	<b>Министерство науки и высшего образования Российской Федерации</b> <b>Федеральное государственное бюджетное образовательное учреждение</b> <b>высшего образования</b> <b>«Московский государственный технический университет</b> <b>имени Н.Э. Баумана</b> <b>(национальный исследовательский университет)»</b> <b>(МГТУ им. Н.Э. Баумана)</b>
---	--

ФАКУЛЬТЕТ	«Информатика и системы управления»
КАФЕДРА	«Программное обеспечение ЭВМ и информационные технологии»

## **ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

Студент	Лагутин Даниил Валерьевич
	_____ фамилия, имя, отчество
Группа	ИУ7-43Б
Тип практики	технологическая
Название предприятия	МГТУ им. Н. Э. Баумана

Студент	_____ подпись, дата	Лагутин Д. В. _____ фамилия, и.о.
Руководитель практики	_____ подпись, дата	Куров А. В. _____ фамилия, и.о.

Оценка \_\_\_\_\_



# Оглавление

Введение.....	4
1 Аналитическая часть.....	5
1.1 Анализ необходимых качеств реалистического изображения.....	5
1.2 Формализация объектов синтезируемой сцены.....	5
1.3 Анализ алгоритмов удаления невидимых линий и ребер поверхностей.....	6
1.4 Реализация теней в алгоритме трассировки лучей.....	8
1.5 Построения отражений и преломления в алгоритме трассировки лучей.....	9
1.6 Реализации глубины резкости в алгоритме трассировки лучей.....	13
1.7 Реализации подповерхностного рассеивания в алгоритме трассировки лучей	14
Вывод.....	16
2 Конструкторская часть.....	17
2.1 Требования в программному обеспечению.....	17
2.2 Алгоритм обратной трассировки лучей.....	18
2.3 Вычислений функций рассеивания.....	19
2.4 Вычислений функций подповерхностного рассеивания.....	20
Заключение.....	21
Список использованной литературы.....	22

# Введение

Компьютерная графика является неотъемлемой частью современного мира в особенности она востребована в сфере кинопроизводства и игровой промышленности. Именно поэтому синтез реалистического изображения является одной из важнейших задач.

Подобные условия требуют создания средств и методик, учитывающих такие оптические явления как преломление, отражение и рассеивание света, зависящие от свойств визуализируемых объектов. Но рост точности и детальности разрабатываемых алгоритмов зачастую приводит к более высоким затратам по времени и памяти.

Целью данной работы является разработка программы, позволяющей создать реалистическое изображение с учетом подповерхностного рассеивания, на основе трехмерной сцены наполненной объектами (сфера, куб, конус, цилиндр, полигональная модель) и источниками света, положение, количество и свойства которых задается пользователем.

Для достижения поставленной цели, требуется решить следующие задачи:

- 1) выявить неотъемлемые качества реалистического изображения;
- 2) провести анализ существующих алгоритмов;
- 3) выбрать оптимальные пути для решения основной задачи;
- 4) реализовать выбранные алгоритмы;
- 5) создать программный продукт для решения задачи.

# **1 Аналитическая часть**

В этом разделе будут рассмотрены модели представления трехмерных тел, формализованы объекты, наполняющие сцену, и требования, предъявляемые к работе программы и конечному результату. Будут рассмотрены алгоритмы построения трехмерного изображения, методы закраски, модели освещения.

## **1.1 Анализ необходимых качеств реалистического изображения**

Большинство алгоритмов, изученных в курсе компьютерной графики нацелены на корректное отображение геометрии объектов и их положения в сцене, однако для построения реалистического изображения этих аспектов недостаточно.

Первым аспектом, наполняющим сцену объемом можно считать тень. Однако большинство моделей подразумевает источник света материальной точкой, из-за чего все тени получаются четкими и однородными, что в общем случае неверно.

Подобное образом, для построения изображения стоит учитывать такую важную составляющую оптических систем, как линза. Изображения предметов на фотографии или у нас перед глазами получаются четкими только в случае, если они находятся в фокальной плоскости, иначе изображение размывается, что также необходимо учитывать.

Не стоит забывать, что объекты в сцене состоят из различных материалов, что вносит на рассмотрение блестящие и прозрачные предметы, с присущими явлениями отражения, преломления и рассеивания.

## **1.2 Формализация объектов синтезируемой сцены**

Сцена состоит из следующего набора объектов.

1. Геометрический объект: непосредственное представление объектов реального мира внутри сцены. В число рассматриваемых тел входят параметрические модели, такие как куб, сфера, конус, цилиндр, а также тела произвольной формы. Данная модель предназначена исключительно для хранения геометрического представления и иерархии составного тела.

Для описания каждого объекта используются:

- куб: координаты двух диагональных вершин;
- сфера: координаты центра и радиус;
- конус: координаты центра основания, радиус основания, высота;
- цилиндр: координаты центра нижнего основания, радиус, высота;
- тело произвольной формы: набор полигонов с указанием нормали к поверхности, определяющей наружную часть объекта. В качестве основной единицы такой модели был выбран треугольник, так как с его помощью можно представить полигон любой формы.

2. Свойство объекта: отражение качеств, присущих реальному объекту, в частности свечение, прозрачность, фактура.
3. Источник света: геометрическое тело, обладающее свойством свечения. Для источника света присущи такие вырожденные случаи геометрического тела, как точка (точечный источник света, направленный свет).
4. Карта окружения: куб, опоясывающий все объекты сцены и бесконечно удаленный от них. Служит для определения фонового изображения и создания дополнительных отражений.
5. Камера: геометрическое тело, определяющее видимую часть сцены. Положение задается координатой центра, направление взгляда совпадает с осью  $Z$  тела.

### **1.3 Анализ алгоритмов удаления невидимых линий и ребер поверхностей**

#### ***Алгоритм, использующий Z-буфер***

Смысл алгоритма заключается в использовании двух буферов: буфера кадра и Z-буфера, хранящих атрибуты и информацию о координате  $Z$  каждого пикселя соответственно.

Z-буфер инициализируется минимальным значением координаты, а буфер кадра — информацией о пикселе, описывающем фон. Глубина каждого нового пикселя при подсчете сравнивается со значением, которое уже есть в Z-буфере. В

случае, если новый пиксель расположен ближе к наблюдателю, то информация о нем заносится в буфер кадра и происходит редактирование Z-буфера.

**Вывод:** данный алгоритм достаточно прост в реализации и, в силу отсутствия сортировок, не требует больших затрат по времени и памяти (для современных компьютеров), однако требуются дополнительные техники для учета отражений, прозрачности и теней.

### *Алгоритм обратной трассировки лучей*

Алгоритм берет свое начало из законов оптики: наблюдатель видит объект, только если до него доходят лучи испускаемого света, прошедшие некоторый путь от источника.

Однако, вопреки физической модели, трассировка лучей происходит в обратном направлении — от наблюдателя к источнику. В противном случае приходится рассматривать множество лучей, которые никогда не попадут в наблюдаемую область.

**Вывод:** алгоритм построен на физически верной модели, что дает широкий простор для реализации различных визуальных явлений, но такая гибкость и обобщенность требует большого количества вычислений (возможность усовершенствования за счет параллельных вычислений, так как каждый пиксель в данном методе не зависит от окружающих).

### *Алгоритм Робертса*

Данный алгоритм решает задачу удаления невидимых ребер и граней только с выпуклыми телами, работает только в объектном пространстве.

У следующие этапы выполнения:

1. Удаление ребер, экранируемых самим телом;
2. Удаление невидимых ребер, экранируемых другими телами сцены;
3. Удаление невидимых ребер, образованных в результате протыкания.

**Вывод:** алгоритм обладает высокой точностью, в силу того, что все вычисления происходят в объектном пространстве, однако тела должны быть выпуклыми, что не подходит для нашей задачи.

## ***Алгоритм художника***

Данный алгоритм работает аналогично тому, как художник рисует картину - то есть сначала рисуются дальние объекты, а затем более близкие. Наиболее распространенная реализация алгоритма - сортировка по глубине, которая заключается в том, что произвольное множество граней сортируется по ближайшему расстоянию от наблюдателя, а затем отсортированные грани выводятся на экран в порядке от самой дальней до самой ближней.

**Вывод:** данный алгоритм требует меньше памяти в сравнении с остальными, однако так же, как и алгоритм Z-буфера требует дополнительных методов отображения визуальных эффектов.

## ***Вывод***

Так как цель работы — создание реалистического изображения, то лучшим выбором станет алгоритм трассировки лучей. Данный алгоритм предоставляет удобную базу для реализации оптических эффектов, а затраты по времени не окажут особого значения, так как объекты в сцене — статические, поэтому можно сконцентрироваться на качестве получаемого изображения.

## **1.4 Реализация теней в алгоритме трассировки лучей**

Выбранный в предыдущем пункте алгоритм трассировки лучей, позволяет легко реализовать отрисовку тени.

Корректное построение достигается за счет того, что после нахождения ближайшей точки пересечения очередного луча, из нее посылаются «теновые» лучи к каждому источнику света в сцене. Если теновой луч не достигает источника света (имеется более близкое пересечение с каким-либо объектом сцены), то точка не освещена, иначе — в точке рассчитывается интенсивность, согласно модели освещения.

Помимо этого, также решается проблема построения мягких теней, так как имеется возможность испускания множества теновых лучей к одному объекту, что позволит определить видимую часть и итоговую интенсивность.



## 1.5 Построения отражений и преломления в алгоритме трассировки лучей

Нахождение отражений и преломления также является логическим продолжением алгоритма трассировки лучей. Используя нормаль к поверхности и направление отраженного луча, нетрудно найти падающий и преломленный лучи, для которого аналогично применяется трассировка.

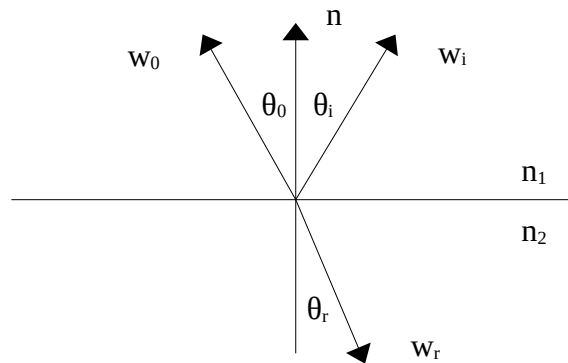


Рисунок 1: Пример расположения лучей на границе объекта

Найдем выражение для отраженного и преломленного лучей.

*Отраженный луч:*

$$\hat{w}_i = \alpha \hat{w}_0 + \beta \hat{n}$$

*Из равенства углов падения и отражения :*

$$\hat{w}_0 \hat{n} = \hat{w}_i \hat{n} = \alpha \hat{w}_0 \hat{n} + \beta \hat{n}^2 = \alpha \hat{w}_0 \hat{n} + \beta$$

$$\beta = \hat{w}_0 \hat{n} - \alpha \hat{w}_0 \hat{n}$$

$$|\hat{w}_0| = |\hat{w}_i|$$

$$|\hat{w}_0|^2 = |\hat{w}_i|^2$$

$$\hat{w}_0^2 = 1 = \hat{w}_i^2 = (\alpha \hat{w}_0 + \beta \hat{n})^2 = \alpha^2 \hat{w}_0^2 + 2\alpha\beta \hat{w}_0 \hat{n} + \beta^2 \hat{n}^2 = \alpha^2 + 2\alpha\beta \hat{w}_0 \hat{n} + \beta^2 =$$

$$= \alpha^2 + 2\alpha(\hat{w}_0 \hat{n} - \alpha \hat{w}_0 \hat{n}) \hat{w}_0 \hat{n} + (\hat{w}_0 \hat{n} - \alpha \hat{w}_0 \hat{n})^2 =$$

$$= \alpha^2 + 2\alpha(\hat{w}_0 \hat{n})^2 - 2\alpha^2(\hat{w}_0 \hat{n})^2 + (\hat{w}_0 \hat{n})^2 - 2\alpha(\hat{w}_0 \hat{n})^2 + \alpha^2(\hat{w}_0 \hat{n})^2 = (\hat{w}_0 \hat{n})^2 + (1 - (\hat{w}_0 \hat{n})^2)\alpha^2$$

$$1 - (\hat{w}_0 \hat{n})^2 = (1 - (\hat{w}_0 \hat{n})^2)\alpha^2$$

$$\text{Если } 1 - (\hat{w}_0 \hat{n})^2 = 0 \Rightarrow \hat{w}_0 \hat{n} = \pm 1 \Rightarrow \hat{w}_0 \parallel \hat{n} \Rightarrow w_i = w_0$$

*Иначе  $\alpha^2 = 1 \Rightarrow \alpha = \pm 1$ , так как падающий и отраженный лучи могут находиться по одну сторону от нормали, то  $\alpha = -1$ ,  $\beta = 2\hat{w}_0 \hat{n}$*

$$\hat{w}_i = 2(\hat{w}_0 \hat{n})\hat{n} - \hat{w}_0$$

$$w_i = \hat{w}_i |w_0| = (2(\hat{w}_0 \hat{n})\hat{n} - \hat{w}_0) |w_0| = 2 \frac{w_0 n}{|\hat{w}_0| |n|} |\hat{w}_0| \frac{n}{|n|} - \frac{w_0}{|\hat{w}_0|} |\hat{w}_0| = 2 \frac{w_0 n}{n^2} n - w_0 \quad (1)$$

Преломленный луч :

$$\widehat{w}_r = \alpha \widehat{w}_0 + \beta \hat{n}$$

По закону Снеллиуса:

$$n_1 \sin(\theta_0) = n_2 \sin(\theta_r)$$

$$n_1^2 \sin^2(\theta_0) = n_2^2 \sin^2(\theta_r)$$

$$n_1^2 (1 - \cos^2(\theta_0)) = n_2^2 (1 - \cos^2(\theta_r))$$

$$n_1^2 (1 - (\widehat{w}_0 \hat{n})^2) = n_2^2 (1 - (\widehat{w}_r \hat{n})^2) = n_2^2 (1 - (\alpha \widehat{w}_0 \hat{n} + \beta \hat{n})^2) = n_2^2 (1 - (\alpha^2 (\widehat{w}_0 \hat{n})^2 + 2\alpha\beta(\widehat{w}_0 \hat{n}) + \beta^2)) \quad (a)$$

$$|\widehat{w}_0| = |\widehat{w}_r| = 1$$

$$|\widehat{w}_0|^2 = |\widehat{w}_r|^2 = 1$$

$$1 = (\alpha \widehat{w}_0 + \beta \hat{n})^2 = \alpha^2 + 2\alpha\beta(\widehat{w}_0 \hat{n}) + \beta^2 \quad (b)$$

$$2\alpha\beta(\widehat{w}_0 \hat{n}) = 1 - \alpha^2 - \beta^2 \Rightarrow (1)$$

$$n_1^2 (1 - (\widehat{w}_0 \hat{n})^2) = n_2^2 (1 - (\alpha^2 (\widehat{w}_0 \hat{n})^2 + 1 - \alpha^2 - \beta^2 + \beta^2))$$

$$n_1^2 (1 - (\widehat{w}_0 \hat{n})^2) = n_2^2 (1 - (\widehat{w}_0 \hat{n})^2) \alpha^2$$

$\alpha = \pm \frac{n_1}{n_2}$ , так как падающий и преломленный лучи не могут находиться по одну сторону

от нормали, то  $\alpha = -\frac{n_1}{n_2} \Rightarrow (b)$

$$\beta^2 - 2\frac{n_1}{n_2}(\widehat{w}_0 \hat{n})\beta + \frac{n_1^2 - n_2^2}{n_2^2} = 0$$

$$D = 4\frac{n_1^2}{n_2^2}(\widehat{w}_0 \hat{n})^2 - 4\frac{n_1^2 - n_2^2}{n_2^2} = \frac{4}{n_2^2}(n_2^2 - n_1^2(1 - (\widehat{w}_0 \hat{n})^2))$$

Пусть  $D' = n_2^2 - n_1^2(1 - (\widehat{w}_0 \hat{n})^2)$ , тогда  $D = \frac{4}{n_2^2} D'$ . В силу того, что  $\frac{4}{n_2^2} > 0$ ,

сравнение знака дискриминанта можно проводить с величиной  $D'$ .

Если  $D' < 0$  — преломления нет.

Если  $D' = 0$  —  $\beta = \frac{n_1}{n_2} \widehat{w}_0 \hat{n} \Rightarrow \widehat{w}_r = -\frac{n_1}{n_2} \widehat{w}_0 + \frac{n_1}{n_2} (\widehat{w}_0 \hat{n}) \hat{n}$

$$\Rightarrow w_r = -\frac{n_1}{n_2} w_0 + \frac{n_1}{n_2} \frac{w_0 n}{n^2} n = \frac{n_1}{n_2} \left( \frac{w_0 n}{n^2} n - w_0 \right). \quad (2)$$

$$\text{Если } D' > 0 \quad \beta = \frac{2\frac{n_1}{n_2}(\widehat{w}_0 \hat{n}) \pm \sqrt{D}}{2} = \frac{2\frac{n_1}{n_2}(\widehat{w}_0 \hat{n}) \pm \frac{2}{n_2} \sqrt{D'}}{2} = \frac{n_1(\widehat{w}_0 \hat{n}) \pm \sqrt{D'}}{n_2}.$$

Для определения знака подставим  $\alpha$  и  $\beta$  в выражение для  $\widehat{w}_r$  и проверим, что  $\widehat{w}_r$  и  $\hat{n}$  расположены по разные стороны от границы пересечения, то есть  $\widehat{w}_r \hat{n} < 0$

$$\begin{aligned}
\widehat{w}_r \hat{n} &= \left( -\frac{n_1}{n_2} \widehat{w}_0 + \frac{n_1(\widehat{w}_0 \hat{n}) \pm \sqrt{D'}}{n_2} \hat{n} \right) \hat{n} < 0 \\
&= -\frac{n_1}{n_2} (\widehat{w}_0 \hat{n}) + \frac{n_1(\widehat{w}_0 \hat{n}) \pm \sqrt{D'}}{n_2} < 0 \\
&\pm \frac{\sqrt{D'}}{n_2} < 0 \Rightarrow \beta = \frac{n_1(\widehat{w}_0 \hat{n}) - \sqrt{D'}}{n_2} \Rightarrow \widehat{w}_r = -\frac{n_1}{n_2} \widehat{w}_0 + \frac{n_1(\widehat{w}_0 \hat{n}) - \sqrt{D'}}{n_2} \hat{n} \Rightarrow \\
\Rightarrow w_r &= -\frac{n_1}{n_2} w_0 + \frac{n_1}{n_2} \frac{(w_0 n)}{n^2} n - \frac{\sqrt{D'}}{n_2} \frac{n}{|n|} |w_0| = \frac{n_1}{n_2} \left( \frac{(w_0 n)}{n^2} n - w_0 \right) - \frac{|w_0| \sqrt{D'}}{n_2 |n|} n
\end{aligned} \quad (3)$$

Интенсивность света, распространяющегося в направлении  $w_0$  из точки пространства  $p$ , может быть описана при помощи следующей формулы.

$$L_0(p, w_0) = \int_{S^2} f(p, w_0, w_i) L_i(p, w_i) \cos(\theta_i) dw_i \quad (4)$$

Где  $w_i$  - направление падающего света в точке  $p$ ;  $f$  - функция рассеяния;  $L_i$  - интенсивность света, падающего по направлению  $w_i$ ;  $\theta_i$  - угол между нормалью к поверхности и вектором направления падающего света. Значение интеграла вычисляется по поверхности единичной сферы.

Такой метод описания дает возможность реализовать любую модель освещения и легко произвести подмену при необходимости, засчет введения *функции рассеяния*, показывающую какая часть падающего света отражается в указанном направлении.

$$f(p, w_0, w_i) \in [0, 1] \quad (5)$$

### **Модель освещения Ламберта**

Данная модель предназначена для представления идеального диффузного рассеивания, то есть подразумевается, что свет при попадании на поверхность равномерно распространяется во всех направлениях.

$$f_d(p, w_0, w_i) = k_d \quad (6)$$

Где  $f_d$  - функция рассеивания для диффузного света,  $k_d$  - коэффициент диффузного отражения.

### **Модель освещения Фонга**

Данная модель базируется на модели Ламберта и добавляет зеркальную составляющую, то есть блик на поверхности объекта. Для достижения этого эффекта к формуле (6) добавляется следующая функция зеркального рассеивания.

$$f_s(p, w_0, w_i) = k_s \cos(\theta_{re})^\alpha = k_s \left( \frac{w_0 w_{re}}{|w_0| |w_{re}|} \right)^\alpha \quad (7)$$

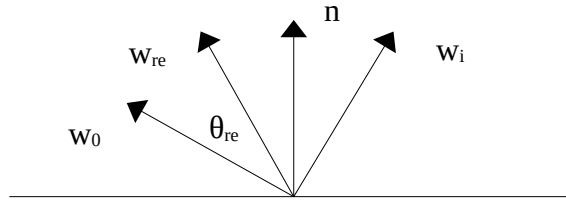


Рисунок 2: Расположение лучей при расчете модели Фонга

Где  $\theta_{re}$  — угол между отраженным лучом и лучом взгляда,  $k_s$  - коэффициент зеркального отражения,  $\alpha$  - коэффициент блеска.

### Отражение Френеля

В отличие от представленных эмпирических моделей, данная — физически корректна и опирается на решение уравнений Максвелла.

$$F(p, w_0, w_i) = \frac{1}{2} (r_\perp^2 + r_\parallel^2) \quad (8)$$

Где  $r_\perp$  - коэффициент отражения для света с параллельной поляризацией,  $r_\parallel$  - коэффициент отражения для света с перпендикулярной поляризацией.

Реализация метода разделяет материалы на два типа: проводники и диэлектрики (согласно их физическим свойствам).

#### 1. Отражение для диэлектрика

$$\begin{aligned} r_\parallel &= \frac{n_2 \cos(\theta_i) - n_1 \cos(\theta_t)}{n_2 \cos(\theta_i) + n_1 \cos(\theta_t)} \\ r_\perp &= \frac{n_1 \cos(\theta_i) - n_2 \cos(\theta_t)}{n_1 \cos(\theta_i) + n_2 \cos(\theta_t)} \end{aligned} \quad (9)$$

Где  $\theta_t$  - угол между нормалью и преломленным лучом, который может быть вычислен по закону Снеллиуса.

#### 2. Отражение для проводника

В отличие от диэлектриков коэффициент преломления для металлов — комплексное число  $\bar{n} = n + ik$ , где  $n$  — показатель преломления,  $k$  — коэффициент экстинкции.

$$r_{\perp} = \frac{a^2 + b^2 - 2a \cos(\theta_i) + \cos(\theta_i)^2}{a^2 + b^2 + 2a \cos(\theta_i) + \cos(\theta_i)^2}$$

$$r_{\parallel} = r_{\perp} \frac{\cos(\theta_i)^2 (a^2 + b^2) - 2a \cos(\theta_i) \sin(\theta_i)^2 + \sin(\theta_i)^4}{\cos(\theta_i)^2 (a^2 + b^2) + 2a \cos(\theta_i) \sin(\theta_i)^2 + \sin(\theta_i)^4} \quad (10)$$

$$\text{Где } a^2 = \frac{\sqrt{(n^2 - k^2 - \sin(\theta_i)^2)^2 - 4n^2 k^2} + (n^2 - k^2 - \sin(\theta_i)^2)}{2},$$

$$b^2 = \frac{\sqrt{(n^2 - k^2 - \sin(\theta_i)^2)^2 - 4n^2 k^2} - (n^2 - k^2 - \sin(\theta_i)^2)}{2},$$

$$a^2 + b^2 = \sqrt{(n^2 - k^2 - \sin(\theta_i)^2)},$$

$$\bar{n} = \frac{\bar{n}_2}{\bar{n}_1} = \frac{n_2 + i k_2}{n_1 + i k_1} = \frac{n_1 n_2 + k_1 k_2}{n_1^2 + k_1^2} + i \frac{k_2 n_1 - n_2 k_1}{n_1^2 + k_1^2} \Rightarrow$$

$$\Rightarrow n = \frac{n_1 n_2 + k_1 k_2}{n_1^2 + k_1^2}, \quad k = \frac{k_2 n_1 - n_2 k_1}{n_1^2 + k_1^2}$$

Однако в таком виде не получится добиться зеркального блика, для этого введем функцию  $\delta_{w_0}(w) = \begin{cases} 1, & \text{если } w = w_0 \\ 0, & \text{иначе} \end{cases}$ . Тогда

$$L_0(p, w_0) = \int_{S^2} \delta_{w_{re}}(w_i) F(p, w_0, w_i) L_i(p, w_i) \cos(\theta_i) dw_i$$

$$= F(p, w_0, w_{re}) L_{re}(p, w_{re}) \cos(\theta_{re}),$$

где  $w_{re}$  — зеркально отраженный луч для  $w_0$ .

Но желаемый результат

$$L_0(p, w_0) = F(p, w_0, w_{re}) L_{re}(p, w_{re}) \Rightarrow$$

$$f_s(p, w_0, w_i) = \frac{\delta_{w_{re}}(w_i)}{\cos(\theta_{re})} F(p, w_0, w_i) \quad (11)$$

## 1.6 Реализации глубины резкости в алгоритме трассировки лучей

### Трассировка лучей по поверхности линзы

Трассировка лучей также позволяет достаточно очевидно реализовать глубину резкости, засчет симуляции множества лучей проходящих через оптическую систему (линзу).

К примеру можно реализовать тонкую линзу, пропуская луч из точки картинной плоскости в произвольные точки на поверхности диска, представляющего собой линзу. Если объект находится в фокусе, то все лучи попадут в одну точку, в результате чего изображение будет четким, иначе — множество различных значений излучения создадут размытое изображение.

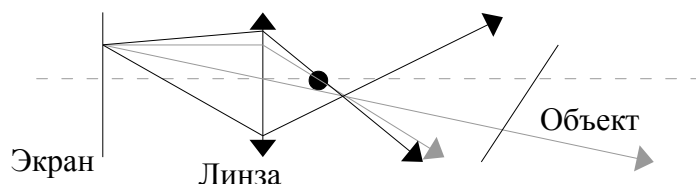


Рисунок 3: Лучи, попадающие в одну точку, после прохождения через тонкую линзу.

### ***Использование накопительного буфера***

Метод подразумевает, что для получения необходимого эффекта, нужно будет получить множество буферов кадра, для каждого из которых положение камеры будет изменяться согласно параметрам камеры. Результирующее изображение получается суммированием и усреднением полученных значений в накопительном буфере на каждой новой итерации алгоритма.

### ***Слоистая глубина резкости***

Алгоритм реализует поставленную задачу, путем разделения сцены на «слои», согласно глубине изображения в точке сцены. Далее к полученным слоям применяется алгоритм фильтрации, для достижения необходимого размытия текущего слоя. Это метод менее затратен в плане вычислений, так как для сцены требуется вычислить только один буфер кадра, разбитый на несколько слоев.

### ***Вывод***

В программе будет использоваться метод распространения лучей по поверхности линзы, так как он позволит реализовать практически любую оптическую систему и произвести подмену в случае необходимости.

## **1.7 Реализации подповерхностного рассеивания в алгоритме трассировки лучей**

Подповерхностное рассеивание описывает механизм распространения света, при котором свет, проникая внутрь полупрозрачного тела через его поверхность, рассеивается внутри самого тела, многократно отражаясь от частиц тела в

случайном направлении и на нерегулярные углы. В итоге свет выходит из объекта в выходной точке, отличной от точки вхождения в объект.

Нетрудно заметить, что необходимый результат можно достичь, произведя дополнительное интегрирование соотношения (4) по поверхности тела  $A$ , что потребует введения дополнительной *функции поверхностного рассеивания*

$$S(p_0, w_0, p_i, w_i)$$

$$L_0(p_0, w_0) = \int_A \int_{H^2} S(p_0, w_0, p_i, w_i) L_i(p_i, w_i) \cos(\theta_i) dw_i dA \quad (12)$$

Однако задание функции рассеивания даже для плоского случая достаточно проблематично, даже без учета составных тел. Поэтому на практике применяется другое соотношение для  $S$ , называемое *разделяемой функцией поверхностного рассеивания*

$$S(p_0, w_0, p_i, w_i) = (1 - F_r(\cos \theta_0)) S_p(p_0, p_i) S_w(w_i) \quad (13)$$

Где  $1 - F_r(\cos \theta_0)$  - формула отражения Френеля (8), показывающая часть света, доходящую до наблюдателя после выхода за границы объекта;

$S_p(p_0, p_i)$  - функция, описывающая часть света, способную пройти расстояние от точки входа до точки выхода внутри тела;

$$S_w(w_i) = \frac{1 - F_r(\cos \theta_i)}{c \pi} \text{ - аналог первой части выражения для света, попадающего}$$

в границы объекта. Нормализующий коэффициент  $c$  служит для выполнения условия  $\int_{H^2} S_w(w_i) \cos \theta dw = 1$  (интегрирование по половине сферы, отсекаемой поверхностью).

Для демонстрации упрощения подставим (13) в (12).

$$\begin{aligned} L_0(p_0, w_0) &= \int_A \int_{H^2} S(p_0, w_0, p_i, w_i) L_i(p_i, w_i) \cos(\theta_i) dw_i dA = \\ &= \int_A \int_{H^2} (1 - F_r(\cos \theta_0)) S_p(p_0, p_i) S_w(w_i) L_i(p_i, w_i) \cos(\theta_i) dw_i dA = \\ &= (1 - F_r(\cos \theta_0)) \int_A S_p(p_0, p_i) \int_{H^2} S_w(w_i) L_i(p_i, w_i) \cos(\theta_i) dw_i dA \end{aligned} \quad (14)$$

Несмотря на то, что мы разделили интеграл на части, внутренняя часть которого вполне укладывается в имеющееся соотношение для отражения (4), остается проблема с поддержкой геометрии произвольных тел, поэтому введем следующее приближение и получим окончательную формулу.

$$S_p(p_0, p_i) \approx S_r(|\overline{p_0 p_i}|) \quad (15)$$

$$\begin{aligned} L_0(p_0, w_0) &= (1 - F_r(\cos \theta_0)) \int_A S_p(p_0, p_i) \int_{H^2} S_w(w_i) L_i(p_i, w_i) \cos(\theta_i) dw_i dA \approx \\ &\approx (1 - F_r(\cos \theta_0)) \int_A S_r(|\overline{p_0 p_i}|) \int_{H^2} S_w(w_i) L_i(p_i, w_i) \cos(\theta_i) dw_i dA \end{aligned} \quad (16)$$

## Вывод

В данном разделе были описаны модели представления моделей в сцене и свойства, необходимые для физически корректной визуализации объектов с учетом подповерхностного рассеивания. В качестве основного алгоритма удаления невидимых линий был выбран алгоритм трассировки лучей, на основании которого будет строиться реализация отражений, теней, глубины резкости и подповерхностного рассеивания.



## 2 Конструкторская часть

### 2.1 Требования в программному обеспечению

Программа должна предоставлять следующий функционал:

- возможность интерактивного изменения положения объектов и источников света в сцене;
- возможность интерактивного добавления/удаления объектов в сцене;
- возможность интерактивного изменения свойств объектов (коэффициент восприятия света, коэффициент преломления, тип материала объекта, интенсивность и цвет излучения);
- возможность интерактивного масштабирования и поворота объектов сцены.

К программе предоставляются следующие требования:

- программа должна корректно реагировать на любые действия пользователя;
- программа должна предоставлять приемлемое время отклика (не более 1 сек) при работе в интерактивном режиме.

## 2.2 Алгоритм обратной трассировки лучей

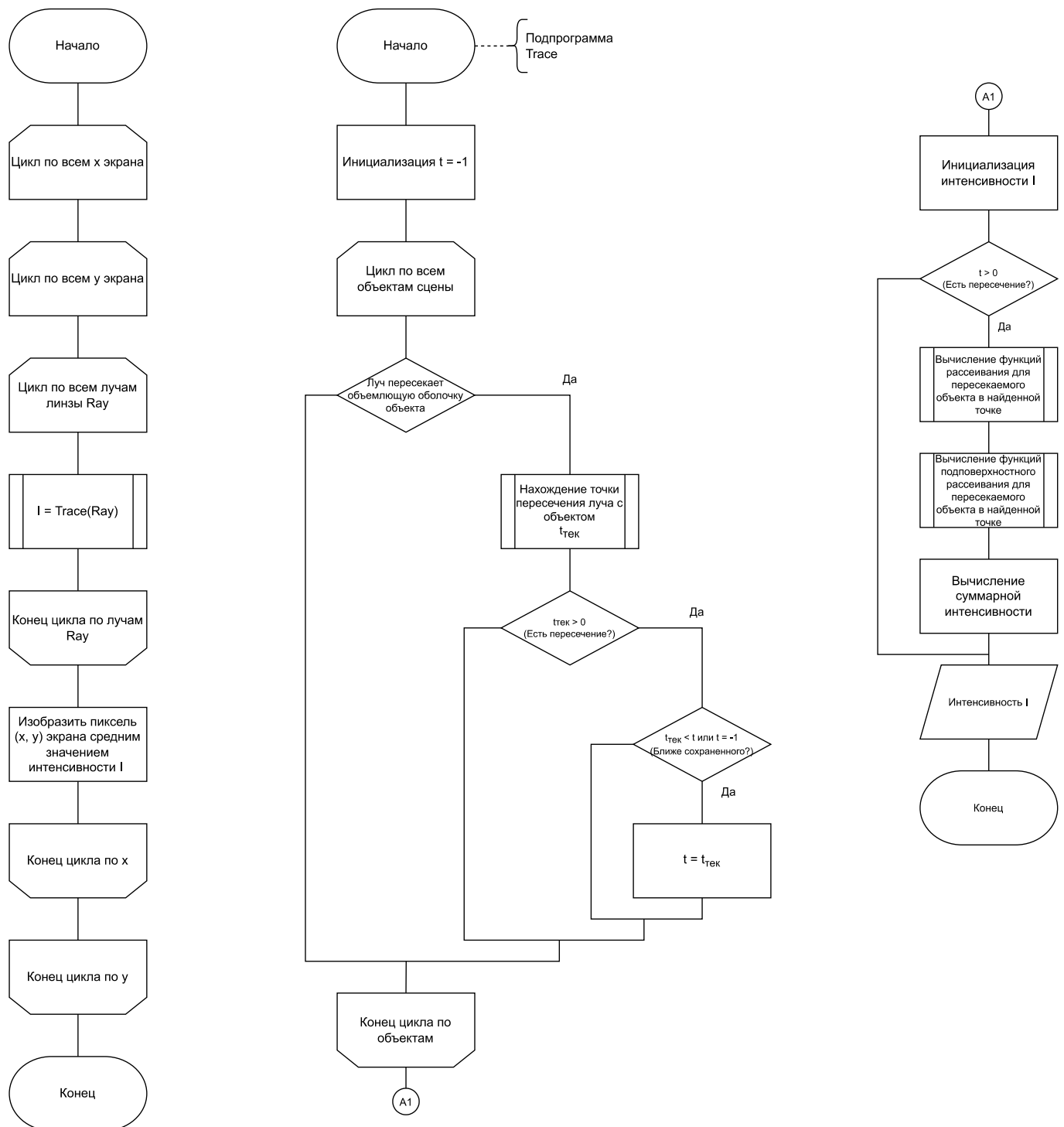


Рисунок 4: Схема алгоритма трассировки лучей

## 2.3 Вычислений функций рассеивания

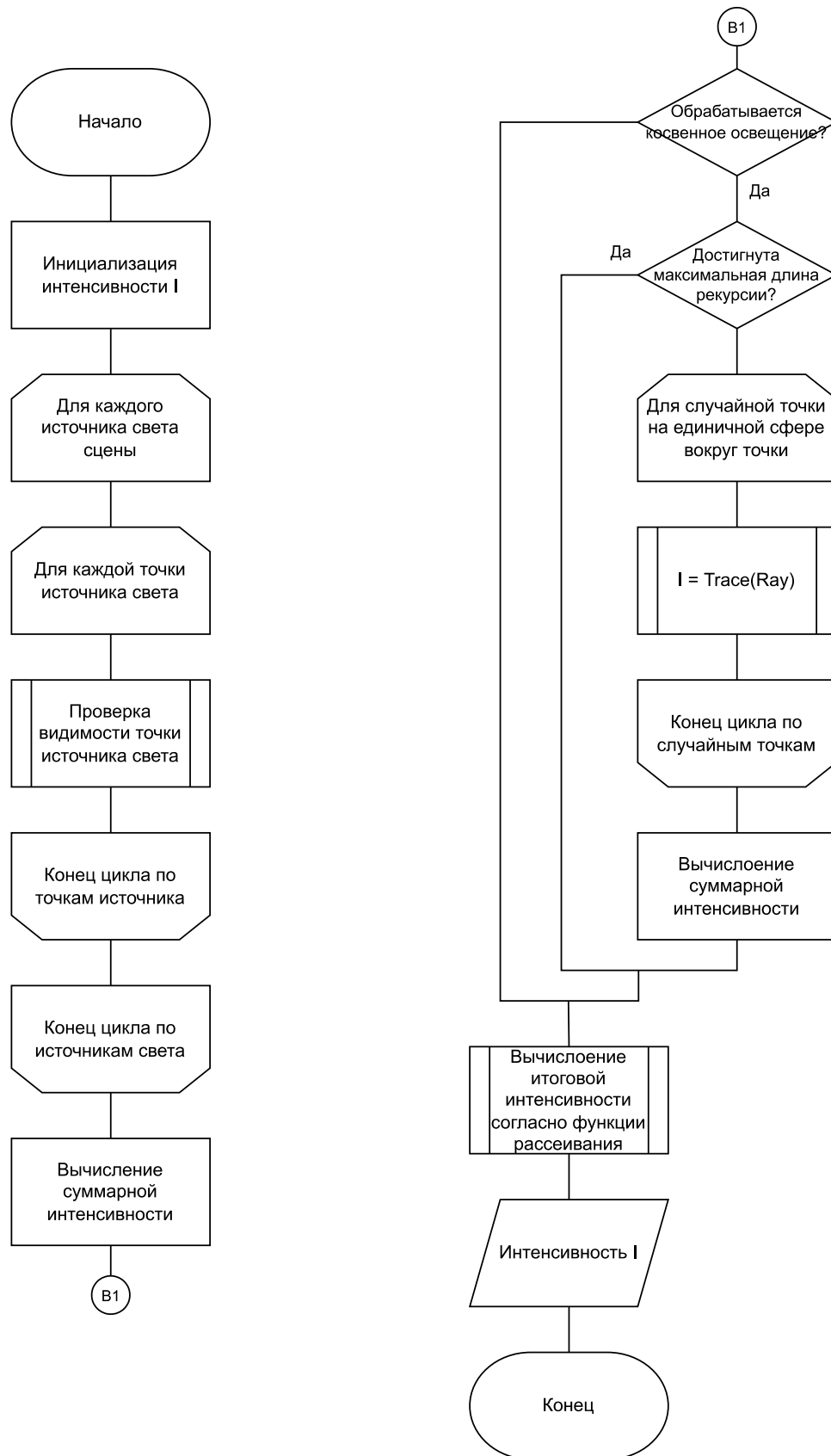


Рисунок 5: Схема вычисления функции рассеивания

## 2.4 Вычислений функций подповерхностного рассеивания

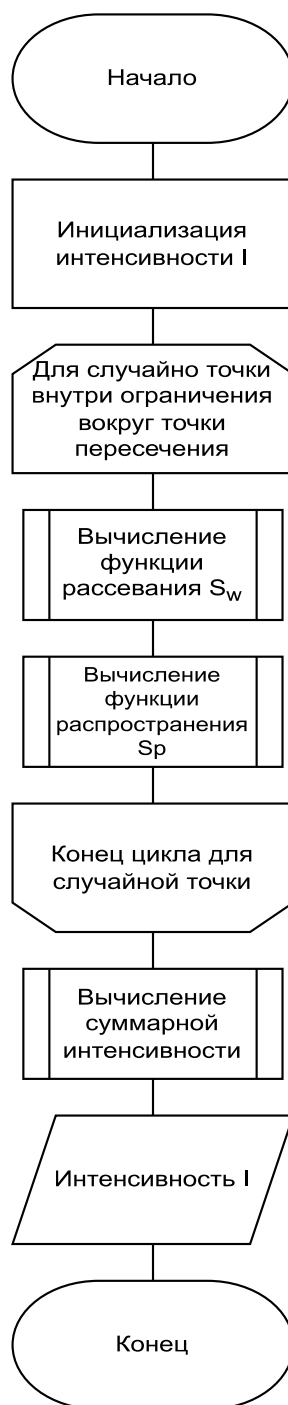


Рисунок 6: Схема вычисления функции подповерхностного рассеивания

## Заключение

Во время выполнения работы были решены следующие задачи:

- были выявлены неотъемлемые качества реалистического изображения;
- были рассмотрены и проанализированы существующие алгоритмы удаления невидимых линий, построения тени, отражения, преломления и эффекта глубины резкости;
- были выбраны и реализованы алгоритмы визуализации сцены.

## Список использованной литературы

- [1] Роджерс Д. Математические основы машинно графики / Д. Роджерс, Дж. Адамс. // М.: Мир, 1989
- [2] Physically Based Rendering: From Theory To Implementation [Электронный ресурс]. Режим доступа: <https://pbr-book.org> (дата обращения 18.09.22)
- [3] A Practical Model for Subsurface Light Transport [Электронный ресурс]. Режим доступа: <https://graphics.stanford.edu/papers/bssrdf/bssrdf.pdf> (дата обращения 18.09.22)
- [4] Approximate Reflectance Profiles for Efficient Subsurface Scattering [Электронный ресурс]. Режим доступа: <https://graphics.pixar.com/library/ApproxBSSRDF/paper.pdf> (дата обращения 18.09.22)