

Трассировка лучей в САПР Компас-3D

С.А. Силифонкин
Самарский национальный исследовательский университет
им. академика С.П. Королёва
Самара, Россия
sergei.silifonkin@yandex.ru

С.А. Дегтярев
Самарский национальный исследовательский университет
им. академика С.П. Королёва
Самара, Россия;
Институт систем обработки изображений - филиал ФНИЦ
«Кристаллография и фотоника» РАН, Самара, Россия
sealek@gmail.com

Аннотация — В статье демонстрируется использование программного интерфейса САПР Компас-3D для получения доступа к возможностям геометрического ядра системы, позволяющим производить трассировку лучей при прохождении их через поверхности и элементы сложных форм.

Ключевые слова — программный интерфейс, трассировка лучей, нормаль, объект, поверхность

I. ВВЕДЕНИЕ

Трассировка лучей требует анализа поверхностей, через которые проходят эти лучи. Метод трассировки лучей заключается в том, что не учитывается волновая природа света [1]. Тем не менее, для построения траектории хода лучей необходимо знать координаты точки пересечения луча с поверхностью и направляющий вектор нормали к поверхности в этой точке.

При симуляции трассировки через малое количество поверхностей, легко описываемых аналитически (например, плоскости, сферы, эллипсоиды и конусы), серьёзных проблем не возникает: возможно найти точки пересечения лучей и поверхностей – а также нормали в этих точках – при помощи решения уравнений. При усложнении задачи такой подход может оказаться тупиковым. Например, при исследовании отражения лучей от неаналитической кривой поверхности, либо при составлении сложных оптических систем.

В данной работе предложено решение данной проблемы на основе использования программного интерфейса САПР Компас-3D.

II. МАТЕМАТИЧЕСКИЕ ОСНОВЫ ТРАССИРОВКИ ЛУЧЕЙ

Трассировка лучей основана на законах геометрической оптики, из которых следуют векторные формулы, используемые для расчёта направляющих векторов отражённого и преломлённого лучей.

Нахождение направляющего вектора отражённого и преломлённого луча, соответственно, находится по формулам [2]

$$\vec{e}' = \vec{e} - 2(\vec{e}, \vec{n})\vec{n}, \quad (1)$$

$$n_2 \vec{e}' = n_1 \vec{e} - (n_1 \vec{e}, \vec{n})\vec{n} \left(1 - \sqrt{\frac{n_2^2 - n_1^2}{(n_1 \vec{e}, \vec{n})^2} + 1} \right), \quad (2)$$

где \vec{e}' – вектор направления отражённого, либо преломлённого луча; \vec{e} – вектор направления падающего (начального) луча; \vec{n} – нормализованный вектор нормали к поверхности, с которой происходит взаимодействие, в

точке её пересечения с лучом; n_1 – показатель преломления среды, в которой распространяется падающий луч; n_2 – показатель преломления среды, в которой распространяется преломлённый луч.

Каждый луч имеет свой оптический путь, равный произведению пройденного геометрического расстояния на оптическую плотность среды, в которой оно пройдено. Энергия лучей вычисляется по формулам Френеля.

III. АЛГОРИТМ ТРАССИРОВКИ ЛУЧЕЙ

Подготовленная программа реализована на языке Python 3 с использованием библиотеки NumPy для реализации векторных операций и СОМ-интерфейса для подключения к программному интерфейсу. Алгоритм использует возможности геометрического решателя системы Компас-3D для получения информации о поведении света в построенной пользователем оптической системе на основе алгоритма трассировки лучей. Ниже представлена схема работы алгоритма.

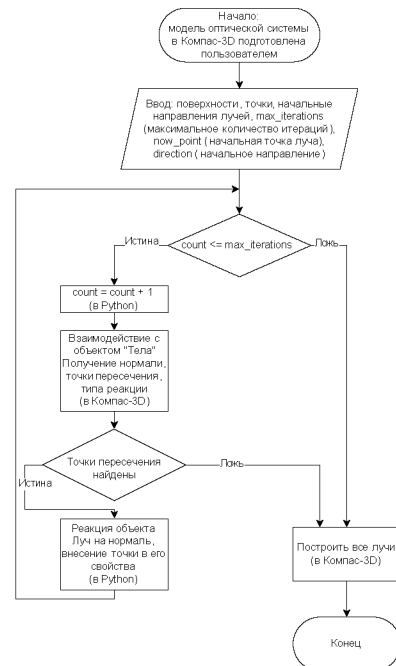


Рис. 1. Логическая схема трассировки одного луча.

На рисунке выше $count$ обозначает количество уже случившихся взаимодействий луча с построенными поверхностями, а $max_iterations$ – максимально допустимое количество этих взаимодействий. Данная переменная предусмотрена на случай, если луч заикнется в каком-нибудь месте. Этой переменной можно управлять и увеличивать для сложных оптических систем. После того, как рассчитаны траектории всех

лучей, их можно построить в 3D-редакторе программы Компас.

Ниже представлены примеры обработки сложных поверхностей и состоящих из них элементов при помощи данной программы.

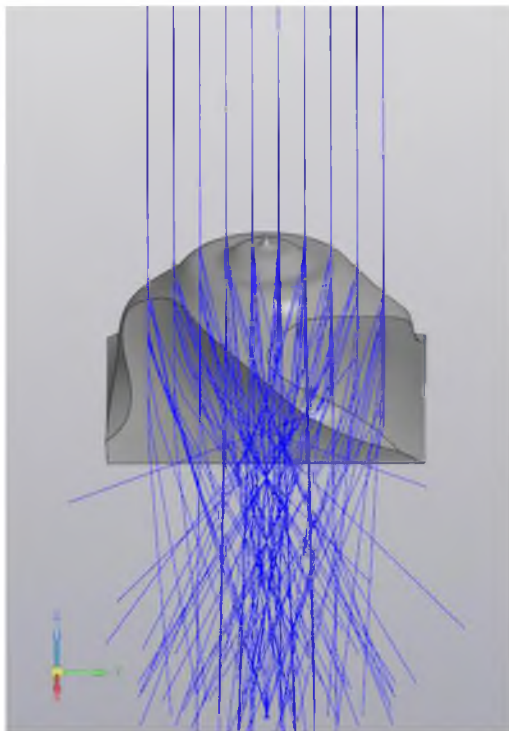


Рис. 2. Трассировка лучей через сложный элемент.

На рис. 2 намеренно выполнен сложный элемент, через который провести трассировку лучей без использования геометрического решателя не представляется возможным.

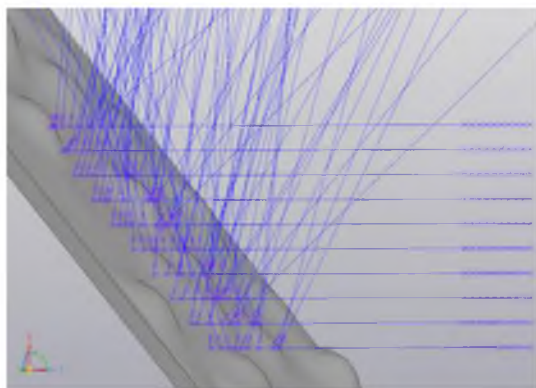


Рис. 3. Отражение лучей от выполненной при помощи сплайновой формы искривлённой поверхности.

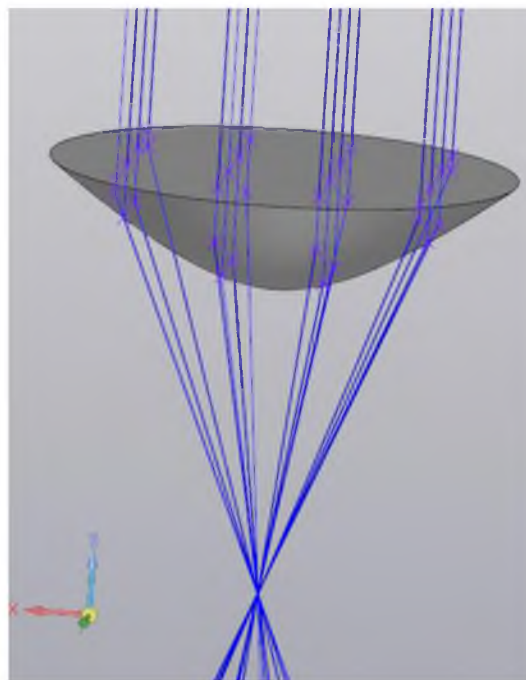


Рис. 4. Трассировка лучей через параболическую линзу.

На рис.4 демонстрируется результат трассировки лучей через параболическую линзу, нижняя (на рисунке) поверхность которой построена на основе решения уравнения

$$n_2x + \sqrt{(f-x)^2 + y^2} = n_2r + (f-r), \quad (3)$$

где n_2 – показатель преломления материала линзы; f – расстояние от плоской поверхности линзы до фокуса; r – расстояние от плоской поверхности линзы до вершины параболы.

Показатель преломления окружающей среды в этом уравнении предполагается равным 1.

На рисунке видно формирование фокуса без сферических аберраций, что подтверждает корректность результата трассировки лучей при помощи программы.

Алгоритм предусматривает хранение данных о ходе лучей, что позволяет производить различные вычисления с ними: расчёт поляризационного состояния и энергии, поиск фокуса системы.

ЛИТЕРАТУРА

- [1] Ли, Дж. Трёхмерная графика и анимация / Дж. Ли, Б. Уэр. — 2-е изд. — М.: Вильямс, 2002. — 640 с.
- [2] Губаев, М.С. Формирование векторного пучка с помощью конической преломляющей поверхности / М.С. Губаев, С.А. Дегтярев, Ю.С. Стрелков, С.Г. Волотовский, Н.А. Ивлиев, С.Н. Хонина // Компьютерная оптика. – 2021. – Т. 45, № 6. – С. 828-838. – DOI: DOI: 10.18287/2412-6179-CO-1036.