



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

---

ФАКУЛЬТЕТ «Информатика и системы управления»

---

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

---

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## *К КУРСОВОЙ РАБОТЕ*

### *НА ТЕМУ:*

*«Моделирование изображения объекта в  
неотполированном цветном зеркале»*

Студент ИУ7-53Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Авдейкина В. П.  
(И. О. Фамилия)

Руководитель курсовой работы

\_\_\_\_\_  
(Подпись, дата)

Новик Н. В.  
(И. О. Фамилия)

*2024 г.*

## РЕФЕРАТ

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>4</b>
<b>1 Аналитическая часть</b>	<b>5</b>
1.1 Формализация задачи синтеза изображения . . . . .	5
1.2 Физическая природа зеркальных поверхностей . . . . .	6
1.3 Существующие алгоритмы решения задачи . . . . .	8
1.3.1 Описание алгоритмов . . . . .	8
1.3.2 Сравнение алгоритмов . . . . .	10
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>13</b>

# ВВЕДЕНИЕ

Компьютерная графика в 21 веке не перестала быть развивающейся наукой и упоминается в [1], [2], [3], [4], [5]. Одной из стандартных задач компьютерной графики является синтез изображения [2], [6].

За последние 20 лет визуализация зеркальных поверхностей (зеркал) остается актуальной проблемой, о чем свидетельствует ее обсуждение в [7], [8], [9], [10].

Цель работы — разработка программного обеспечения для моделирования статической сцены с изображением предмета в неотполированном цветном зеркале.

Для достижения поставленной цели требуется решить следующие задачи:

- 1) проанализировать предметную область зеркальных поверхностей, выполнить формализацию задачи синтеза изображения в контексте моделирования статической сцены, рассмотреть известные методы и алгоритмы ее решения;
- 2) спроектировать программное обеспечение;
- 3) выбрать средства реализации и разработать программное обеспечение;
- 4) исследовать характеристики разработанного программного обеспечения.

# 1 Аналитическая часть

В данном разделе формализована задача синтеза изображения предмета в зеркале, проанализированы известные алгоритмы решения задачи синтеза.

## 1.1 Формализация задачи синтеза изображения

На рисунке 1 представлена диаграмма IDEF0 формализуемой задачи.

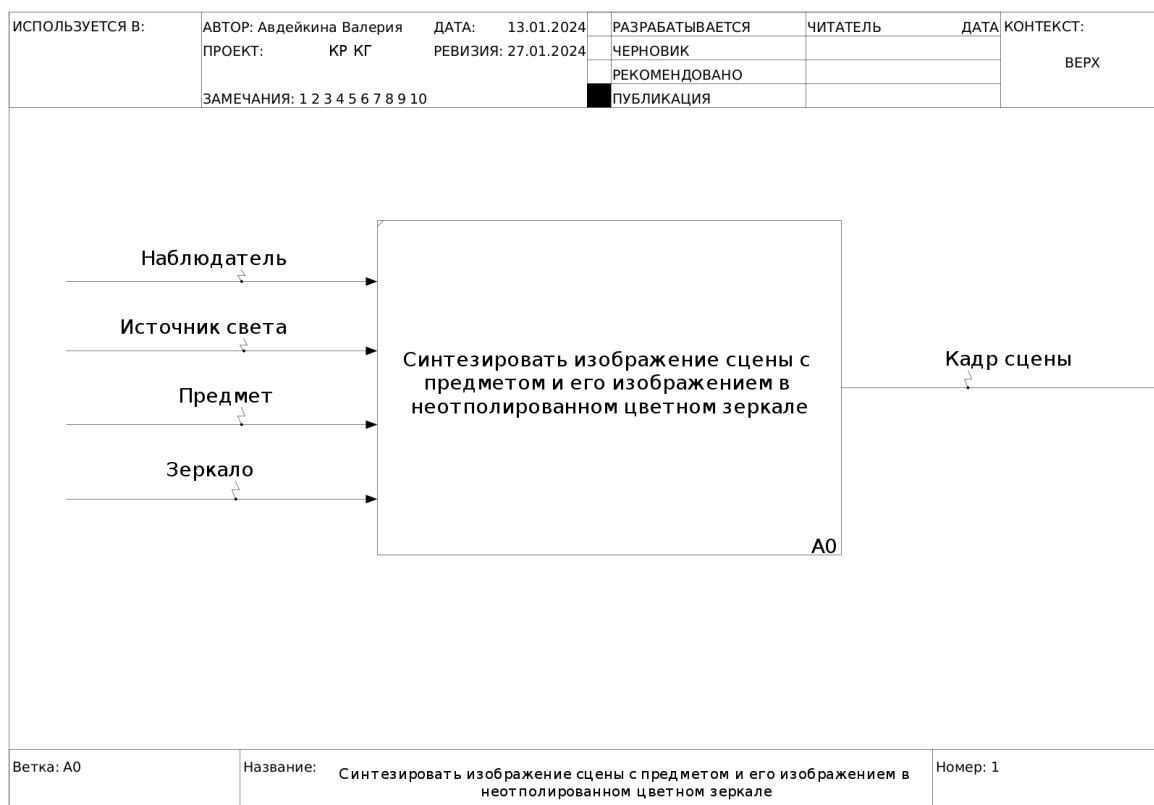


Рисунок 1 — Диаграмма IDEF0

*Модель* — отображение формы и размеров какой-либо сущности [6]. В данной работе используется поверхностное представление моделей, при котором они представляются аналитически (полностью или частично) [6]. Процесс получения модели называется *моделированием*.

Сцена состоит из набора сущностей:

- 1) *наблюдатель (камера)*, характеризующийся положением точки наблюдения в пространстве, направлениями взгляда и оси, направленной вверх [6];
- 2) *точечный источник света*, характеризующийся положением в пространстве, интенсивностью и цветом света [6];

- 3) *предмет* — модель геометрического тела, свойства (радиус, высота и так далее) которого определяются заданными пользователем параметрами;
- 4) *зеркало* — модель ограниченной поверхности, обладающая степенью полировки, цветом, радиусом и описанием границ.

Зеркало и предмет называются *объектами (сцены)*.

Модели, используемые в работе, требуют соблюдения физических законов.

## 1.2 Физическая природа зеркальных поверхностей

*Луч (световой)* — узкий пучок света, представленный геометрической линией, вдоль которой в определенном направлении распространяется свет [11], [12].

В геометрической оптике полагается, что луч распространяется прямолинейно до тех пор, пока не встретится отражающая поверхность или граница среды преломления [2].

Если луч падает на поверхность в точку  $A$  и отражается от нее, а через эту точку к поверхности проведена нормаль  $n$ , то углы, заключенные между нормалью и направлениями падающего, отраженного лучей ( $a$ ,  $b$ ), называются углами *падения*, *отражения* соответственно [11], [12], [13]. Описанные элементы представлены на рисунке 2.

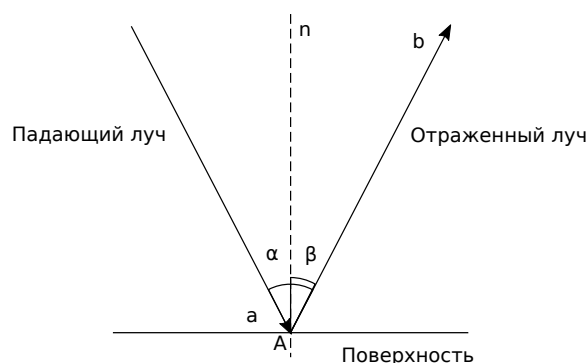


Рисунок 2 — Отражение луча от поверхности

*Зеркало* — поверхность, от которой отражаются лучи.

*Зеркальным отражением* называется такое отражение, для которого справедлива формула 1.1:

$$\alpha = \beta, \quad (1.1)$$

где  $\alpha$  — угол падения,  $\beta$  — угол отражения, лучи и нормаль находятся в одной плоскости [2], [12], [13].

Отражение луча света от некоторого объекта сцены представлено на рисунке 3. Рисунок 3 содержит следующие обозначения:  $\vec{N}$  — нормаль к поверхности в точке падения луча,  $\vec{V}$ ,  $\vec{L}$ ,  $\vec{R}$  — векторы, обозначающие направления от точки объекта к наблюдателю, обратное вектору наблюдения и отраженного луча соответственно,  $\alpha = \angle(\vec{L}, \vec{N})$ ,  $\beta = \angle(\vec{N}, \vec{R})$ ,  $\omega = \angle(\vec{R}, \vec{V})$ .

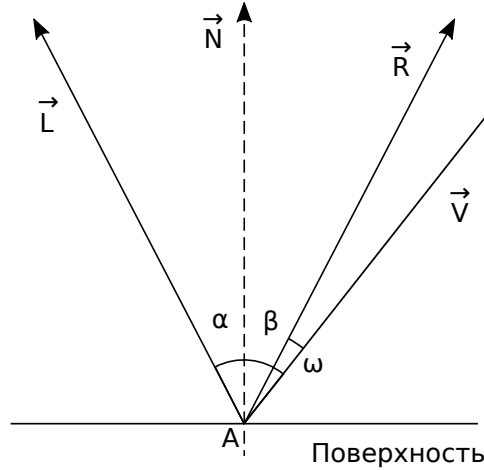


Рисунок 3 — Отражение луча света от некоторого объекта сцены

*Идеальным зеркалом* называется такое зеркало, отраженный от которого свет наблюдатель с направлением взгляда  $-\vec{V}$  сможет увидеть только в том случае, если угол  $\omega$  равен нулю, а для  $\alpha$  и  $\beta$  выполняется формула 1.1 [14].

Для зеркал, которые не являются идеальными, отражение описывается моделью Фонга, согласно которой интенсивность  $I_s$  отраженного света представлена в виде формулы 1.2 [2], [14]:

$$I_s = I_p K_s \cos^n \omega, \quad (1.2)$$

где  $K_s$  — коэффициент, учитывающий свойства объекта параметром зеркального отражения,  $n$  — числовой коэффициент, связанный со скоростью убывания интенсивности отраженного света,  $I_p$  — интенсивность точечного источника света.

*Диффузным отражением* называется такое отражение, при котором равенство из формулы 1.1 не выполняется и происходит равномерное по всем направлениям рассеивание отраженного света [2], [14]. Его интенсивность  $I_d$  можно найти по закону Ламберта, выраженному формулой 1.3 [14]:

$$I_d = I_p K_d \cos \alpha, \quad (1.3)$$

где  $K_d$  — коэффициент, учитывающий свойства объекта параметром диффузного отражения,  $\cos \alpha = (\vec{L}, \vec{N})$ .

Согласно модели Уиттеда интенсивность  $I$  некоторой точки с некоторыми оптическими свойствами определяется суммарной интенсивностью по формуле 1.4 с использованием формул 1.2, 1.3 [2], [14]:

$$I = I_a K_a C + I_t K_t + I_p K_d \cos(\vec{L}, \vec{N}) + I_p K_s \cos^n \omega, \quad (1.4)$$

где  $I_t$ ,  $I_a$  — интенсивности преломленного луча и фонового света,  $K_t$ ,  $K_a$  — коэффициенты, учитывающие свойства объекта параметрами прозрачности и фоновой подсветки,  $C$  — заданный цвет точки.

### 1.3 Существующие алгоритмы решения задачи

Среди известных алгоритмов визуализации поверхностей существуют такие алгоритмы, как выбрасывание лучей (raycasting), пошаговое распространение лучей (ray marching), обратная трассировка лучей.

#### 1.3.1 Описание алгоритмов

*Обратная трассировка лучей* представлена авторами [2], [14] с помощью этапов:

- 1) от наблюдателя в каждую точку (пиксель) экрана проводятся первичные лучи;
- 2) один из лучей достигает объекта;
- 3) луч преломляется или отражается, то есть порождает вторичные лучи;
- 4) обработка луча прекращается, когда он перестает пересекать объекты сцены.

Интенсивность некоторой точки объекта в описанном алгоритме вычисляется по формуле 1.4.

Пример визуализации обработки луча в алгоритме обратной трассировки представлен на рисунке 4.



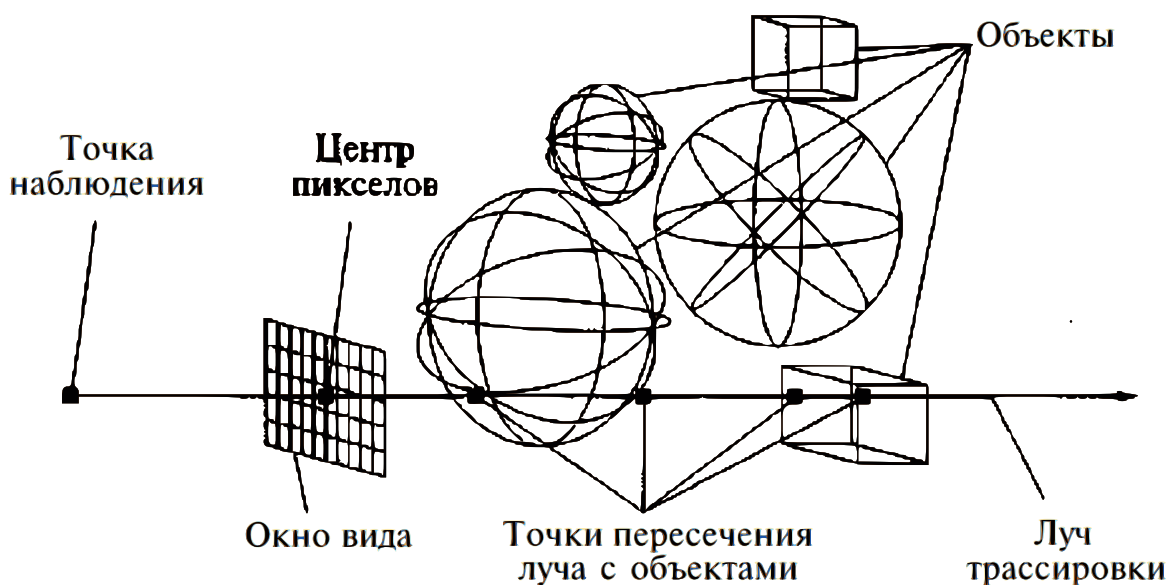


Рисунок 4 — Визуализация обработки луча в алгоритме обратной трассировки [15]

В соответствии с [16] *алгоритм выбрасывания лучей (raycasting)* позволяет обрабатывать поверхностные модели, описываемые с помощью карт, и представляется в виде последовательности шагов (выполняемыми для каждого шага угла обзора):

- 1) выпустить луч из точки наблюдения;
- 2) рассчитать расстояние от места испускания луча до каждого из объектов сцены;
- 3) определить, для какого из них оно является наименьшим;
- 4) выполнить визуализацию вертикальной «полосы» этого объекта.

Пример визуализации процесса обработки 2D карты алгоритмом выбрасывания лучей представлен на рисунке 5.

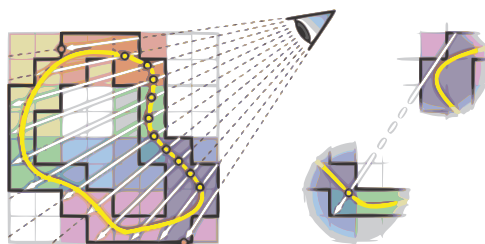


Рисунок 5 — Визуализация процесса обработки 2D карты алгоритмом выбрасывания лучей (raycasting) [17]

Алгоритм пошагового распространения лучей (*ray marching*) описывается в [18] для каждого луча, испускаемого из точки наблюдения через пиксели экрана, следующими действиями:

- 1) рассчитать значения функции расстояния со знаком до каждого объекта сцены относительно текущей точки начала распространения луча;
- 2) выбрать наименьшее из этих значений;
- 3) если выбранное расстояние меньше некоторой заданной малой величины, запустить рекурсивное распространение лучей из точки объекта в соответствии с оптическими свойствами его поверхности;
- 4) если выбранное расстояние больше некоторой заданной большой величины, визуализировать фон;
- 5) принять за следующую точку начала ту, которая находится на выбранном расстоянии по направлению распространения исходного луча, перейти на шаг 1.

Пример визуализации обработки одного луча в алгоритме пошагового распространения представлен на рисунке 6.

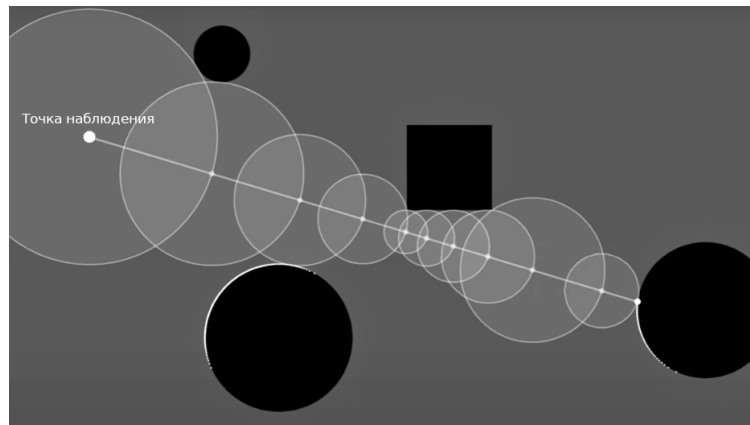


Рисунок 6 — Визуализация обработки одного луча в алгоритме пошагового распространения [18]

### 1.3.2 Сравнение алгоритмов

На основе источников, которые были использованы для приведенного выше описания, осуществлено сравнение анализируемых алгоритмов.

В таблице 1 приведены результаты сравнения, критерии которого расположены по горизонтали и включают в себя:

- 1) сложность алгоритма в зависимости от чисел пикселей  $C$  и объектов  $N$ ;
- 2) рабочее пространство алгоритма (сцены — «С», экранное — «Э»);
- 3) форма моделей, которая может быть использована при использовании алгоритма (каркасная — «К», поверхностная — «П», объемная — «О»);
- 4) возможность совмещения алгоритма и модели освещения Уиттеда;
- 5) принадлежность обрабатываемых точек объекту,
- 6) наличие обработки только тех первичных лучей, которые образуют текущее изображение на экране.

Таблица 1 — Сравнение существующих алгоритмов решения задачи синтеза изображения в контексте моделирования статической сцены

	1	2	3	4	5	6
Выбрасывание лучей	$O(C \cdot N)$	Э	П	Нет	Да	Да
Пошаговое распространение лучей	$O(C \cdot N)$	Э	О	Да	Нет	Да
Обратная трассировка лучей	$O(C \cdot N)$	Э	О	Да	Да	Да

## Вывод

В данном разделе была выполнена формализация объектов сцены и задачи синтеза изображения предмета в неотполированном цветном зеркале в контексте моделирования статической сцены, проанализированы известные алгоритмы решения этой задачи, предметная область зеркал.

В качестве основного алгоритма выбрана обратная трассировка лучей с использованием модели освещения Уиттеда.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Залогова Л. А.* Компьютерная графика // Элективный курс: Практикум. — М., 2005.
2. *Порев В.* Компьютерная графика // СПб.: БХВ-Петербург. — 2002. — Т. 432. — С. 3.
3. *Большаков В. П., Тозик В. Т., В Ч. А.* Инженерная и компьютерная графика. — БХВ-Петербург, 2013. — С. 288.
4. *Митин А. И., Свертилова Н. В.* Компьютерная графика : справочно-методич. пособие. — М.–Берлин : Директ-медиа, 2016. — С. 251.
5. *Турлюн Л. Н.* Компьютерная графика как особый вид современного искусства. — Издательство АлтГУ, 2014.
6. *Куров А. В.* Компьютерная графика : неопубликованный конспект лекций. — 2023.
7. Realistic materials in computer graphics / Н. Р. Lensch [и др.] // ACM SIGGRAPH 2005 Courses. — 2005.
8. *Reshetouski I., Ihrke I.* Mirrors in computer graphics, computer vision and time-of-flight imaging // Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications: Dagstuhl 2012 Seminar on Time-of-Flight Imaging and GCPR 2013 Workshop on Imaging New Modalities. — Springer. 2013. — С. 77—104.
9. *Miguel A. L., Nogueira A. C., Goncalves N.* Real-time 3D visualization of accurate specular reflections in curved mirrors a GPU implementation // 2014 International Conference on Computer Graphics Theory and Applications (GRAPP). — IEEE. 2014. — С. 1—8.
10. Demonstration of light reflection concepts for rendering realistic 3D tree images / Т. Hiranyachattada [и др.] // Journal of Physics: Conference Series. Т. 2145. — IOP Publishing. 2021. — С. 12—74.
11. *Тюрин Ю. И., Чернов И. П., Крючков Ю. Ю.* Физика. Оптика: учебник. — Томск : Издательство Томского политехнического университета, 2009. — С. 240.

12. *Ландсберг Г. С.* Оптика. Учебное пособие: Для вузов. — М : ФИЗМАТ-ЛИТ, 2003. — С. 848.
13. *Родионов С.* Основы оптики // СПб.: СПб ГИТМО (ТУ). — 2000.
14. *Демин А. Ю.* Основы компьютерной графики: учебное пособие. — Томск : Издательство Томского политехнического университета, 2011. — С. 191.
15. *Божко А. Н., Жук Д. М., Маничев В. Б.* Компьютерная графика : учебное пособие для вузов. — М. : изд-во МГТУ им. Н.Э. Баумана, 2007. — С. 392.
16. *Евстратов В. В.* Создание программы визуализации псевдотрехмерного изображения с помощью рейкастинга // Молодой ученый. — 2020. — № 50. — С. 12—15.
17. The ray casting engine and ray representatives / J. L. Ellis [и др.] // Proceedings of the first ACM symposium on solid modeling foundations and CAD/CAM applications. — 1991. — С. 255—267.
18. *Bredenbals A.* Visualising Ray Marching in 3D / Bredenbals Anton. — Groningen : University of Groninge, 2022. — Режим доступа: <https://fse.studenttheses.ub.rug.nl/id/eprint/27977> (дата обращения: 05.12.2023).