

Выбор начального приближения в трассировке лучей оптических систем с асферическими поверхностями

В. В. Санжаров¹, С. В. Ершов², В. А. Фролов^{1,2}, А. Г. Волобой²

¹Институт перспективных исследований проблем искусственного интеллекта и интеллектуальных систем МГУ им. М.В. Ломоносова, Москва, Россия

² Институт прикладной математики им. М.В.Келдыша РАН, Москва, Россия

Аннотация. В работе рассматривается проблема поиска пересечения луча с асферическими поверхностями в оптических системах. В качестве асферических поверхностей рассматривались поверхности, заданные полиномами высокого порядка. Для поиска пересечения луча с такими поверхностями обычно используется метод Ньютона, который чувствителен к начальному приближению. Предложен метод вычисления начального приближения, позволяющий повысить устойчивость расчетов пересечения луча с поверхностью. Разработанный метод позволяет с высоким уровнем точности находить пересечения с поверхностями сложной формы, в том числе для лучей, падающих под большим углом к оптической оси. В сравнении с протестированными существующими решениями предложенный метод позволяет вычислить пересечения для значительно большего числа лучей.

Ключевые слова: трассировка лучей, оптика, асферические поверхности

Initial guess in ray tracing optical systems with aspheric surface

V. V. Sanzharov¹, S. V. Ershov², V. A. Frolov^{1,2}, A. G. Voloboy²

¹IAI Moscow State University, Moscow, Russia

²Keldysh Institute of Applied Mathematics RAS, Moscow, Russia

Аннотация. The paper considers the problem of finding the intersection of a ray with aspherical surfaces in optical systems. Surfaces defined by high-order polynomials were considered as aspherical surfaces. The intersection of a ray with such surfaces is usually computed using Newton's method, which is sensitive to the initial approximation. A method for calculating the initial approximation has been proposed, which makes it possible to increase the stability of calculations of the intersection of the ray with the surface. The developed method makes it possible to find intersections with surfaces of complex shapes with a high precision, including rays incident at a large angle to the optical axis. In comparison with existing solutions, proposed method makes it possible to calculate intersections for a much larger number of rays.

Ключевые слова: ray tracing, optics, aspheric surface

Введение

Точное пересечение луча с поверхностью имеет решающее значение при трассировке лучей в оптических системах. Для несферических преломляющих поверхностей (асферических линз или оптики свободной формы/сплайна) пересечение с лучом находится путем решения нелинейного уравнения по методу Ньютона. Однако для сходимости метода Ньютона требуется хорошее начальное предположение.

Выбор подходящих начальных предположений для метода Ньютона имеет решающее значение в приложениях трассировки лучей по следующим причинам:

- надежность сходимости: плохие начальные приближения могут привести к тому, что метод не будет сходиться или будет сходиться к неправильным точкам пересечения;
- эффективность вычислений: лучшие начальные приближения обычно требуют меньшего числа итераций для достижения желаемой точности.

В приложении к оптическим системам существует несколько подходов к вычислению начального приближения для метода Ньютона. Рассмотрим существующие решения.

1. Методы на основе аналитической аппроксимации

Многие методы линеаризируют или упрощают поверхность локально, чтобы получить первое приближение. Самый простой подход – пересечение с плоскостью. В этом случае поверхность аппроксимируется плоскостью, перпендикулярной оптической оси. Это может быть касательная плоскость к оптической поверхности или же плоскость, проходящая через её центр. Производится

подстановка уравнения луча в уравнение плоскости и ищется решение. Это дает оценку t , которую можно вычислить легко и эффективно. На практике такая аппроксимация широко используется в трассировке лучей в оптических системах, в том числе в современных работах, таких как [Nie23, Wang22, Yang24].

Однако использование начального приближения, полученного таким путем, не гарантирует сходимость метода Ньютона. Для поверхностей со сложной формой и/или для лучей, идущих под большим углом, аппроксимации плоскостью может быть недостаточно (рис. 1).

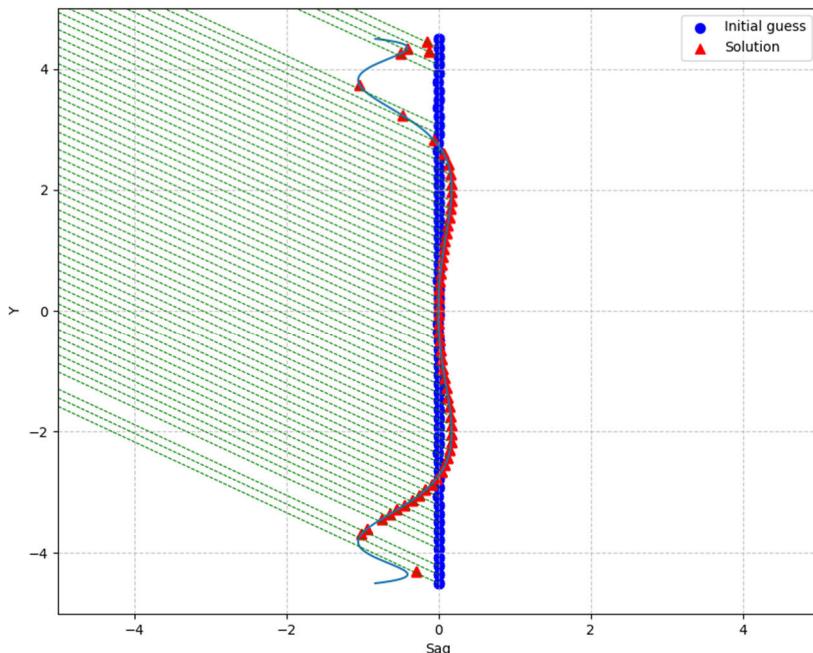


Рисунок 1. Пересечение пучка лучей с асферической поверхностью. Синими точками показаны начальные приближения для метода Ньютона, красными треугольниками – найденные решения

Если поверхность имеет перегиб или большую кривизну, то пересечение с касательной плоскостью может лежать за пределами физической апертуры, что приводит к тому, что метод Ньютона сходится к неверному решению.

Другой схожий подход – пересечение с "базовой" сферой. Многие асферические поверхности определяются как базовая коника (например, сфера или параболоид) плюс члены высших порядков. Можно найти пересечения луча с базовой сферой с заданным радиусом кривизны аналитически и использовать полученное значение t в качестве начального приближения. Такой подход может давать точку, расположенную ближе к реальной поверхности, чем в случае с аппроксимацией плоскостью. Пересечение со сферой тоже является вычислительно простым способом, так как требуется лишь решить квадратное уравнение. Несмотря на потенциально лучшее приближение для умеренно асферических поверхностей, данный подход имеет те же проблемы для сложных и сильно асферических поверхностей.

В работе [Joo16] авторы оценивают начальное приближение, шаг вдоль луча и интервал поиска пересечения, используя две аналитические поверхности, ограничивающие оптическую поверхность с обеих сторон. В качестве ограничивающих поверхностей предлагается использовать сферы или плоскости. При расчете пересечения лучей с поверхностями линз авторы отдают предпочтение методу бисекции с начальным интервалом поиска, заданным ограничивающими поверхностями. Основным недостатком метода бисекции является то, что для достижения высокого уровня точности, необходимого в оптических расчетах, ему требуется в несколько раз больше итераций, чем для метода Ньютона.

В целом, аналитические аппроксимации плоскостью или сферой просто реализовать. Они требуют минимальных вычислительных затрат, но им не хватает надежности для сложных форм – метод Ньютона может расходиться или сходиться к неправильному пересечению (например, за пределами апертуры). Как правило, такие аппроксимации работают, если поверхность достаточно близка к плоской или сферической в интересующей области.

2. Методы на основе подразбиений

Подходы из данной группы основаны на дискретизации оптической поверхности с целью поиска начального приближения для метода Ньютона. Общей идеей является аппроксимация поверхности сеткой из плоских участков, треугольников или точек. При поиске пересечения луча с поверхностью, заданной сплайнами или NURBS, часто строится грубая управляющая сетка или треугольная сетка. Лучи пересекаются с этой сеткой, чтобы найти приблизительную точку пересечения. Например, может использоваться равномерная сетка точек на поверхности, расстояние до пересечения с которой используется в качестве начального для метода Ньютона. Схожим образом можно выполнить разбиение для поверхности в параметрической области. Тогда каждый участок («патч») предоставляет потенциальный интервал, содержащий пересечение. Такие методы гарантируют, что начальное предположение лежит вблизи действительного корня при условии, что сетка достаточно мелкая. Эти методы хорошо справляются со сложными формами и множеством пересечений и применимы к поверхностям свободной формы и симметричным поверхностям. Однако для этого требуется плотная сетка с высоким уровнем подразбиения, что приводит к значительным вычислительным затратам.

В работе [Koning23] триангуляция для поиска пересечений используется для поверхностей свободной формы, моделируемых B-сплайнами. Авторы используют свойства B-сплайнов для ускорения процесса поиска пересечения лучей и треугольников. Однако эффективность предложенного подхода не исследовалась, и оптимизация предложенного алгоритма пересечения B-сплайнов оставлена для будущих работ.

Для повышения производительности поиска пересечения с представлениями поверхности в виде полигональной сетки могут также использоваться ускоряющие структуры (такие как BVH-дерево). В [Zhdanov22] такой подход предлагается для асферических поверхностей, а в [Zhdanov24] применение расширено на поверхности свободной формы и предложены дополнительные оптимизации расчетов. Пересечение с триангулированными сетками сначала дает хорошее начальное предположение, которое ускоряет сходимость, а также позволяет справиться с проблемными лучами, такими как касательные (или почти касательные) к поверхности. Однако в таких задачах, как проектирование оптических систем, когда целевая поверхность постоянно меняется в процессе оптимизации, использование такого подхода может вызвать трудности, поскольку триангулированную геометрию и ускоряющую структуру также потребуется постоянно обновлять.

Авторы работы [Morita10] предлагают аппроксимировать асферическую поверхность квадратичными интерполянтами (патчами). Пересечение между лучом и патчем может быть найдено аналитически. В работе показано, что найденное пересечение и нормаль в нем быстрее приближаются к истинным значениям с увеличением числа патчей, чем при использовании треугольников для аппроксимации поверхности. Авторы утверждают, что сходимость к пределу точности плавающей точки двойной точности достигается при использовании 10^7 патчей, в то время как для этого потребовалось бы 10^{16} треугольников. Что, однако, все равно требует достаточно большого количества вычислений. Поэтому в целом потенциальные проблемы при таком подходе аналогичны тем, что возникают при использовании треугольников.

Методы на основе подразбиений могут работать с различными представлениями поверхностей, включая поверхности свободной формы, и дают начальное предположение, близкое к истинному пересечению, если сетка достаточно плотная. Однако это приводит к увеличению как вычислительных затрат, так и затрат памяти (для хранения сетки). А для задач проектирования оптических систем, когда целевая поверхность меняется в процессе оптимизации, необходимо также обновлять и новое представление поверхности (а также ускоряющую структуру, если она используется), что может быть неэффективным.

3. Методы на основе когерентных лучей и опорных точек

При трассировке лучей в оптических системах часто необходимо рассчитывать лучи, имеющие близкие направления. Методы на основе когерентности лучей используют этот факт для повторного использования результатов: можно использовать точку пересечения для соседних лучей (или предыдущего луча) в качестве начального предположения для текущего луча [Joy86]. В плотно дискретизированных пучках (например, последовательные лучи в пространстве изображения)

соседние пересечения оказываются близки. Этот подход очень дешев поскольку повторно использует уже рассчитанный результат и может значительно ускорить поиск пересечения. Но, например, если некоторый луч пропустит поверхность, на которую попали его соседи, метод не будет работать. Поэтому наивное повторное использование предыдущего пересечения иногда приводит к сходимости к неправильному корню. Кроме того, нужно каким-то образом получить предыдущее пересечение.

Более надежным вариантом является метод на основе опорных точек. Сначала задается набор опорных точек на поверхности, который может быть получен, например, варьированием координат по достаточно грубой сетке. Затем для каждого входящего луча находится опорная точка, ближайшая к нему по евклидову расстоянию. Наконец, эта точка проецируется перпендикулярно на луч и используется как начальное приближение для расстояния пересечения [Ren25]. По сути, такой подход выпускает луч вблизи известной точки поверхности. Это гарантирует, что начальное приближение находится вблизи поверхности, и улучшает сходимость для сильно асферических поверхностей или поверхностей свободной формы, позволяя избежать сходимость к неверным корням, которые могут возникать для методов, использующих аппроксимацию плоскостью. Недостатком являются дополнительные затраты на хранение и поиск опорных точек, а также вычисление ближайших расстояний. Кроме того, в задачах проектирования оптических систем, опорные точки требуется вычислять заново при итеративном изменении поверхности.

4. Методы в пространстве параметров

Существуют специальные методы параметризации сплайновых поверхностей, позволяющие ослабить требования к начальному приближению. Например, в работе [Tang23] авторы заменяют параметры поверхности (u,v) преобразованной парой (ξ,η) , которая принадлежит интервалу $[0,1]$, даже если оценки для (u,v) выходят за его пределы. Таким образом, метод Ньютона будет сходится к нужному решению даже для начальных значений (u_0, v_0) , выбранных из достаточно грубой сетки.

Другой параметрический подход – двумерное пересечение. Вместо решения уравнения луча можно решить эквивалентную систему в (u,v) путем пересечения луча с параметрической поверхностью, спроектированной на плоскость. Это часто приводит к двум нелинейным уравнениям в (u,v) и снова требует использования метода Ньютона. В таких случаях хорошие начальные приближения для (u,v) можно получить, проецируя направление луча на область параметров поверхности или используя среднюю точку ограничивающего параллелепипеда [Martin00].

В целом, данная группа методов специфична для сплайновых и NURBS-поверхностей. Они также усложняют процедуру поиска пересечений, поскольку необходимо реализовать преобразование и обратное преобразование.

Таким образом, существуют различные подходы, которые можно разделить на несколько классов, рассмотренных выше. Наиболее распространеными для асферических поверхностей, заданных полиномами, являются методы на основе аналитической аппроксимации плоскостями или сферами. Они просты в реализации и быстро работают, но в сложных случаях не дают достаточно хорошего начального приближения для обеспечения сходимости. Методы на основе подразбиений позволяют найти близкое к решению начальное приближение, но требуют вычислительных затрат и затрат памяти на хранение новых представлений, что также ограничивает их применимость. В методах на основе опорных точек также есть затраты на вычисление и хранение вспомогательных данных, хоть и меньшие.

Интерес представляет разработка метода, который позволял бы получать начальное приближение, позволяющее надежно находить пересечения в сложных случаях, но не требовал бы хранения и вычисления дополнительных структур данных. При этом желательным свойством нового метода также является адаптивность. Для распространенных простых случаев поиска пересечения он должен требовать меньше вычислений, чем для сложных случаев (лучи, идущие почти по касательной к поверхности и т.п.).

Предлагаемое решение

1. Постановка задачи

Рассматривается задача пересечения луча, заданного как

$$o + vt,$$

где $o = (x_0, y_0, z_0)$ – координаты начальной точки;

t – величина расстояния;

$v = (v_x, v_y, v_z)$ – вектор направления с асферической поверхностью.

В качестве примера будем рассматривать представление поверхности в виде полинома с четными степенями:

$$f(r) = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2 r^2}} + \sum_{i=2}^n \alpha_i \rho^{2i},$$

$$\rho = \frac{r}{r_{norm}}$$

где $f(r)$ – функция расстояния от оптической оси до точки с координатами (x, y) : $r = \sqrt{x^2 + y^2}$;

c – кривизна;

k – коника;

α_i – коэффициенты полинома;

r_{norm} – параметр нормализации радиуса.

Для дальнейших рассуждений детали представления поверхности не будут иметь значения, мы будем оперировать только значениями функции поверхности $f(r)$ и её производной $f'(r)$. Таким образом, предложенное решение применимо и для других видов функции $f(r)$.

Запишем выражения для точки на луче $p_{ray}(t)$ при некотором конкретном значении t и соответствующему ей значению функции $z_{surf}(t)$:

$$p_{ray}(t) = o + vt = (x_{ray}(t), y_{ray}(t), z_{ray}(t)),$$

$$z_{surf}(t) = f\left(\sqrt{x_{ray}^2(t) + y_{ray}^2(t)}\right).$$

При пересечении луча с поверхностью $z_{ray}(t) \equiv z_{surf}(t)$.

Пересечение луча с поверхностью приводит к уравнению относительно t :

$$f(r_0^2 + at^2 + bt) - (z_0 + v_z t) = 0, \quad (1)$$

$$r_0^2 \equiv x_0^2 + y_0^2,$$

$$a \equiv 1 - v_z^2,$$

$$b \equiv 2(x_0 v_x + y_0 v_y).$$

Тогда итерация метода Ньютона будет иметь вид

$$t_{k+1} = t_k - \frac{f(r_0^2 + at_k^2 + bt_k) - (v_z t_k + z_0)}{f'(r_0^2 + at_k^2 + bt_k)(2at_k + b) - v_z}.$$

Целью данной работы является разработка метода для вычисления начального приближения $t_{k=0}$, достаточно близкого к искомому t , чтобы обеспечить сходимости итераций Ньютона к искомому решению.

2. Предложенный метод

Часто оценка снизу для расстояния до пересечения предполагает, что поверхность достаточно гладкая, и исходит из оценок Липшица [Ban25], Харнака [Gillispie24]. Такой подход работает для определения поверхности через неявную функцию, поэтому он не может быть применен к явному определению поверхности в решаемой задаче «как есть». Предложенный подход использует базовые идеи из этих работ, адаптируя их к случаю гладкой и вращательно-симметричной поверхности. Все корни уравнения (1) являются точками пересечения. Для удобства будем считать, что начальная точка равна $t=0$. Поделим интервал r^2 на некоторые точки R_i^2 , на каждом отрезке $[R_i^2, R_{i+1}^2]$ вычисляем и запоминаем границы f' в этом интервале:

$$f'_{min,i} \leq f'(r^2) \leq f'_{max,i} \text{ для } r^2 \in [R_i^2, R_{i+1}^2].$$

Теперь найдем интервал, содержащий r_0^2 . Его индекс обозначим i , то есть $r_0^2 \in [R_i^2, R_{i+1}^2]$. Тогда

$$f(r_0^2) + (r^2 - r_0^2)f'_{min,i} \leq f(r^2) \leq f(r_0^2) + (r^2 - r_0^2)f'_{max,i} \quad (2)$$

для $r^2 \in [R_i^2, R_{i+1}^2]$. Подставив в уравнение (1), мы увидим, что корень t лежит в интервале положительных корней двух квадратных уравнений

$$\begin{aligned} f(r_0^2) + (at^2 + bt)f'_{min,i} - (z_0 + v_z t) &= 0, \\ f(r_0^2) + (at^2 + bt)f'_{max,i} - (z_0 + v_z t) &= 0. \end{aligned} \quad (3)$$

На рисунке 2 показан график (синим цветом) изменения разности между координатой z_{ray} точки на луче и координатой соответствующей точки на поверхности z_{surf} от значения искомого параметра расстояния t .

Однако неравенство (2) справедливо только для $r^2 \in [R_i^2, R_{i+1}^2]$, на рисунке 2 оно не выполняется, при $r^2 \in r_0^2 + at^2 + bt = R_{i+1}^2$. Соответствующая область (криволинейный треугольник) заштрихована желтым. Корень для второго уравнения из (3) находится вне этой области, поэтому его нельзя использовать для предсказания корня t . Другими словами, мы должны использовать уравнение (3) следующим образом:

1. Вычислить граничное значение t_c для допустимой области, т.е. так, что $r_0^2 + at_c^2 + bt_c$ покидает $[R_i^2, R_{i+1}^2]$.

2. Для каждого из двух квадратных уравнений (3) вычислить их корни и взять наименьший положительный. Отсутствие положительных корней говорит о том, что парабола не пересекает ось. В этом случае мы принимаем в качестве корня бесконечность. Таким образом, имеем два значения (по одному для каждого из уравнений), которые обозначим через t_1 (меньшее) и t_2 (большее).

- a. Если ни одно из этих значений не лежит в допустимой области, тогда нижняя граница искомого пересечения $t = t_c$, и мы сдвигаем начальную точку луча в t_c .
- b. Если только одно из значений лежит в допустимой области (очевидно это t_1), тогда нижняя граница искомого пересечения $t = t_1$, и мы сдвигаем начальную точку луча в t_1 .
- c. Если оба значения лежат в допустимой области, то искомое пересечение лежит между ними, т.е. $t_1 \leq t \leq t_2$, и мы сдвигаем начальную точку луча в t_2 .

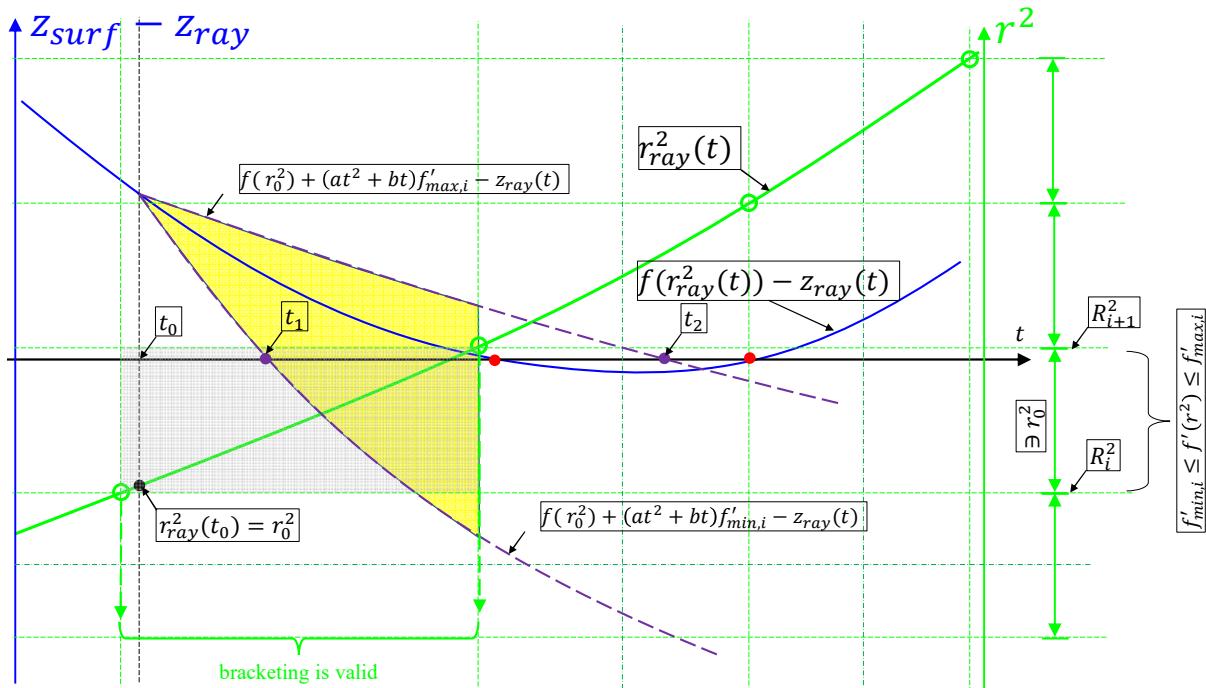


Рисунок 2. Границы корней уравнения (1), вычисленные из двух квадратных уравнений относительно $\min f'$ и $\max f'$. Красными точками отмечены корни уравнения (1), а фиолетовыми точками — положительные корни уравнения (3), по горизонтальной оси — параметр расстояния t , по левой вертикальной оси — разница между значением координаты z для точки на луче z_{ray} и для поверхности z_{surf} , по правой — значение r^2

Это очень похоже на один шаг марширования по лучу. Как и там, после шага мы перемещаем начало луча в новую точку и повторяем процедуру.

Частота подразбиения области r^2 на интервалы $[R_i^2, R_{i+1}^2]$, в которых мы берем границы f' , играет двойную роль:

- с одной стороны, чем чаще подразбиение, тем ближе (2), то есть стороны желтого треугольника на рисунке 2 проходят ближе к искомому $f(r_0^2 + at^2 + bt)$, и интервал для пересечения более плотный;
- с другой стороны, при более частом подразбиении допустимая область становится меньше, поэтому может случиться так, что $r_0^2 + at^2 + bt$ покинет ее до того, как одна из парабол в (3) пересечет ось. В этом случае границы f' вообще не используются, и все, что мы можем сделать, это переместить луч на границу допустимой области t_c . Такое перемещение может быть слишком мало.

На рисунке 3 показана ситуация для двух уровней подразбиения (сегменты для второго уровня в два раза меньше).

Для второго уровня минимальный положительный корень (3), показанный фиолетовым крестиком на рисунке 3, больше, чем таковой для более грубого уровня (фиолетовая точка). Соответственно, мы могли бы подвинуть начальную точку луча дальше, если бы не ограничение, – этот корень для второго уровня лежит вне допустимой области (оранжевый треугольник). Таким образом, второй уровень подразбиения позволит нам подвинуть начало луча только до границы допустимой области, что все равно дальше, чем первый уровень. То есть в данном примере более мелкое подразбиение дает лучший результат, но при других параметрах более грубое подразбиение может быть лучше. На практике мы использовали 4-го уровня подразбиения. Для каждого из уровней вычисляется допустимый шаг по лучу и выбирается наибольший.

Луч от объекта в оптической системе может стартовать с некоторой произвольно удаленной точки и выполнять движение по лучу по описанному алгоритму в этом случае будет очень неэффективно. Поэтому необходимо выбрать начальное значение, с которого следует запускать алгоритм. Для этого предлагается использовать простой и быстрый метод на основе ограничивающих плоскостей в качестве предварительного этапа для предложенного метода (рис. 4).

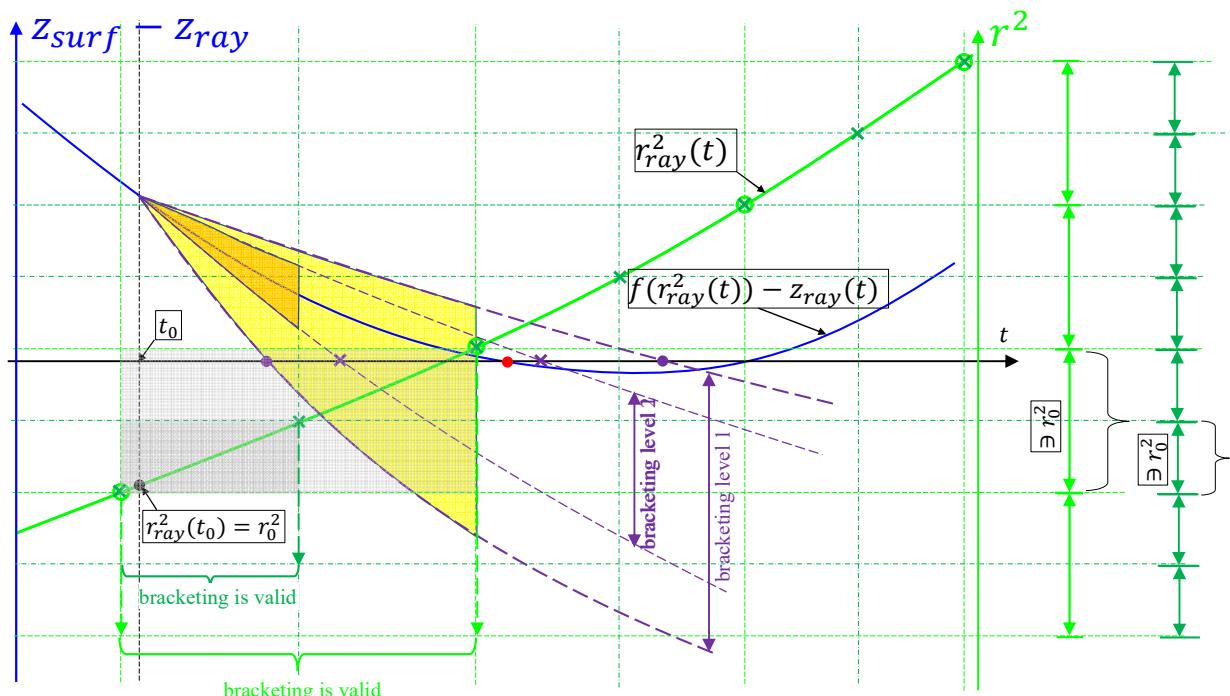


Рисунок 3. Два уровня подразбиения. Две параболы (3) для первого уровня показаны фиолетовыми толстыми пунктирными линиями, а для второго уровня – фиолетовыми тонкими пунктирными линиями. Допустимая область для второго уровня – оранжевый криволинейный треугольник. По горизонтальной оси – параметр расстояния t , по левой вертикальной оси – разница между значением координаты z для точки на луче z_{ray} и поверхности Z_{surf} , по правой – значение r^2

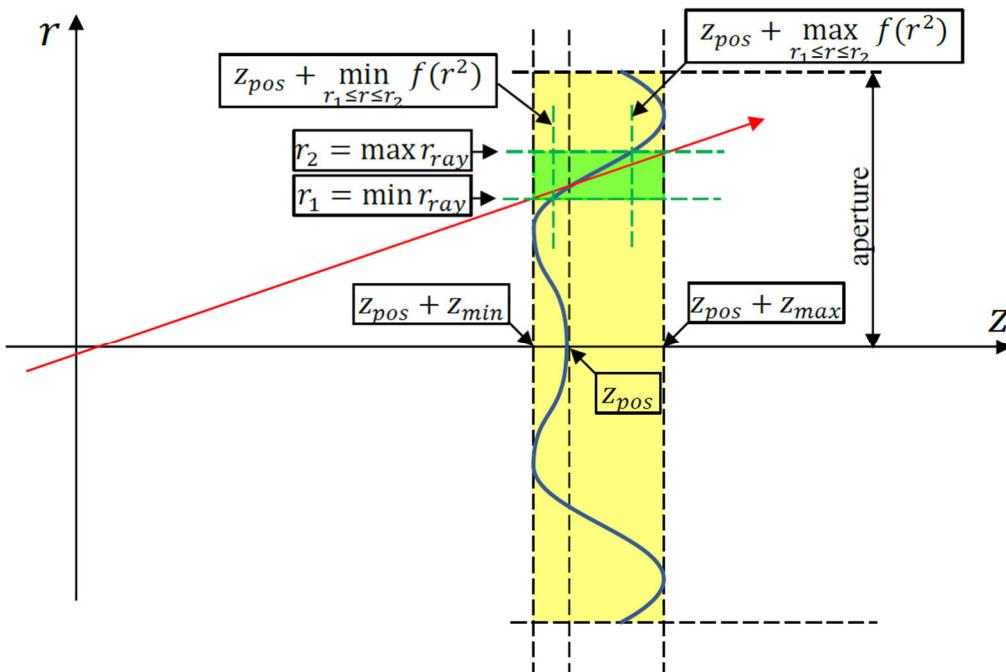


Рисунок 4. Получение начальной точки для основного алгоритма. Луч (красная стрелка) направлен на поверхность линзы (синяя кривая) издалека. Ограничивающие плоскости показаны вертикальными пунктирующими линиями вместе с центральной плоскостью. Зеленая область показывает интервал, в котором лежит пересечение

В области, ограничивающей поверхность (обозначена желтым на рис. 4), значение $r_{ray}(t)$ лежит в $[r_1, r_2]$. Для $r_1 \leq r \leq r_2$ значения функции поверхности лежат в интервале, меньшем чем $[z_{min}, z_{max}]$. Этот интервал отмечен на рисунке 4 зелеными вертикальными пунктирующими линиями. Основываясь на том, что поверхность радиально симметрична и меняется достаточно плавно в интервале допустимых значений радиуса, значения z_{min} и z_{max} могут быть вычислены как минимум и максимум значений функции поверхности f в N точках, равномерно расставленных от 0 до максимального значения радиуса r_{max} . В проведенных экспериментах использовалось $N = 100$ точек, а для вычисленных значений был также добавлен небольшой «зазор» в 0.1 % от величины найденного интервала:

$$\delta = 0.001 * (z_{max} - z_{min}), \\ z_{min} - \delta, z_{max} + \delta.$$

Алгоритм определения начальной точки следующий:

- 1) вычислить пересечения луча t_1, t_2 с ограничивающими плоскостями;
- 2) вычислить интервал $[r_1 = \min(r_{ray}(t_1), r_{ray}(t_2)), r_2 = \max(r_{ray}(t_1), r_{ray}(t_2))]$;
- 3) вычислить интервал $[z_1 = z_{pos} + f(r_1), z_2 = z_{pos} + f(r_2)]$;
- 4) вычислить пересечения луча t'_1, t'_2 с плоскостями $z = z_1$ и $z = z_2$.

Если угол между лучом и осью небольшой, то $t'_1 \approx t'_2$ и точка на луче для t'_1 лежит достаточно близко к поверхности. В общем случае можно ещё приблизиться к поверхности, повторив описанный алгоритм, взяв на первом шаге значения t'_1, t'_2 . Для больших углов луча с осью Oz итеративный процесс может начать расходиться, поэтому необходимо остановиться, если на очередной итерации интервал значений t увеличивается.

Экспериментальная оценка

Чтобы протестировать предложенный метод поиска начального приближения, были сгенерированы лучи с известными точными пересечениями. Для генерации лучей использовался следующий алгоритм:

1. Установить количество точек на поверхности выборки, количество значений для двух углов φ и θ и расстояний от поверхности t .

- a. Точки равномерно распределены по половине поперечного сечения исследуемой поверхности (так как она симметрична) и не выходят за пределы допустимого радиуса или апертуры.
- b. Угол φ равномерно выбирается от 0 до заданного максимального значения (например, 60 градусов).
- c. Угол θ выбирается равномерно от 0 до π .
- d. Расстояние выбирается равномерно от небольшого расстояния до поверхности (например, 1e-3) до максимально допустимого $r * 3$.
2. Для каждой возможной комбинации точек, углов и расстояний мы вычисляем ray_origin и ray_dir следующим образом:
- $dir1 = (\cos\varphi * \sin\theta, \sin\varphi * \sin\theta, \cos\theta);$
 - $dir2 = (-\cos\varphi * \sin\theta, -\sin\varphi * \sin\theta, \cos\theta);$
 - $ray_{dir} = concat(dir1, dir2);$
 - $ray_{origin} = sampled_{pt} - sampled_{dist} * ray_{dir}$, где $sampled_{pt}$ – выбранная точка на поверхности, $sampled_{dist}$ – выбранное расстояние от поверхности.
3. Ищем пересечение этих лучей с поверхностью:
- Если для некоторого луча пересечение не найдено, считаем поиск пересечения неуспешным и увеличиваем счетчик ненайденных пересечений.
 - Если пересечение найдено, но отличается от истинного значения, то это может означать, что на луче между его началом ray_{origin} и точкой поверхности, от которой был построен луч, располагаются другие пересечения (рис. 5). Поэтому мы передвигаем начальную точку луча в найденную точку пересечения и повторяем поиск. Этот процесс повторяется до ста раз.
 - Если мы нашли пересечение, соответствующее истинной точке на поверхности, относительно которой строился луч, то считаем поиск успешным.

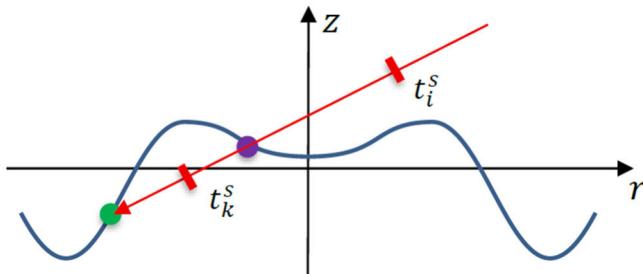


Рисунок 5. Ситуация нескольких пересечений с поверхностью (синяя линия) для сгенерированного тестового луча (красная линия). Как t_i^s и t_k^s обозначены разные выбранные расстояния и, соответственно, разные начальные точки луча. Для t_k^s будет сразу найдена точка пересечения (зеленая точка), из которой был построен луч и которая рассматривается тестом как истинная. Для t_i^s сначала будет найдено другое ближайшее к ней пересечение (фиолетовая точка), поэтому при тестировании будет произведена повторная трассировка луча с тем же направлением из этой точки

Был проведен эксперимент со следующими параметрами описанного алгоритма генерации лучей: 101 точка на поверхности, 21 значение угла φ от 0 до 60 градусов, 21 значение угла θ от 0 до π , 21 значение расстояний до поверхности от 1e-3 до $r * 3$. Такие значения параметров дают набор из 1870722 лучей. Набор лучей был сгенерирован для каждой из тестируемых поверхностей. Использовался набор из 978 асферических поверхностей, для которых была выполнена трассировка сгенерированных наборов лучей с тремя разными способами выбора начального приближения:

- на основе ограничивающих плоскостей (широко используется в существующих работах (см. подраздел «Методы на основе аналитической аппроксимации» во введении));
- на основе опорных точек на поверхности [Ren25];
- предложенный метод.

На рисунке ниже представлены примеры объективов, поверхности из которых использовались для тестирования (рис. 6).

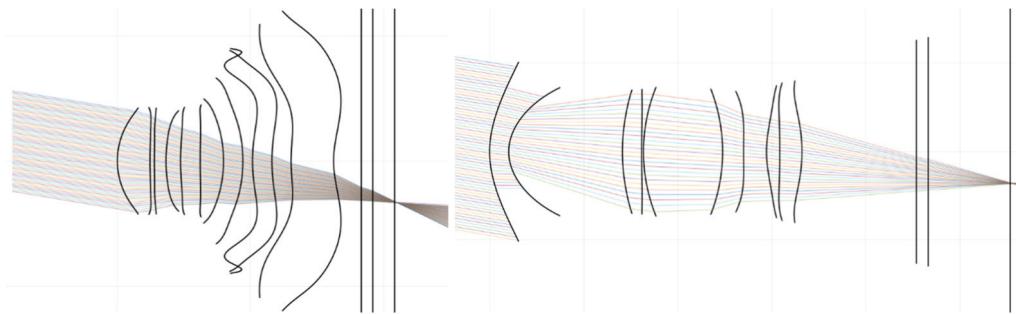


Рисунок 6. Примеры объективов с асферическими поверхностями, на которых проводилось тестирование

Для метода опорных точек [Ren25] использовалась реализация авторов, которая была перенесена с языка Python и библиотеки pytorch на C++. В качестве параметров в данном методе выступает разрешение сетки опорных точек на поверхности. Использовалось значение, выбранное авторами значением по умолчанию – 31 по x и y с удалением точек, лежащих вне апертуры.

Результаты эксперимента приведены в таблице 1. Как видно из них, предложенный метод позволяет уменьшить количество ненайденных пересечений на порядок по сравнению с двумя другими рассмотренными методами. При этом максимальное количество ненайденных лучей также остается небольшим, порядка ~0.51%. В тоже время, метод на основе ограничивающих плоскостей и метод на основе опорных точек в отдельных случаях могут терять существенное количество пересечений – больше 5%, что для рассмотренного набора лучей составляет более 100 тыс. лучей.

Таблица 1. Тестирование поиска пересечения

Метод	Средний процент ненайденных лучей	Среднее количество ненайденных лучей	Наибольший процент ненайденных лучей	Наибольшее количество ненайденных лучей
Ограничивающие плоскости	0.25%	4676	6.12%	114576
Опорные точки [Ren25]	0.17%	3164	5.44%	101808
Предложенный метод	0.013%	237	0.51%	9576

Конечно, предложенный тест является синтетическим и демонстрирует значительно более сложную задачу поиска пересечений, нежели в типичном случае последовательной трассировки лучей через систему линз. Но и в обычных сценариях последовательной трассировки лучей через оптическую систему могут встречаться случаи, когда у более простых подходов возникают сложности с поиском пересечения. Ниже приведен пример одной из поверхностей сложной формы с визуализацией начальных приближений для каждого из методов и найденного решения (рис.7, рис. 8). Видно, что приближение на основе плоскости в этом случае очень грубое и приводит к тому, что для некоторых лучей решение не найдено. Метод на основе опорных точек уже лучи не теряет, но находит одно неверное решение.

Предложенный метод находит все пересечения, при этом все начальные приближения также располагаются близко к истинным решениям.

Были проведены отдельные эксперименты и для последовательной трассировки лучей. В этом случае отличия между разными методами выбора начальных приближений обычно сравнительно небольшие и проявляются при больших углах наклона.

Также были проведены замеры производительности. Для этого выполнялась трассировка пучка из 262144 лучей (512 на 512) для того же набора поверхностей. Замеры выполнялись на CPU Ryzen 9 5950X. Результаты представлены в таблице 2.

Предложенный метод ожидаемо работает в несколько раз медленнее, чем простой метод на основе ограничивающих плоскостей. При этом время выполнения может достаточно сильно варьироваться в зависимости от поверхности, что связано с тем, что у метода нет фиксированного количества итераций. Для простых случаев, таких как пучок лучей под углом 0° к оптической оси, предложенный

метод работает в только в 2-3 раза медленнее метода на основе плоскостей. А для более сложных случаев может быть выполнено достаточно большое количество шагов, что приводит к падению производительности.

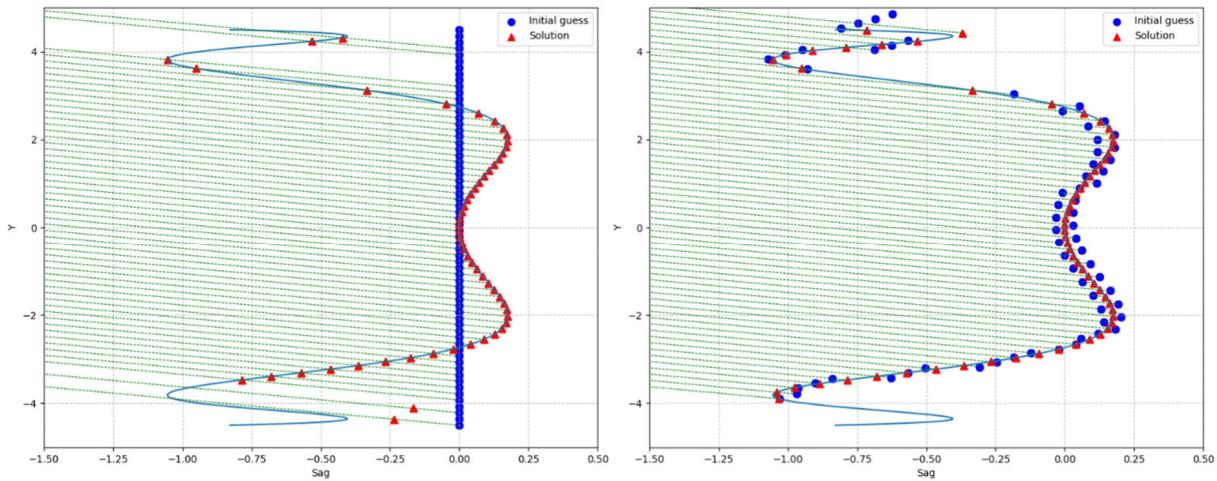


Рисунок 7. Визуализация начального приближения (синие точки) и найденного решения (красные треугольники): слева – метод на основе плоскости, справа – на основе опорных точек

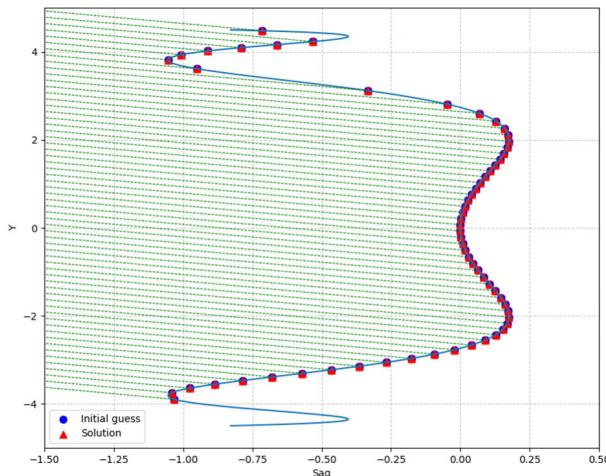


Рисунок 8. Визуализация начального приближения (синие точки) и найденного решения (красные треугольники) для предложенного метода

Таблица 2. Замеры производительности

Метод	Среднее время для пучка 0°, миллисекунды	Среднее время для пучка 20°, миллисекунды	Среднее время для пучка 40°, миллисекунды
Ограничивающие плоскости	28.39 ± 4.39	32.07 ± 6.46	36.31 ± 10.89
Опорные точки [Ren25]	446.21 ± 5.88	456.39 ± 4.96	463.26 ± 13.67
Предложенный метод	57.34 ± 20.48	353.86 ± 174.04	413.30 ± 261.82

Метод на основе опорных точек в проведенном эксперименте работает стабильно медленно, вне зависимости от угла наклона лучей. Это связано с тем, что в любом случае выполняется поиск ближайшей опорной точки к лучу. Вероятно, у этого метода имеется некоторый потенциал ускорения при реализации более эффективного метода поиска ближайшей опорной точки. Также следует отметить, что у этого метода имеется ещё один недостаток – необходимость вычислять и хранить опорные точки для поверхности. А в случае выполнения оптимизации оптической системы точки потребуется вычислять заново при каждом изменении поверхности.

Заключение

Предложенный метод выбора начального приближения для задачи поиска пересечения луча и асферической поверхности позволяет обеспечить нахождение пересечения с высокой точностью в большем числе случаев, чем наиболее распространённый метод на основе ограничивающих плоскостей и метод на основе опорных точек [Ren25]. В разработанном тестовом сценарии предложенный метод смог найти на порядок большее количество пересечений, максимальное количество потерянных пересечений не превышает 0.51 %. При этом производительность предложенного метода сравнима с методом на основе ограничивающих плоскостей в простых случаях. В более сложных случаях эффективный поиск пересечений требует большего числа итераций и тем самым большего времени работы. При этом производительность предложенного метода сохраняется на уровне метода на основе опорных точек, который, судя по проведенным экспериментам, находит меньше пересечений. Таким образом, предложенный метод применим в задачах, где необходимо моделировать случаи сложного распространения света в оптических системах, что важно в таких задачах, как, например, анализ рассеянного света.

Также следует отметить, что разработанный метод требует лишь возможности вычисления функции поверхности и её производной и тем самым может быть потенциально использован не только для асферических поверхностей, заданных полиномами четных степеней, но и для других способов задания оптических поверхностей, к которым применим метод Ньютона.

Источник финансирования

Исследование выполнено за счет гранта Российского научного фонда № 25-11-00054 «Спектральное моделирование распространения света и спектральный рендеринг в реальном времени на гибридных и мобильных платформах», <https://rscf.ru/project/25-11-00054>

Список литературы

1. [Wang22] Wang, Congli, Ni Chen, and Wolfgang Heidrich. "do: A differentiable engine for deep lens design of computational imaging systems". IEEE Transactions on Computational Imaging 8 (2022): 905-916.
2. [Yang24] Yang, Xinge, Qiang Fu, and Wolfgang Heidrich. "Curriculum learning for ab initio deep learned refractive optics". Nature communications 15.1 (2024): 6572.
3. [Heemels24] Heemels, Alexander, et al. "Optimizing freeform lenses for extended sources with algorithmic differentiable ray tracing and truncated hierarchical B-splines". Optics Express 32.6 (2024): 9730-9746.
4. [Nie23] Nie, Y., Zhang, J., Su, R., & Ottevaere, H. (2023). Freeform optical system design with differentiable three-dimensional ray tracing and unsupervised learning. Optics Express, 31(5), 7450-7465.
5. [Yang24Diff] Yang, Xinge, et al. "End-to-end hybrid refractive-diffractive lens design with differentiable ray-wave model". SIGGRAPH Asia 2024 Conference Papers. 2024.
6. [Tang23] Tang, Haisong, et al. "Parallel ray tracing through freeform lenses with NURBS surfaces". Chinese Optics Letters 21.5 (2023): 052201.
7. [Zhdanov22] Zhdanov, Dmitry, et al. "Use of two-level geometry representation for the ray tracing acceleration in optical systems with aspheric surfaces". Optical Design and Testing XII. Vol. 12315. SPIE, 2022.
8. [Zhdanov24] Zhdanov, D. D., I. S. Potemin, and A. D. Zhdanov. "Dual Representation of Geometry for Ray Tracing Acceleration in Optical Systems with Freeform Surfaces". Programming and Computer Software 50.3 (2024): 273-282.
9. [Martin00] Martin, W., Cohen, E., Fish, R., & Shirley, P. "Practical ray tracing of trimmed NURBS surfaces". Journal of Graphics Tools 5. 1. 2000: 27-52.
10. [Joo16] Joo, Hyuntae, et al. "Efficient ray tracing through aspheric lenses and imperfect bokeh synthesis". Computer Graphics Forum. Vol. 35., no. 4. 2016.
11. [Joy86] Joy, K. I., & Bhetanabhotla, M. N. (). Ray tracing parametric surface patches utilizing numerical techniques and ray coherence. ACM SIGGRAPH Computer Graphics, 20. 4. 1986: 279-285.
12. [Ren25] Ren, Z., Zhou, J., Zhang, W., Yan, J., Chen, B., Feng, H., & Chen, S. Successive optimization of optics and post-processing with differentiable coherent psf operator and field information. IEEE Transactions on Computational Imaging. 2025.
13. [Gillespie24] Gillespie, M., Yang, D., Botsch, M., & Crane, K. Ray Tracing Harmonic Functions. ACM Trans. Graph 43. 4. 2024.
14. [Ban25] Ban, R., & Valasek, G. Generalized Lipschitz Tracing of Implicit Surfaces. In Computer Graphics Forum. 2025, February. Vol. 44, no. 1. P. e15202.