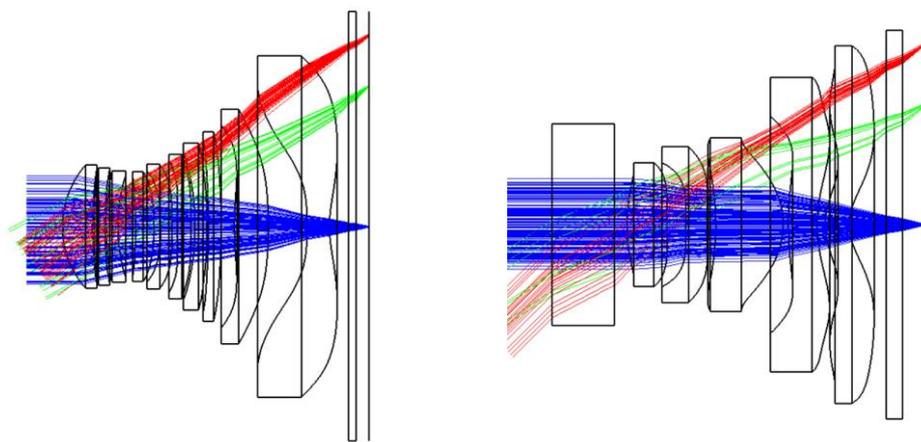


Рис. 1. Типичные компоновки оптических систем камер мобильного устройства с асферическими поверхностями Q-типа

При использовании в САПР асферических поверхностей высокого порядка, заданных тем или иным типом полинома, возникает проблема эффективной и физически корректной трассировки лучей через такой тип поверхности. Наиболее важным этапом трассировки лучей является нахождение точки пересечения луча с асферической поверхностью, заданной полиномом высокого порядка. Одним из основных методов нахождения пересечения луча с такой асферической поверхностью является метод последовательных приближений, опубликованный Федером [4]. На рисунке 2 показаны два этапа процесса нахождения точки пересечения луча с асферической поверхностью.

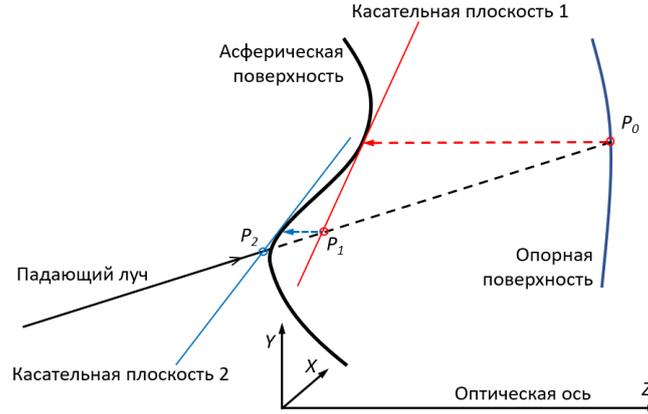


Рис. 2. Определение точки встречи луча с асферической поверхностью методом последовательных приближений

Сначала вычисляются координаты x_P и y_P точки P_0 , в которой падающий луч пересекает опорную поверхность $z_o(x, y)$. Как правило, это сфера или плоскость. Затем в этой точке вычисляются координата $z_A(x_P, y_P)$ асферической поверхности и ее производные. Эти значения используются для вычисления касательной плоскости 1 к поверхности в этой точке. Далее определяется новая точка пересечения P_1 , которая является первым приближением к искомой асферической поверхности. Затем итерационный процесс повторяется до тех пор, пока расстояние между двумя последними касательными плоскостями не достигнет заданной точности. Полученная точка возвращается как результат поиска пересечения луча с асферической поверхностью.

Такой способ нахождения точки встречи луча поверхностью используется для всех типов асферических поверхностей, включая как стандартное полиномиальное представление:

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

так и представление в виде полиномов Якоби:

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

где z – оптическая ось, $\rho = \sqrt{x^2 + y^2}$, c – кривизна поверхности в вершине ($x = y = 0$), k – коническая константа, a_m – коэффициенты полинома, $D_{con}(u) = u^4 \sum_{m=0}^M a_m Q_m^{con}(u^2)$, Q_m^{con} задаются полиномами Якоби следующего вида:

$$Q_m^{con}(u) = P_m^{(0,4)}(2u - 1), \quad (3)$$

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.

Чуть позже метод, предложенный Федером, был расширен для нецентрированных поверхностей [5]. Спустя десятилетие Спенсер расширил этот метод на различные типы поверхностей, включая дифракционные решетки [6]. Но принцип определения точки пересечения луча с асферической поверхностью высокого порядка остался прежним: итерационный метод последовательного приближения. Впрочем, такой метод имеет ряд проблем. Одна из проблем заключается в том, что это трудоемкий процесс, поскольку для достижения требуемой точности необходимо много итераций. Другая проблема метода заключается в том, что для сложных форм требуется моделирование с двойной точностью.

Метод работает правильно и быстро, когда лучи попадают на опорную поверхность (плоскость или сферу), форма которой близка к форме асферической поверхности, и наклон луча к нормали невелик (рис. 3а). Метод работает неправильно и может замедлить трассировку лучей, когда лучи не попадают на поверхность или попадают на нее вблизи касательной плоскости (рис. 3б). Но именно эта ситуация имеет место в случае расчета рассеянного света. Поэтому такое решение вряд ли может быть применено для анализа рассеянного света в оптических приборах.

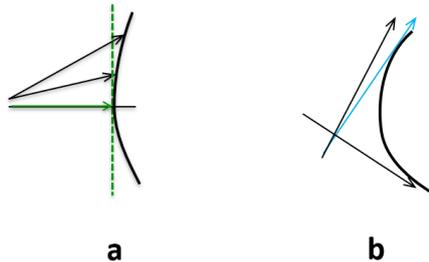


Рис. 3. Случаи корректной (а) и некорректной (б) работы метода определения точки встречи луча с поверхностью

Другим методом решения задачи определения точки пересечения луча с асферической поверхностью, широко применяемым в компьютерной графике, является трассировка лучей через геометрию, представленную в виде очень мелкой треугольной сетки. Этот метод обеспечивает достаточно точную трассировку, но является весьма затратным по объему данных и скорости трассировки. Вариант трассировки лучей с явно заданной геометрией (аналитически) гораздо более выгоден с точки зрения сокращения объема данных, но приводит к очень большим временным затратам, которые могут превышать затраты, возникающие в случае представления геометрии мелкой треугольной сеткой. Поэтому актуальна задача поиска метода, позволяющего обеспечить достаточную точность и скорость физически корректной трассировки лучей при минимальном объеме данных, необходимых для представления геометрии. Для решения этой задачи предлагается гибридная

модель, включающая как формирование треугольной сетки, так и аналитическое представление геометрии формы поверхности.

2. Точность представления геометрии, заданной треугольной сеткой, при трассировке лучей

Как было отмечено выше, в ряде случаев поиск пересечения луча с треугольной сеткой является более эффективным решением с точки зрения производительности и скорости вычислений, чем поиск пересечения луча с аналитической геометрией, заданной, например, полиномом, степень которого достигает нескольких десятков. Формирование треугольной сетки для аналитически заданной поверхности, поиск точки пересечения луча с треугольником сетки, интерполяция нормали и самой точки пересечения являются задачами, для которых уже десятки лет существуют готовые решения [7]. На рисунке 4 представлен принцип формирования результата пересечения луча с треугольной сеткой. Как правило, возвращаемый результат – нормаль и точка на поверхности.

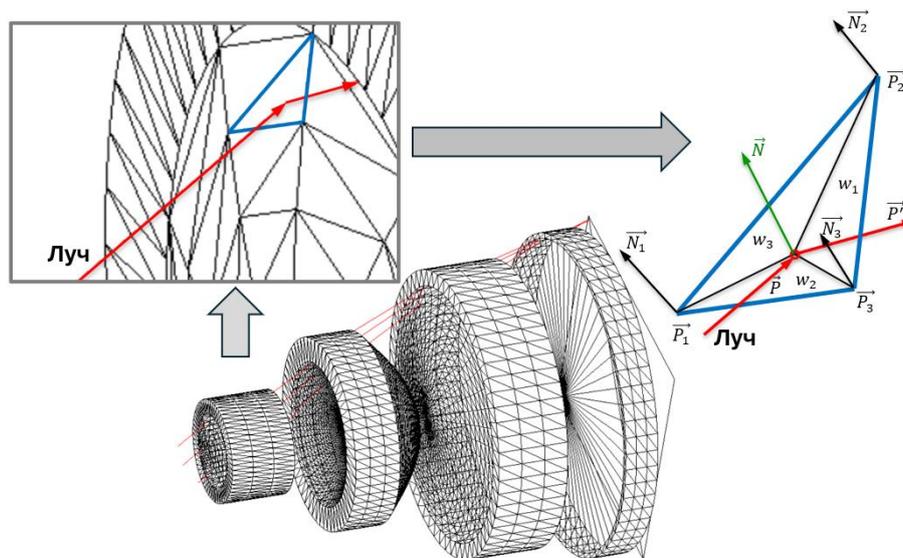


Рис. 4. Пересечение луча с треугольником сетки

Поскольку геометрия поверхности задана аналитическим образом, то истинное значение нормалей в вершинах треугольника может быть легко вычислено и нормаль в точке пересечения и сама точка пересечения могут быть интерполированы следующим образом:

$$\begin{aligned}\vec{N} &= f_N(\vec{N}_1, \vec{N}_2, \vec{N}_3, w_1, w_2, w_3), \\ \vec{P} &= f_P(\vec{N}_1, \vec{N}_2, \vec{N}_3, \vec{P}_1, \vec{P}_2, \vec{P}_3, \vec{P}, w_1, w_2, w_3),\end{aligned}$$

где: w_1, w_2, w_3 – барицентрические координаты треугольника в точке встречи луча с плоскостью треугольника \vec{P} ; $\vec{P}_1, \vec{P}_2, \vec{P}_3$ – вершины треугольника; $\vec{N}_1, \vec{N}_2, \vec{N}_3$ – истинные векторы нормалей в вершинах треугольника; f_N – функция интерполяции нормали \vec{N} в точке \vec{P} (как правило, это простое взвешивание $\vec{N} = w_1 \cdot \vec{N}_1 + w_2 \cdot \vec{N}_2 + w_3 \cdot \vec{N}_3$); f_P – функция интерполяции точки пересечения луча с поверхностью \vec{P}' (для интерполяции может использоваться поверхность второго порядка).

Данный подход хорошо работает в системах компьютерной графики, где не требуется высокая точность вычислений (до долей длины волны). Однако применение такого подхода в задачах вычислительной оптики имеет как свои преимущества, так и свои недостатки.

Основные преимущества:

- Простое решение для поиска точки встречи луча с поверхностью произвольной формы (возможно решение не только для поверхностей, заданных полиномами, но и для поверхностей свободной формы, не имеющих аналитического представления).
- Отсутствуют проблемы поиска точек встречи лучей, идущих по касательным к поверхностям или не попадающих на поверхность (эта проблема становится актуальной в задачах анализа рассеянного света в оптических системах).
- Быстрая трассировка лучей (высокая производительность может быть достигнута на CPU с использованием SIMD инструкций или на GPU).

Основные недостатки:

- Для обеспечения высокой точности вычислений требуется очень высокая степень разбиения треугольной сетки (миллионы треугольников, что приводит к увеличению RAM и ужесточению требований к используемому hardware).
- Координаты точки пересечения луча с поверхностью и нормаль в точке пересечения являются результатом интерполяции, что добавляет неточность к алгоритму трассировки лучей.
- Даже если результат трассировки лучей обеспечил корректный расчет геометрических aberrаций, нет гарантии того, что волновые aberrации были вычислены корректно.

Для оценки точности представления геометрии в виде треугольной сетки было проведено исследование зависимости точности трассировки лучей на примере расчета функции рассеяния точки (построение изображения точечного объекта методом прямой трассировки лучей) от разрешения треугольной сетки (разбивки асферической поверхности по азимутальному углу и высоте на поверхности линзы r). Исследование проводилось для двух видов асферических поверхностей, представленных формулами (2) и (3) соответственно. Для моделирования оптической системы с асферической поверхностью, представленной формулой (2), использовался трехлинзовый объектив,

показанный на рисунке 5(а), а для моделирования оптической системы с асферической поверхностью, представленной формулой (3), использовался пятилинзовый объектив, показанный на рисунке 5(б). Диаметры входных зрачков обоих объективов составляли порядка 1 мм. Для вычисления функции рассеяния точки использовались 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, в которой вершины 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].

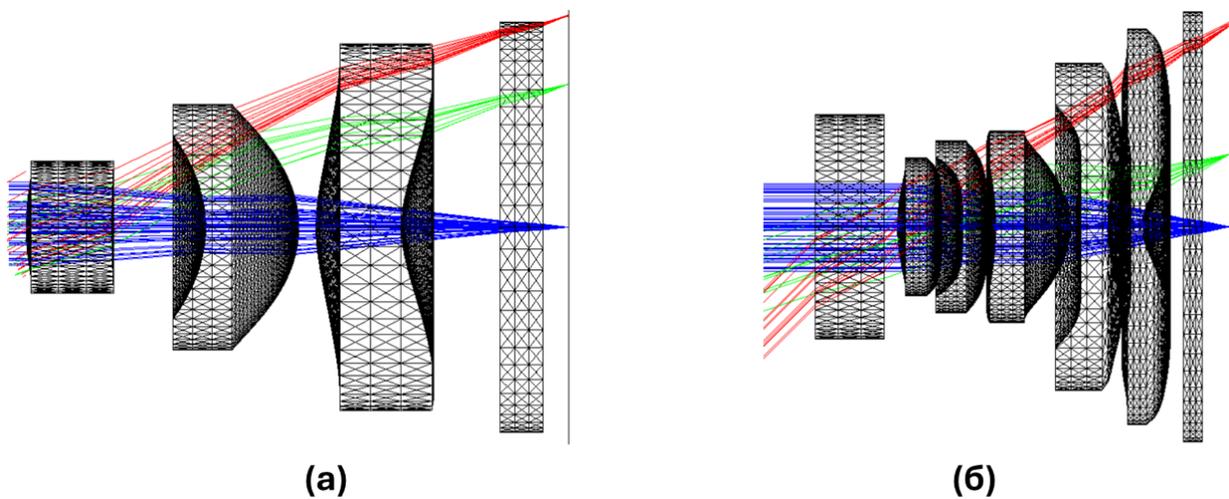


Рис. 5. Пример объективов, используемых для расчета функции рассеяния точки, (а) – трехлинзовый объектив с асферической поверхностью, определяемой уравнением (2), (б) – пятилинзовый объектив с асферической поверхностью, определяемой уравнением (3)

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 1024x1024. 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 14400 треугольников, а в максимуме, когда изображения, вычисленные по алгоритмам [4] и [1], визуально совпали с изображениями на треугольной сетке, сетка содержала 7200000 треугольников. Соответствующие изображения функции рассеяния точки

представлены на рисунке 6. Ряд (1) – изображения, вычисленные по алгоритмам [4] и [1], ряд (2) – изображения, вычисленные для сетки из 14400 треугольников, ряд (3) – изображения, вычисленные для сетки из 7200000 треугольников. Размер изображения для осевых точек обоих объективов, см. рисунок 6 (колонки а, б), составлял 5 мкм, для полевых точек обоих объективов, см. рисунок 6 (колонки в, г), – 10 мкм.

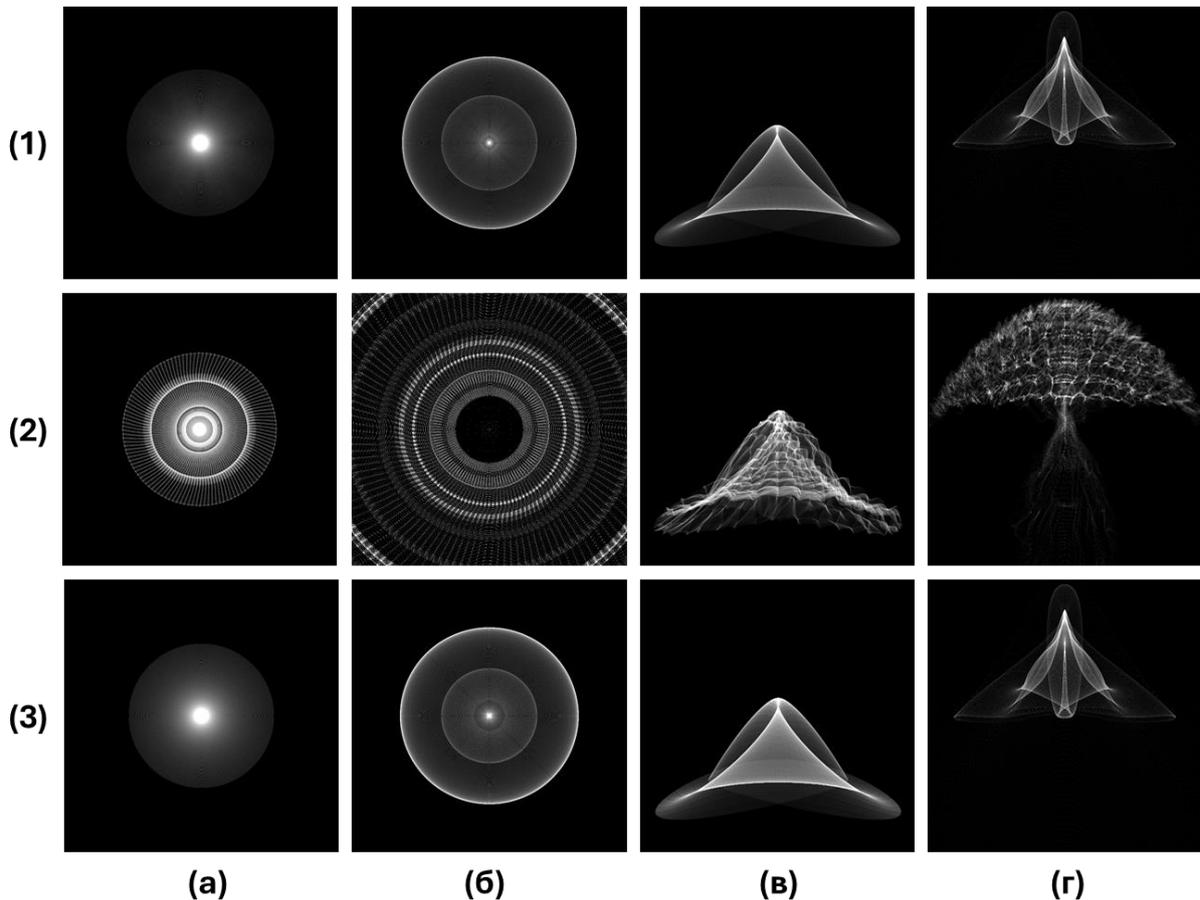


Рис. 6. Результаты моделирования функции рассеяния точки трехлинзового объектива (колонки (а) и (в)) и пятилинзового объектива (колонки (б) и (г)), Ряд (1) – вычисление по алгоритмам [4] и [1], Ряд (2) – вычисление по сетке из 14400 треугольников, Ряд (3) – вычисление по сетке из 7200000 треугольников

Видно, что использование треугольной сетки позволяет добиться корректного результата для осевой точки предмета и очень близкого к корректному – для полевых точек предмета. 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 гибридное 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”, однако данное решение нацелено не только на снижение объема данных, но и на ускорение расчета.

3. Метод двухэтапной трассировки лучей в оптических системах

Для решения задачи двухэтапной трассировки лучей был разработан специализированный трассировщик лучей, ориентированный на расчет изображений, формируемых оптическими системами. В качестве базового решения для трассировки лучей была использована библиотека Embree от компании Intel [8], работающая на центральном процессоре. Основные причины выбора API Embree следующие:

- 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 [9];
- API поддерживает специальные функции, позволяющие модифицировать геометрию поверхности во время трассировки лучей.

Учитывая специфику задач прикладной оптики, а именно: луч распространяется последовательно от одной поверхности оптической системы к следующей поверхности по порядку их задания в оптической системе – а также возможность API Embree, позволяющую на одном устройстве иметь неограниченное количество сцен, было принято решение о формировании отдельных Embree сцен для каждой поверхности оптической системы. Данное решение проиллюстрировано на рисунке 7.

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 мультikonфигурационные оптические системы. Кроме того, последовательность может быть и случайной, что позволяет моделировать такой эффект, как блики на 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).

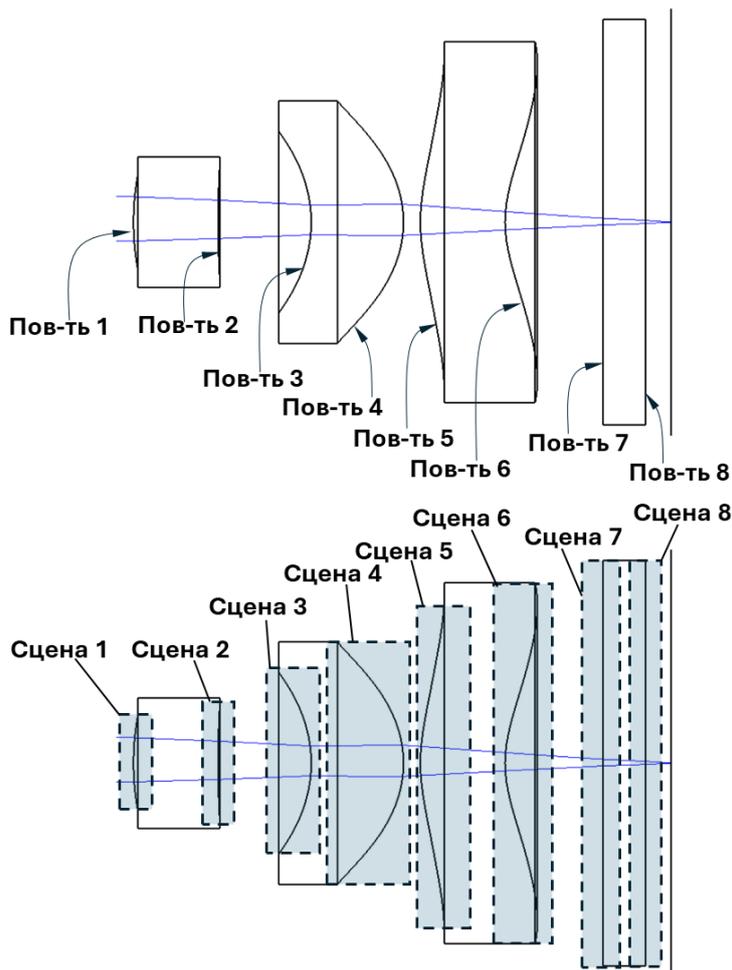


Рис. 7. Представление оптической системы в виде набора Embree сцен, сверху – исходная оптическая система, внизу – оптическая система, разбитая на ряд Embree сцен (каждой поверхности соответствует одна сцена) с граничными объемами для каждой поверхности оптической системы

Следует заметить, 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 проводится 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.

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 поверхностей 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 8, иллюстрирующем 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 новый результат как результат своего поиска.

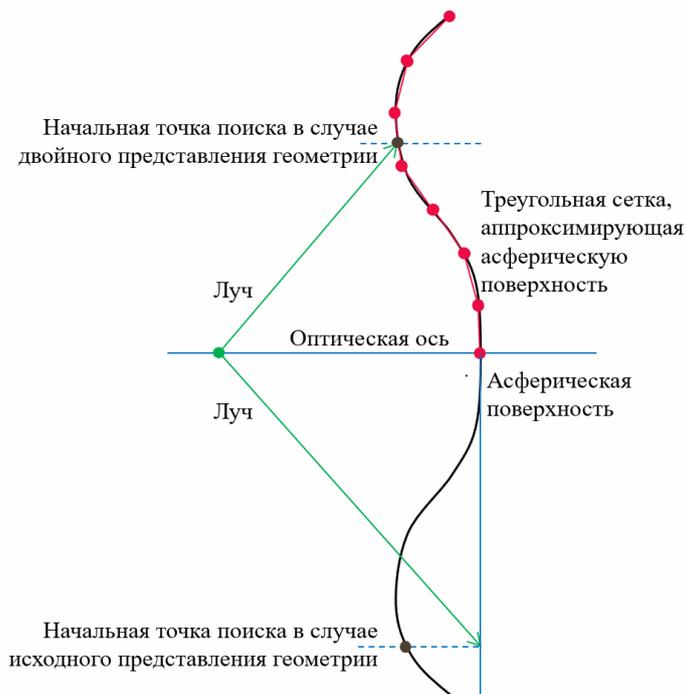


Рис. 8. Определение начальной точки встречи луча с асферической поверхностью: верхняя часть рисунка демонстрирует случай двойного представления геометрии (треугольная сетка и аналитическая формула), нижняя часть рисунка демонстрирует случай работы с одной аналитической формулой

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 луча 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.

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 очки 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 на метод 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 с двойной точностью. В результате вектор из восьми float разбивается на два вектора из четырех double. Эта процедура показана на рисунке 9. 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).

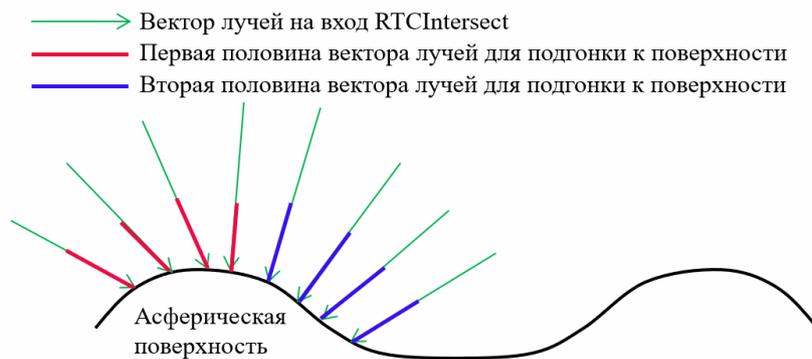


Рис. 9. Разбиение вектора из восьми лучей на выходе RTCIntersect для работы с числами двойной точности в методе подгонки лучей

Для оценки качества изображения по рассчитанной функции рассеяния точки вычисляются среднеквадратические отклонения (СКВ) относительно главного луча и относительно центра тяжести распределения лучей в полученной функции рассеяния точки (ФРТ).

Среднеквадратичное отклонение σ^2 относительно главного луча с координатой E_p на входном зрачке и P_0 в плоскости изображения определяется выражением:

$$\sigma^2 = \frac{\sum_{i=1}^{all\ image\ points} (P_0 - P_i)^2}{N}, \quad (4)$$

где N – число лучей, попавших на изображение, P_i – координаты i -го луча, попавшего на изображение. Среднеквадратические отклонения по осям x и y в плоскости изображения определяются как

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.

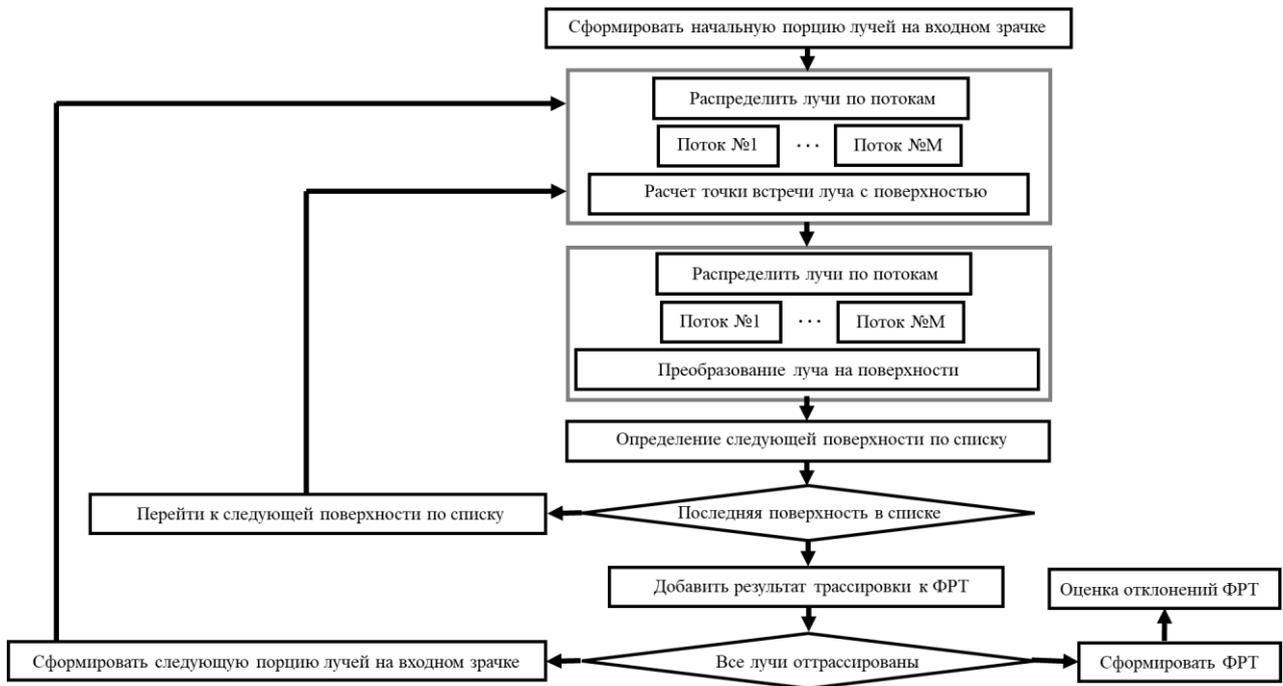


Рис. 11. Алгоритм расчета функции рассеяния точки

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

Разработанные программные решения двухуровневой трассировки лучей были реализованы в виде программных модулей расчета геометрической PSF, а также как plugin системы физически корректного рендеринга для анализа бликов, возникающих в оптических системах. 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 5. Тестирование производилось на

компьютере с восьмиядерным процессором AMD Ryzen 9 и 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 луча. Поля зрения для трехлинзового объектива составляли 0° , 26° и 30° , для пятилинзового объектива поля зрения составляли 0° , 28° и 38° . Изображения функции рассеивания точки трехлинзового объектива и оценки СКВ, полученные в результате тестирования, представлены на рисунке 12. Контур красного цвета соответствует СКВ относительно центра тяжести ФРТ, желтый контур соответствует СКВ относительно главного луча, а контур голубого цвета обозначает область диска Эри. 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 рисунках 12 и 13.

Таблица 1

Время расчета различных полей зрения для двух объективов

Трехлинзовый объектив	$\omega=0^\circ$ время расчета в мс	$\omega=26^\circ$ время расчета в мс	$\omega=30^\circ$ время расчета в мс
Двухуровневая модель	28	26	27
Модель с подгонкой	34	42	44
Пятилинзовый объектив	$\omega=0^\circ$ время расчета в мс	$\omega=28^\circ$ время расчета в мс	$\omega=38^\circ$ время расчета в мс
Двухуровневая модель	80	45	25
Модель с подгонкой	82	64	57

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 8. 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.

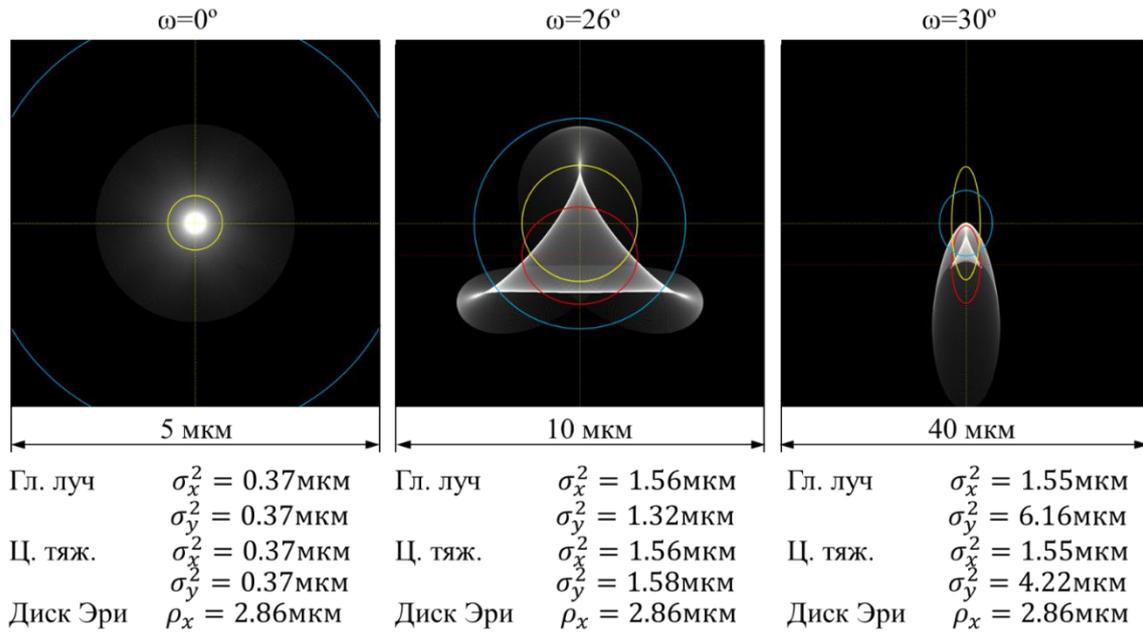


Рис. 12. Изображения и оценки ФРТ, формируемых объективом, представленным на рисунке 5а, для трех точек поля зрения

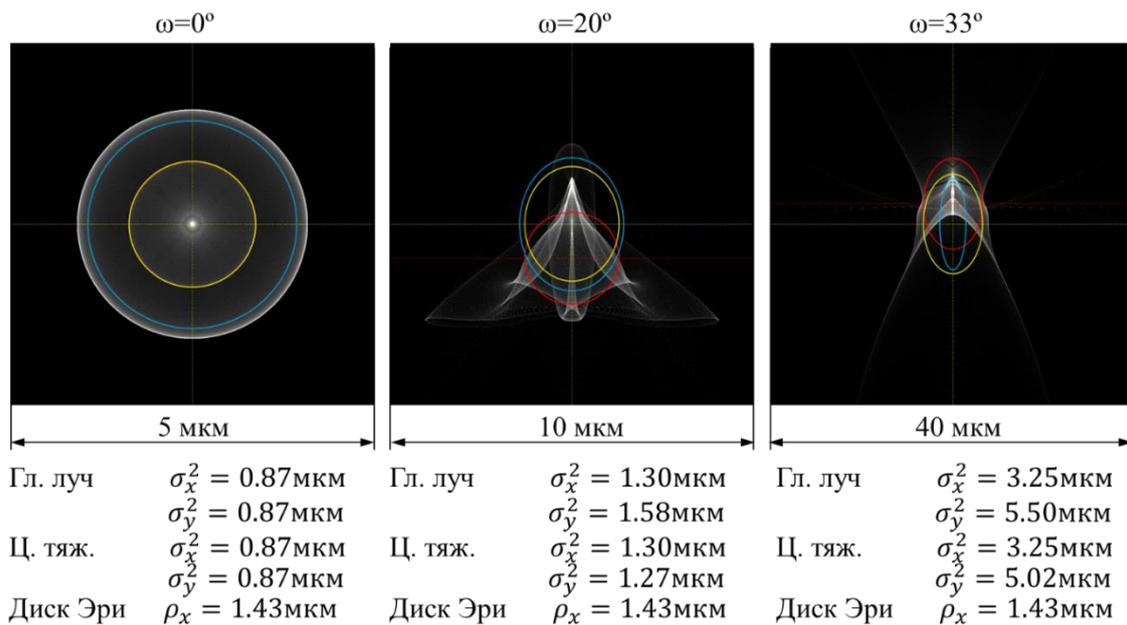


Рис. 13. Изображения и оценки ФРТ, формируемых объективом, представленным на рисунке 5б, для трех точек поля зрения

Другим примером использования данного решения является расчет бликов в линзовых системах. Для трехлинзового и пятилинзового объективов, представленных на рисунке 5а и 5б соответственно, была задана модель

оптических свойств поверхностей, соответствующая непросветленной оптике. Факт отражения или пропускания определялся вероятностным образом по коэффициенту отражения, вычисляемому по формуле Френеля. В соответствии с выбором события отражения или преломления формировался список чередования поверхностей оптической системы для трассировки лучей. Результаты моделирования бликов, вызванных бесконечно удаленными источниками засветки, расположенными вне поля зрения объективов, представлены на рисунке 14.

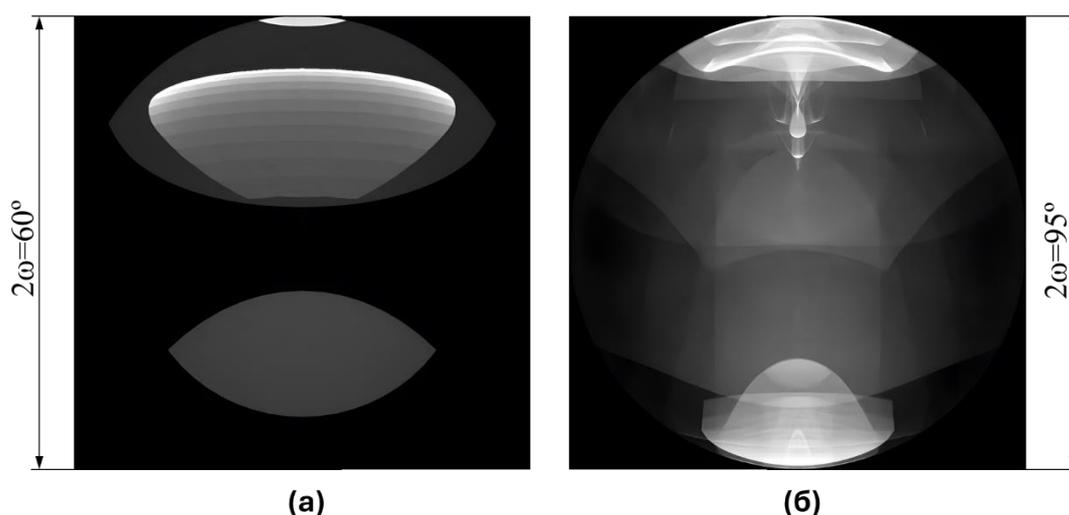


Рис. 14. Моделирование бликов от источников засветки вне поля зрения для трехлинзового объектива с углом засветки $\omega_{\text{засветки}}=31^\circ$ (а) и для пятилинзового объектива с углом засветки $\omega_{\text{засветки}}=49^\circ$ (б)

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

В рамках проведенного исследования была разработана концепция двухуровневого представления рабочих поверхностей оптической системы. Основным критерием, используемым при построении данной концепции, было улучшение процедуры поиска точки встречи луча асферическими поверхностями оптической системы, а именно ускорение, сохранение, а в ряде случаев повышение точности работы и повышение устойчивости работы для нестандартных направлений лучей. В результате:

- Для повышения скорости и устойчивости вычислений был предложен подход, основанный на двухуровневом представлении геометрических поверхностей оптических устройств.
- На основе Embree API была разработана библиотека классов, реализующая программные интерфейсы для высокопроизводительной трассировки

лучей в оптических системах, содержащих асферические поверхности высокого порядка, включая асферические поверхности, заданные полиномами Якоби.

- Программные приложения для расчета и анализа ФРТ и рассеянного света, основанные на разработанной библиотеке классов, показали высокую производительность и устойчивость вычислений к экстремальным условиям освещения.

Библиографический список

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. К. Р. 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. GitHub — embree/embree: Embree ray tracing kernels repository, URL: <https://github.com/embree/embree>
9. Intel® Threading Building Blocks Tutorial (ed.ac.uk), URL: <https://www.inf.ed.ac.uk/teaching/courses/ppls/TBBtutorial.pdf>

Оглавление

1. Введение	3
2. Точность представления геометрии, заданной треугольной сеткой, при трассировке лучей	6
3. Метод двухэтапной трассировки лучей в оптических системах.....	10
4. Тестирование разработанных алгоритмов.....	16
5. Заключение.....	19
Библиографический список.....	20