

УДК 004.921:004.925

## МЕТОД ТРАССИРОВКИ ЛУЧЕЙ КАК ОСНОВНАЯ ТЕХНОЛОГИЯ ФОТОРЕАЛИСТИЧНОГО РЕНДЕРИНГА

Ульянов А.Ю., Котюжанский Л.А., Рыжкова Н.Г.

ФГАОУ ВПО «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина», Екатеринбург, e-mail: drboolean@gmail.com

В статье рассматривается базовая механика метода трассировки лучей – технология отложенного построения фотореалистичных изображений, ее достоинства и недостатки. В частности, материал затрагивает анализ алгоритма построения растрового изображения при проецировании трехмерных графических примитивов на двумерную плоскость, обзор моделей освещения и техники построения теней, отражений. В процессе растеризации графического примитива результирующий цвет каждого пикселя определяется на основе выборки из центра пересечения луча с поверхностью геометрического объекта. Статья содержит описание основы полноэкранного сглаживания – алгоритма постобработки изображения, при котором производится формирование субпикселей избыточным количеством дискретных выборок на конкретный пиксель для устранения эффекта «ступенчатости» результирующего растрового изображения. Исследование всех алгоритмов выполнено с помощью авторского демонстрационного приложения.

**Ключевые слова:** трассировка лучей, компьютерная графика, рендеринг, обработка изображений

## RAY TRACING AS AN ESSENTIAL TECHNOLOGY FOR PRODUCING PHOTOREALISTIC COMPUTER-GENERATED IMAGERY

Ulyanov A.Y., Kotyuzhanskiy L.A., Ryzhkova N.G.

Ural Federal University named after the first President of Russia B.N. Yeltsin,

Ekaterinburg, e-mail: drboolean@gmail.com

The article explains the basic mechanics of the ray tracing technology for generating an image by tracing the path of light through pixels in an image plane and simulating the effects of its encounters with virtual objects in a Cartesian coordinate system. This paper includes the analysis of ray tracing algorithm – its advantages and disadvantages with brief overview of common shading techniques and reflection models. It also describes supersampling as an example of spartial anti-aliasing method. A line can only be shown as a collection of pixels, and therefore appears jagged unless it is perfectly horizontal or vertical. The aim of supersampling is to reduce this effect.

**Keywords:** ray tracing, computer graphics, rendering, image processing

Одним из методов визуализации (рендеринга) в компьютерной графике является метод «бросания лучей» (ray casting), при котором растровое изображение строится на основе замеров пересечения лучей с визуализируемой поверхностью в пространстве. Этот термин впервые использовался в компьютерной графике в 1982 году в работе Скотта Рота [1], который применил его для описания метода рендеринга CSG-моделей.

Первый алгоритм рейкастинга, используемый для рендеринга, был представлен Артуром Аппелем в 1968 году [2]. В основе рейкастинга лежит идея генерировать лучи из точки наблюдения сцены, один луч на пиксель, и находить самый близкий объект, который блокирует путь распространения этого луча. Используя свойства материала и эффект света в сцене, алгоритм рейкастинга может определить затенение данного объекта.

В реальной природе источник света испускает луч света, который, проходя через пространство, пересекает какую-либо преграду, которая прерывает распространение

этого светового луча. В любой точке пути с лучом света может случиться любая комбинация трех процессов: поглощение, отражение и преломление. Некоторые из лучей, сгенерированных источником света, распространяются по пространству и в конечном счете попадают на область просмотра.

Алгоритм трассировки лучей (ray tracing) применяет рейкастинг для расчета первичных пересечений луча с объектами сцены и дополняет его генерацией дополнительных лучей для формирования световых бликов, теней, отражений, тем самым повышая уровень фотореалистичности изображения.

Для реализации алгоритма трассировки лучей и получения результирующего изображения сцены, состоящей из геометрических примитивов в пространстве  $R^3$ , понадобятся следующие программные модели:

- модель луча в трехмерном декартовом пространстве;
- модель лучевого эмиттера (камера);
- математические модели графических примитивов;
- модель точечного источника света.

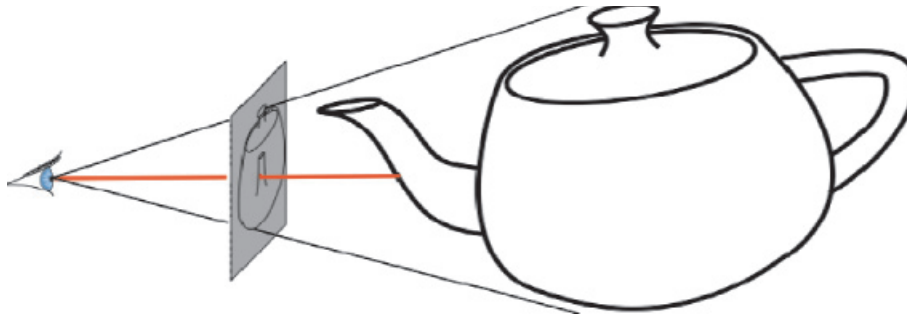


Рис. 1. Принцип построения изображения. Механизм упрощенной камеры-обскуры [3]

Для построения первичного луча и определения первых пересечений с объектами сцены вводятся понятия источника лучей и плоской области обзора. В основе модели лучевого эмиттера лежит механизм упрощенной камеры-обскуры (рис. 1) с бесконечно малым отверстием, через которое свет проникает на область обзора.

Определим плоскую область обзора как массив пикселей  $\{n \times m\}$ , где  $n$  и  $m$  – количество пикселей в ширину и высоту соответственно, используя предварительно подготовленные структуры данных.

Установим начало координат в точку  $O(x_o, y_o, z_o)$ , центр камеры в точку  $C(x_c, y_c, z_c)$ . Зададим фокусный центр камеры точкой  $F(x_f, y_f, z_f)$ . Вектор  $\vec{w} = \overline{CF}$  определяет направление камеры.

Для корректного определения направления распространения луча от позиции камеры к центру каждого пикселя плоской области обзора понадобится сформировать ортогональный базис  $\{\vec{u}, \vec{v}, \vec{w}\}$  при помощи операций векторного произведения:

$$\vec{u} = \vec{w} \times \vec{j}_{(0,1,0)}; \quad (1)$$

$$\vec{v} = \vec{u} \times \vec{w}. \quad (2)$$

Направление первичного луча  $\vec{R}$  (рис. 2) определяется по следующим формулам:

$$\alpha = \tan\left(\frac{fov_x}{2}\right) \cdot \left(\frac{i - \left(\frac{n}{2}\right)}{\frac{n}{2}}\right); \quad (3)$$

$$\beta = \tan\left(\frac{fov_y}{2}\right) \cdot \left(\frac{\left(\frac{m}{2}\right) - j}{\frac{m}{2}}\right); \quad (4)$$

$$\vec{R} = \overline{OC} + \frac{\vec{w} + \alpha\vec{u} + \beta\vec{v}}{|\vec{w} + \alpha\vec{u} + \beta\vec{v}|}, \quad (5)$$

где  $\alpha, \beta$  – величины смещения луча по осям  $X, Y$  области обзора;  $i, j$  – целочисленные координаты пикселя, для которого генерируется луч;  $fov_x, fov_y$  – горизонтальный и вертикальный углы обзора камеры.

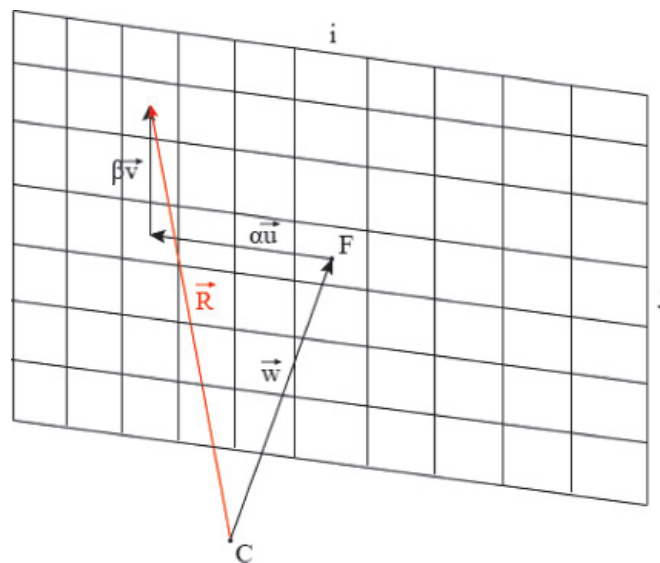


Рис. 2. Вычисление направления распространения луча от позиции камеры к центру каждого пикселя координатной плоскости

Пусть

$$\vec{d} = \frac{\vec{w} + \alpha\vec{u} + \beta\vec{v}}{|\vec{w} + \alpha\vec{u} + \beta\vec{v}|}.$$

Таким образом, можно записать векторное уравнение пучка лучей, исходящих из точки  $O$  с направляющими векторами  $\vec{d}$ :

$$\vec{R}_{ij}(t) = \vec{OC} + t\vec{d}, \quad t \geq 0, \quad (6)$$

где  $t$  – расстояние от начала луча до любой точки на нем;  $\vec{d}$  – вектор направления пространства луча.

Ниже приведены алгоритмы определения пересечений со следующими объектами, которые используются в демонстрационном приложении: плоскость и сфера.

Рассмотрим векторное уравнение плоскости, проходящей через фиксированную точку  $Q$  перпендикулярно вектору нормали  $\vec{n}$ :

$$\vec{n} \cdot \vec{QP} = 0, \quad (7)$$

где  $P(x_p, y_p, z_p)$  – произвольная точка плоскости.

Решение системы уравнений (6), (7)

$$\begin{cases} \vec{OC} + t\vec{d} \\ \vec{n} \cdot \vec{QP} = 0 \end{cases}$$

определяет значение параметра  $t$ , соответствующего точке пересечения луча с плоскостью:

$$t = \frac{\vec{n} \cdot \vec{OQ}}{\vec{n} \cdot \vec{d}}. \quad (8)$$

Если  $t < 0$ , плоскость располагается позади камеры и луч ее не пересекает. Если  $t \geq 0$ , точка пересечения находится на расстоянии  $|\vec{OC} + t\vec{d}|$  от центра расположения лучевого эмиттера. Если  $\vec{n} \cdot \vec{d} = 0$ , то луч проходит параллельно плоскости и не пересекает ее.

Аналогично решается задача о пересечении луча и сферы радиуса  $R$  с центром в точке  $Q_s(x_s, y_s, z_s)$ :

$$(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2 = R^2. \quad (9)$$

Значения параметра  $t$ , при котором луч пересекает сферу, определяются корнями квадратного уравнения

$$at^2 + bt + e = 0, \quad (10)$$

где

$$a = |\vec{Q_sC}|^2; \quad b = 2((\vec{Q_sC}) \cdot \vec{d}); \quad e = |\vec{Q_sC}|^2 - R^2.$$

В зависимости от значений  $a$ ,  $b$ ,  $e$  возможны следующие варианты решения уравнения (9) относительно  $t$ :

- Уравнение имеет два положительных вещественных корня. Луч пересекает сферу в двух точках (входит и выходит из нее). Соответственно далее необходимо работать с пересечением с наименьшей дистанцией.

- Уравнение имеет два одинаковых положительных или отрицательных вещественных корня. Луч проходит по касательной к сфере.

- Уравнение имеет положительный и отрицательный вещественные корни. Лучевой эмиттер находится в самой сфере, и луч только выходит из нее.

- Уравнение имеет комплексные корни. Луч не пересекает сферу.

Установив минимальное и максимальное значения параметра  $t$ , определим ближнюю и дальнюю плоскости отсекающей трехмерной сцены, тем самым завершив конфигурацию усеченной пирамиды обзора камеры.

Реализовав алгоритм рейкастинга, описанный на предыдущем этапе, сформируем базовую сцену, используя подготовленные ранее графические примитивы. Расположим плоскость, сферу, точечный источник света<sup>1</sup>; установим соответственно перечисленным объектам коричневый, зеленый и белый цвета<sup>2</sup>.

Представленное выходное изображение является результатом проверки пересечений лучей с объектами сцены. Для достижения эффекта объема можно использовать известные модели освещения и затенения, в частности:

- изотропные модели диффузного освещения Ламберта [4] и Фонга [5], последняя из которых дополняет рассеянное освещение поверхности бликовой составляющей;

- модель стенсильного затенения поверхностей, которая визуализирует теневой объем.

В простейшей своей реализации технология теневого объема в рейтрейсинге заключается в генерации «теневого луча» из точки пересечения сцены первичным лучом по направлению к источнику света. Без учета коэффициента затухания света, если «теневого луча» достиг источника света, не пересекая при этом других объектов сцены, то точка освещена. В противном случае, если найдено хотя бы одно пересечение – точка находится в тени. При корректной программной реализации моделей освещения и затенения результирующее изображение приближается к реалистичному (рис. 3).

<sup>1</sup> Плоскость: точка (0, -1, 0), вектор нормали (0, 1, 0); сфера: точка (0, 0, 0), единичный радиус; источник света: точка (-7, 10, -10).

<sup>2</sup> Цвета покомпонентно, нормировано в цветовой модели RGB: коричневый (0,5; 0,25; 0,25), зеленый (0,5; 1,0; 0,5), белый (1,0; 1,0; 1,0).

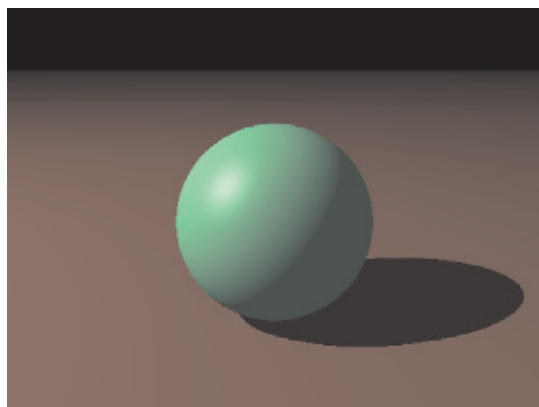


Рис. 3. Результат генерации стенильных теней и работы модели освещения по Блинну – Фонгу. Результат работы демонстрационного приложения

Для вычисления цветовой компоненты отражения в точке пересечения первичного луча и объекта необходимо создать новый луч в этой точке и направить его по нормали, восстановленной из этой точки по отношению к поверхности объекта, затем рекурсивно повторить общую процедуру получения цвета для первичного луча. Отметим, что в настоящей работе будет рассмотрена модель идеальных отражений. Поэтому для исключения бесконечного формирования лучей для взаимного расчета цвета отражения объектов, поверхности которых будут наделены таким свойством, необходимо ввести ограничивающее правило: поверхность должна отражать не всю энергию луча, а только некоторый процент.

После имплементации алгоритма генерации дополнительных лучей для формирования отражений усложним сцену, добавив примитив (сферу) и присвоив шахматный паттерн плоскости для придания наглядности работы новому алгоритму, получим следующий результат (рис. 4).

Если через каждый пиксель области обзора проводить один луч, то линии, гладкие в трехмерном пространстве, окажутся «ступенчатыми» на экране после построения проекции. Чтобы этого избежать, необходимо генерировать несколько лучей на каждый пиксель, считать для каждого цвет и находить его среднее значение.

На рис. 5 представлено сравнение двух результирующих изображений с различными настройками количества лучей, задействованных при вычислении цвета одного пикселя.

К достоинствам метода трассировки лучей можно отнести следующие положения:

- возможность рендеринга гладких объектов без аппроксимации их полигональными поверхностями;

- вычислительная сложность метода слабо зависит от сложности сцены;

- отсечение невидимых поверхностей, перспектива и корректные изменения поля зрения являются логическим следствием алгоритма.

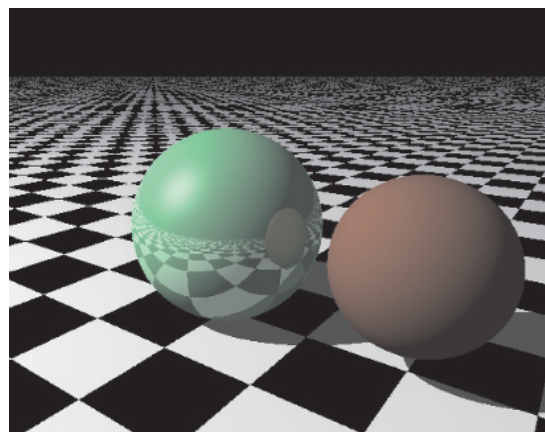


Рис. 4. Выходное изображение после генерации зеркальных отражений на поверхности сферы. Результат работы демонстрационного приложения

Серьезным недостатком метода является производительность. Метод растеризации и сканирования строк использует когерентность данных, чтобы распределить вычисления между пикселями. В то время как метод трассирования лучей каждый раз начинает процесс определения цвета пикселя заново, рассматривая каждый луч наблюдения в отдельности.

В статье представлены иллюстрации, полученные в результате работы демонстрационного приложения, которое было реализовано с использованием языка программирования C++ в интегрированной среде разработки Microsoft Visual Studio 2012. Зависимость времени генерации одного кадра растрового изображения с разрешением 640×480 пикселей и глубиной цвета 32 бита одной и той же сцены от количества дискретных выборок на один пиксель представлена в таблице.

Зависимость времени генерации кадра растрового изображения от количества избыточных дискретных выборок на один пиксель

Количество субпикселей на пиксель	Время генерации кадра изображения, мс
0	1783
4	7504
16	34344
25	61333

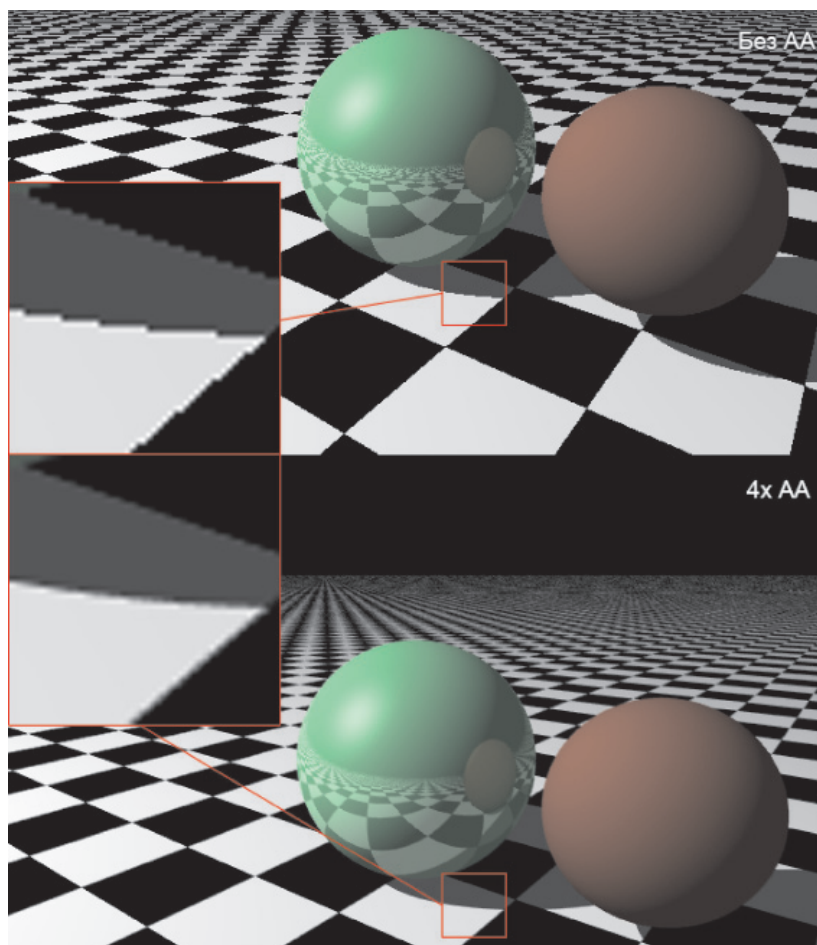


Рис. 5. Выходные изображения с различными настройками сглаживания демонстрационного приложения. Отчетливо видно устранение эффекта «ступенчатости» в зонах изображения, ограниченных красной рамкой

Благодаря тому, что каждый луч может обрабатываться независимо от других, метод трассировки лучей предоставляет возможность одновременного параллельного вычисления коллизий множества лучей. Этот факт подчеркивает необходимость дальнейших исследований в области оптимизации алгоритма для многоядерных вычислительных устройств CPU/GPU с целью повышения его производительности, а соответственно уменьшения времени на генерацию одного кадра результирующего изображения.

#### Список литературы/References

1. Roth S.D. Ray casting for modeling solids // Computer Graphics and Image Processing. –1982. – № 18. – P. 109–144.

2. Appel A. Some techniques for shading machine renderings of solids // AFIPS spring joint computer conference. IBM Research Center, Yorktown Heights, N.Y. – 1968. – P. 37–45.

3. Han-Wei Shen. Ray tracing basics [Электронный ресурс]. – Режим доступа: [http://web.cse.ohio-state.edu/~hwshen/681/Site/Slides\\_files/basic\\_algo.pdf](http://web.cse.ohio-state.edu/~hwshen/681/Site/Slides_files/basic_algo.pdf) (дата обращения: 30.10.2015).

4. Foley J.D., van Dam A. Feiner S.K., Hughes J.F. Computer Graphics: Principles and Practice // Addison-Wesley Publishing Company. – 1996. – P. 718–739.

5. Phong B.T. Illumination for computer generated pictures // Communications of ACM. – 1975. – № 6. – P. 311–317.

#### Рецензенты:

Осипов А.В., д.ф.-м.н., доцент, заведующий сектором топологии, Институт математики и механики УрО РАН, г. Екатеринбург;  
Гребеников В.И., д.ф.-м.н., профессор, главный научный сотрудник, Институт физики металлов УрО РАН, г. Екатеринбург.