



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» _____

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

Студент _____ Аскарян Каринэ Артуровна _____
фамилия, имя, отчество

Группа _____ ИУ7-42Б _____

Тип практики _____ технологическая _____

Название предприятия _____ МГТУ им. Н. Э. Баумана, каф. ИУ _____

Студент _____ Аскарян К. А. _____
подпись, дата *фамилия, и.о.*

Руководитель практики _____ Исаев А. Л. _____
подпись, дата *фамилия, и.о.*

Оценка _____

2022 г.

**«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ
Заведующий кафедрой _____
« ____ » _____ 20 ____ г.

ЗАДАНИЕ
на прохождение производственной практики
технологическая
Тип практики

Студент

_____ Аскарян Каринэ Артуровна _____ 2 курса группы ИУ7-42Б
Фамилия Имя Отчество № курса индекс группы

в период с ____ . ____ .20 ____ г. по ____ . ____ .20 ____ г.

Предприятие: _____ МГТУ им. Н. Э. Баумана _____

Подразделение: _____ каф. ИУ7 _____
(отдел/сектор/цех)

Руководитель практики от предприятия (наставник):

(Фамилия Имя Отчество полностью, должность)

Руководитель практики от кафедры:

(Фамилия Имя Отчество полностью, должность)

Задание:

- 1. Начать разработку программного обеспечения для визуализации геометрических объектов, расположенных за полупрозрачной сферой.**
- 2. Решить вопрос о способе представления объектов, проанализировать и выбрать алгоритмы для их разработки**
- 3. Закрепить знания и навыки, полученные в ходе аудиторных занятий по пройденным курсам.**

Дата выдачи задания « ____ » _____ 20 ____ г.

Руководитель практики от предприятия _____ / _____ /

Руководитель практики от кафедры _____ / _____ /

Студент _____ / _____ /

Содержание

Введение	2
1. Аналитическая часть	3
1.1. Формализация объектов сцены	3
1.2. Обзор способов задания трехмерной модели	3
1.3. Обзор алгоритмов построения изображения	4
1.4. Обзор моделей освещения	7
1.5. Итоговый выбор алгоритмов и методов	8
2. Конструкторская часть	10
2.1. Требования к программному обеспечению	10
2.2. Описание алгоритма трассировки лучей	10
2.3. Освещение	11
2.4. Схема алгоритма	13
3. Технологическая часть	16
3.1. Выбор технологии программирования	16
3.2. Выбор языка программирования	16
3.3. Выбор среды разработки программирования	16
3.4. Примеры работы программы	17
Заключение	19
Список литературы	20

Введение

Компьютерная графика является неотъемлемой частью жизни человека. Она обладает высоким спросом: её используют в сфере развлечений, например, эффекты в кинофильмах и сериалах, анимации в играх; в научных и инженерных дисциплинах, таких как, медицина, геофизика, ядерная физика, картография, для визуализации и лучшего восприятия информации. Компьютерную графику так же можно использовать в школе на уроках физики, информатики, математики для демонстрации различных явлений.

Прогресс компьютерной графики не стоит на месте: со временем развиваются и оптимизируются алгоритмы, которые позволяют получить реалистичное трехмерное изображение. Вместе с прогрессом растут и требования к реалистичности, что приводит к росту как сложности алгоритмов, так и к затратам ресурсов компьютера, таких как оперативная память и время работы алгоритма.

Целью практики является разработка программного обеспечения визуализации геометрических объектов, расположенных за полупрозрачной сферой.

Для достижения поставленной цели необходимо решить следующие задачи:

- проанализировать задачу и разбить ее на подзадачи;
- проанализировать методы решения выделенных подзадач;
- формализовать объекты сцены;
- выбрать и описать алгоритмы реализации поставленных подзадач;
- составить требования к программному продукту;
- выбрать язык программирования;
- выбрать технологию программирования;

Решение данной задачи актуально на занятиях по физике для изучения и демонстрации оптических свойств полупрозрачных объектов.

1. Аналитическая часть

1.1. Формализация объектов сцены

Сцена стоит из следующих объектов:

1. Прозрачная сфера: видимый объект, задается аналитически, коэффициент прозрачности и преломления указываются пользователем.
2. Геометрические объекты: видимые объекты из ограниченного списка вариантов, характеризующиеся следующими основными параметрами: координатами центра, цветом.
3. Камера: невидимый точечный объект, с помощью которого рассчитывается перспективное отображение объектов сцены.
4. Источник света: невидимый точечный объект, который описан тремя координатами положения и коэффициентом освещенности.

1.2. Обзор способов задания трехмерной модели

Модель – отображение форм и размеров объекта, чье основное предназначение состоит в том, чтобы правильно отобразить форму и размер конкретного объекта.

Используются три вида модели: каркасная, поверхностная и твердотельная.

Каркасная модель.

Это простейший вид модели, в котором задается минимум информации, а именно о вершинах и рёбрах объектов. Однако, у моделей такого вида есть один недостаток: она может некорректно передать представление о форме и размерах объекта.

Поверхностная модель.

Поверхность в такой модели может задаваться как аналитически, так и другим способом, например, задавать отдельные участки как поверхности того или иного вида, а сложные криволинейные поверхности можно представлять в упрощенном виде, используя, к пример, полигональную аппроксимацию.

Недостатком таких моделей является отсутствие информации о том, с какой стороны находится собственный материал, а с какой – пустота.

Твердотельная модель.

Отличие такой модели состоит в том, что она включает в себя информацию о том, с какой стороны находится материал. Чаще всего для этого задается направление внутренней нормали.

1.3. Обзор алгоритмов построения изображения

Рассмотрим возможные алгоритмы построения изображения.

Алгоритм Варнока

Алгоритм Варнока работает в пространстве изображений. К сожалению, единого представления данного алгоритма не существует, но все его варианты используются с применением когерентности больших областей изображения. В рамках Алгоритма Варнока рассматривается окно и решается то, пусто ли оно или нет, или его содержимое является достаточно простым для визуализации. Если же содержимое окна не является таковым, то необходимо разбивать окно до тех пор, пока области, которые содержат не один многоугольник, или размеры области не достигнут одного пикселя. После этого обзревается и находится значение координаты Z каждого многоугольника и визуализируется пиксель с максимальным значением данной координаты. Достоинством такого алгоритма является простым для понимания, но может потребоваться большое количество разбиений, что потребует много времени не только на отображение содержимого всех областей, но и на анализ.

Алгоритм, использующий Z -буфер

Данный алгоритм работает в пространстве изображений, используя буфер кадра для заполнения интенсивности каждого пикселя, здесь вводится некоторый Z -буфер (буфер глубины каждого пикселя).

Значение каждого нового пикселя, который нужно занести в буфер кадра, сравнивается с глубиной пикселя, занесенного в Z -буфер. Если сравнение

показывает, что новый пиксель расположен ближе к наблюдателю, то новое значение Z заносится в буфер и корректируется значение интенсивности.

Алгоритм, использующий Z -буфер крайне прост в своей реализации по сравнению с другими анализируемыми алгоритмами, также не тратится время на сортировку элементов сцены.

Но несмотря на его быстроедействие увеличиваются затраты по памяти при использовании данного алгоритма, запоминается информация по каждому пикселю изображения.

Вычислительная сложность данного алгоритма равна $O(n*m*k)$, где $m*n$ – количество пикселей в буфере кадра, k – количество полигонов.

Алгоритм Робертса

Данный алгоритм работает в объектном пространстве, в нем используется уравнения плоскостей, пересечением которых образовывается тело. Алгоритм Робертса может применяться для изображения большого множества выпуклых многоугольников одной сцены в качестве проволочной модели с удаленными невидимыми линиями.

Основные этапы данного алгоритма заключены в следующем:

- подготовка исходных данных;
- удаление линий, которые экранируются самим телом;
- удаление линий, которые экранируются другими телами;
- удаления тех линий пересечения тел, которые экранируются непосредственно самими телами, связанных отношением протыкания.

Данный алгоритм из-за трудоёмких вычислений не может являться эффективным, в этом заключается и большой его недостаток. К преимуществам можно отнести то, что здесь используются точные и мощные методы математики.

Метод бросания лучей.

В основе метода бросания лучей (ray casting) лежит принцип обратимости световых лучей. Из камеры на каждый пиксель испускается один луч, и находится ближайший объект, который блокирует путь распространения этого

луча. После вычисляется цвет точки попадания и заносится в соответствующий пиксель. Данный метод не позволяет синтезировать сцену с учётом отражения и естественной проекции теней.

Метод конечных элементов.

Метод конечных элементов (radiosity) основан на равновесии обмена света между объектами сцены или, другими словами, на законе сохранения энергии, и разбиении поверхностей на маленькие участки, которые считаются локально плоскими. Освещенность находится решением системы линейных уравнений, описывающих обмен энергии между участками. При использовании все меньших участков результат будет стремиться к реальной физической модели. Алгоритм является довольно медленным, из-за расчета для множества маленьких участков поверхностей. Подобные алгоритмы имеют проблемы с представлением резких теней.

Алгоритм обратной трассировки лучей.

В основе алгоритма обратной трассировки лучей лежит принцип обратимости световых лучей, то есть вместо просчёта всех лучей испускаемых из источников сцены, можно изначально испускать первичные лучи из камеры, что значительно повышает эффективность визуализации. При попадании первичного луча в объект вычисляется цвет точки попадания. Для этого создаются вторичные лучи в зависимости от типа поверхности объекта.

Вывод

Главный алгоритм, используемый в программе для синтеза изображения, это алгоритм обратной трассировки лучей. Выбор был сделан в его пользу, так как в сравнении с его аналогами он даёт приемлемый результат за небольшое время синтеза. Алгоритм можно модернизировать, добавив в него параллельные вычисления для уменьшения времени синтеза сцены. Также он позволяет строить качественные тени с учетом большого числа источников света. Достоинством алгоритма является то, что он не требователен к памяти. А недостатком является то, что он не позволяет строить изображения в реальном времени.

1.4. Обзор моделей освещения

Модели освещения разделяются на глобальные и локальные. Локальная модель освещения является самой простой. Она рассматривает расчет освещенности самих объектов, не учитывая процессы светового взаимодействия между объектами, кроме этого взаимодействие ограничивается однократным отражением света от непрозрачной поверхности.

Глобальная модель освещения рассматривает трехмерную сцену как единую систему, описывая освещение с учетом взаимного влияния объектов, а так же учитывая многократное отражение и преломление света.

Модель освещения Ламберта.

Модель освещения Ламберта моделирует идеальное диффузное отражение от поверхности. Это означает, что свет при попадании на поверхность рассеивается равномерно во все стороны с одинаковой интенсивностью.

$$I = I_A + I_m \frac{\langle N, L \rangle}{|N||L|},$$

где I_m – интенсивность источника m ,

I_A – интенсивность окружающего освещения,

L – направление на источник света,

N – нормаль в данной точке.

$$I_\alpha = I_0 \cos(\alpha)$$

Данное выражение отражает зависимость между силой света, излучаемого плоской рассеивающей поверхностью в каком-либо направлении от угла между этим направлением и перпендикуляром к поверхности. Оно означает, что сила света плоской поверхности (I_0) максимальна по перпендикуляру к ней, убывает с увеличением угла α и становится равной нулю в касательных к поверхности направлениях.

Модель Ламберта одна из самых простых моделей освещения, она очень часто используется в комбинации других моделей, практически в любой другой модели освещения можно выделить диффузную составляющую.

Модель освещения Фонга.

Модель Фонга (глобальная модель) – классическая модель освещения, которая представляет собой комбинацию диффузной составляющей (модели Ламберта) и зеркальной составляющей и работает таким образом, что кроме равномерного освещения на материале может еще появляться блик. Местонахождение блика на объекте, освещенном по модели Фонга, определяется из закона равенства углов падения и отражения.

$$I = I_A + I_m \frac{\langle N, L \rangle}{|N||L|} + I_m \left(\frac{\langle R, V \rangle}{|R||V|} \right)^\beta,$$

где I_m – интенсивность источника m ,

I_A – интенсивность окружающего освещения,

L – направление на источник света,

N – нормаль в данной точке,

$R = 2N\langle N, L \rangle - L$,

V – вектор направления на наблюдателя,

β – коэффициент блеска.

1.5. Итоговый выбор алгоритмов и методов

Для синтеза изображения был выбран алгоритм обратной трассировки лучей, так как он дает приемлемый результат за небольшое время в сравнении с аналогами. Данный алгоритм можно модернизировать, добавив в него параллельные вычисления для уменьшения времени синтеза сцены. Кроме этого, алгоритм обратной трассировки лучей позволяет строить качественные тени с учетом большого количества источников света.

Достоинством данного алгоритма являются небольшие затраты по памяти, однако, он не позволяет строить изображения в реальном времени.

В качестве модели освещения была выбрана модель освещения Фонга, так как аппроксимирует физическую модель освещения и не требует много вычислительных мощностей.

2. Конструкторская часть

2.1. Требования к программному обеспечению

Программа должна предоставлять следующие возможности:

- изменение коэффициента прозрачности и коэффициента преломления сферы;
- изменение цвета сферы;
- изменение положения наблюдателя;
- визуализация изображения на сцене;
- добавление объектов из заданного списка за сферу;
- изменение положения и цвета объектов, находящихся за сферой;
- изменение положения и характеристик источника света.

2.2. Описание алгоритма обратной трассировки лучей

Алгоритм обратной трассировки лучей работает в пространстве изображения. Предполагается, что сцена преобразована в это пространство.

Рассмотрим, как формируется изображение. В случае алгоритма обратной трассировки лучи отслеживаются не от источников, а от камеры, таким образом, трассируется определенное количество лучей, которое равно разрешению картинки.

Предположим, у нас есть камера (наблюдатель), который находится на положительной части оси Oz , и картинная плоскость, перпендикулярная оси Oz . По очереди проведем воображаемые лучи из камеры до объектов сцены через каждый пиксель раstra. Для каждого луча отслеживается траектория, чтобы определить какие именно объекты пересекаются с лучом, и есть ли вообще пересечения. Пересечение с максимальным значением z представляет видимую поверхность для данного пикселя.

Для определения цвета пикселя лучи испускаются из найденной точки пересечения луча наблюдения с объектом к каждому источнику света. Если на пути к источнику света луч пересекает другой объект, то свет от данного

источника света не учитывается в расчете цвета. Если для луча от наблюдателя не найдено объектов пересечения, то пиксель закрашивается цветом фона.

Для расчета преломлений луча, встретившего на своей траектории объект, используются физические законы, такие как равенство угла падения и отражения и закон Снеллиуса, рассчитывается направление преломленного луча, найденная точка пересечения становится новой точкой наблюдения и алгоритм испускания луча повторяется до достижения максимальной глубины рекурсивных погружений.

На рисунке 2.1 предоставлена визуализация работы алгоритма обратной трассировки лучей.

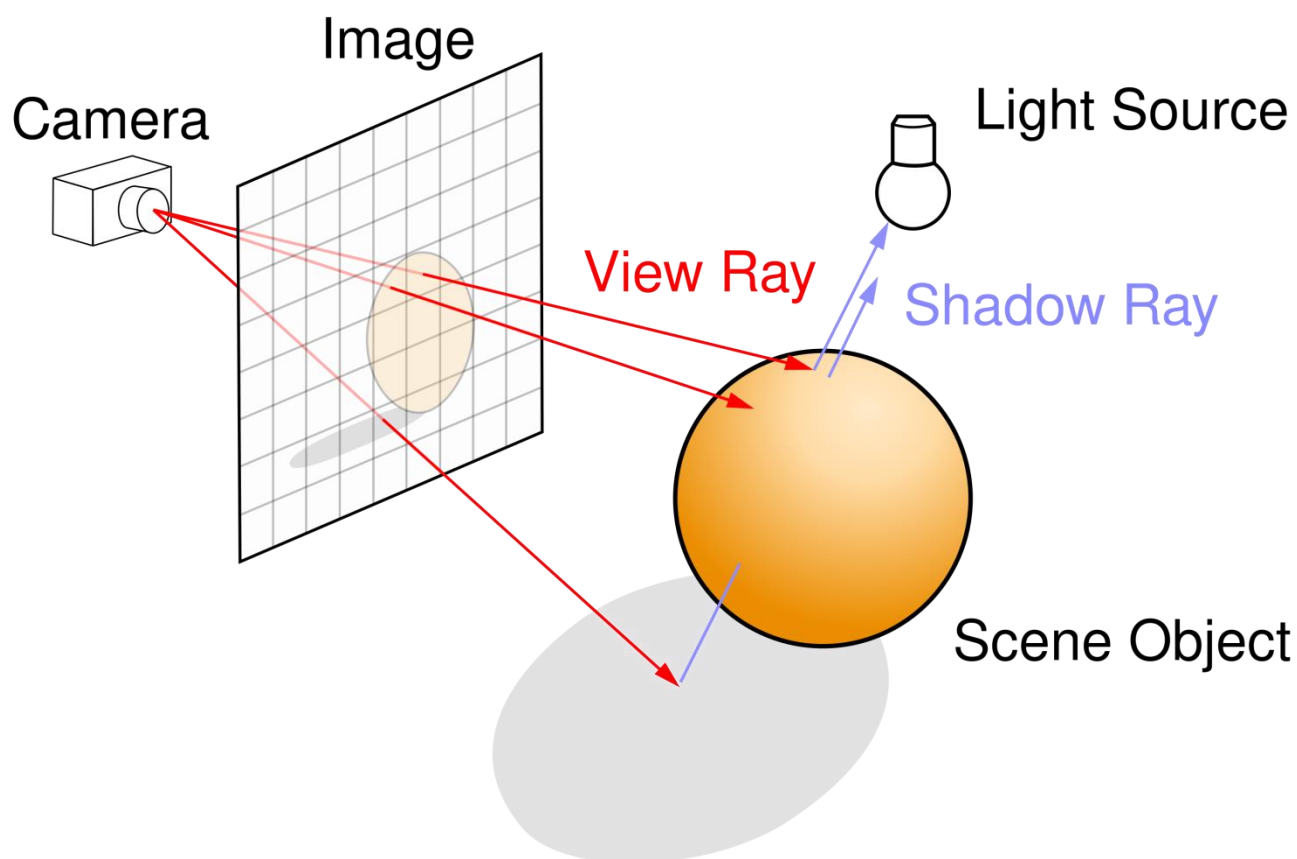


Рисунок 2.1 – визуализация алгоритма обратной трассировки лучей

2.3. Освещение

В качестве модели освещения была выбрана модель Фонга. В ней учитываются три составляющих:

1. Ambient (фоновое освещение) — константа, присутствует в любом участке сцены, не зависит от пространственных координат освещаемой точки и источника.
2. Diffuse (рассеянный свет) — рассчитывается по закону косинусов (закону Ламберта):

$$I_d = k_d \cos(L, N) i_d = k_d (L \cdot N) i_d,$$

где I_d — рассеянная составляющая освещенности в точке,

k_d — свойство материала воспринимать рассеянное освещение,

i_d — мощность рассеянного освещения,

L — направление из точки на источник,

N — вектор нормали в точке.

3. Specular (зеркальная составляющая) — зависит от того, насколько близко (насколько мал угол) находятся вектор отражённого луча и вектор к наблюдателю.

Модель Фонга представляет собой комбинацию модели Ламберта и зеркальной составляющей, поэтому при определенных обстоятельствах на материале будет появляться блик, расположение которого определяется из закона равенства углов падения и отражения.

Падающий, отраженный лучи и нормаль к отражающей поверхности лежат в одной плоскости. Нормаль делит угол между лучами на две равные части. Получается, что отраженная составляющая зависит от угла между направлением на наблюдателя и отраженным лучом.

$$I_s = k_s \cos^\alpha(R, V) i_s = k_s (R \cdot V)^\alpha i_s,$$

где I_s — зеркальная составляющая освещенности в точке,

k_s — коэффициент зеркального отражения,

i_s — мощность зеркального освещения,

R — направление отраженного луча,

V — направление на наблюдателя,

α — коэффициент блеска, свойство материала.

Тогда, если N — вектор нормали к поверхности в точке,

L – падающий луч,

R – отраженный луч,

V – вектор, направленный к наблюдателю,

k_α – коэффициент фонового освещения,

k_d – коэффициент диффузного освещения,

k_s – коэффициент зеркального освещения,

I_L – интенсивность источника,

d – расстояние от источника,

K – некоторая постоянная,

ρ – степень, аппроксимирующая пространственное распределение зеркально отраженного света, то верна следующая формула:

$$I_\alpha = k_\alpha \cdot I_\alpha + \frac{I_L}{d + K} \cdot (k_d(L, N) + k_s(R, V)^\rho)$$

На рисунке 2.2 изображен пример работы модели Фонга.

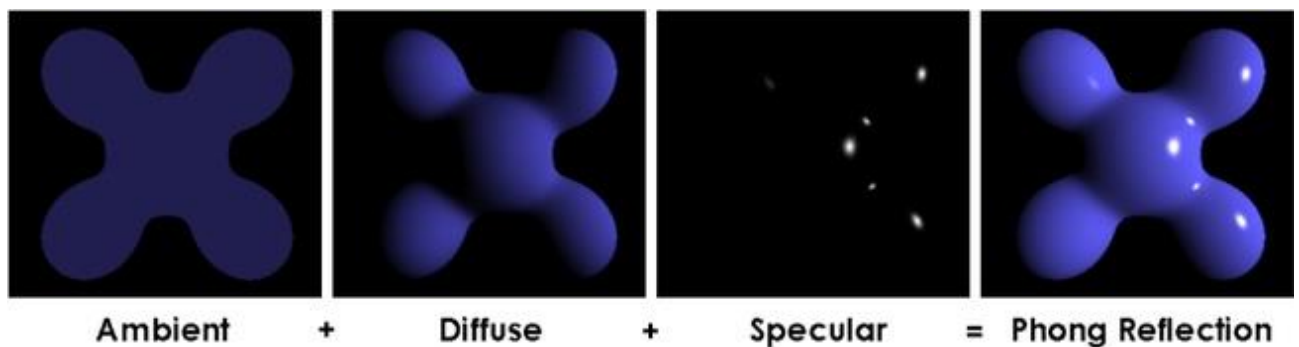


Рисунок 2.2 – пример работы модели Фонга

2.4. Схема алгоритма

На рисунках 2.3 – 2.4 приведена схема работы алгоритма.

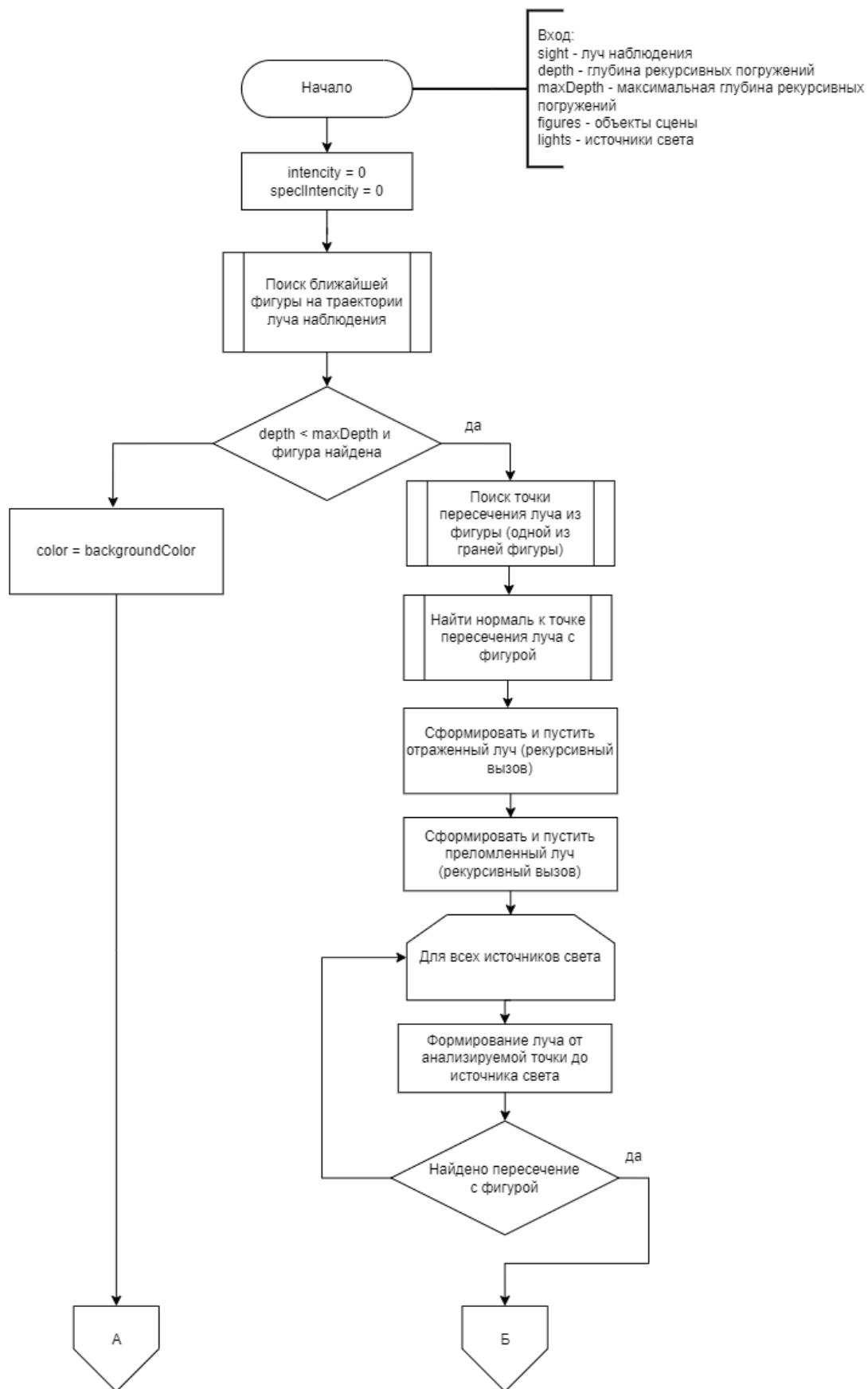


Рисунок 2.3 – схема алгоритма, часть 1.

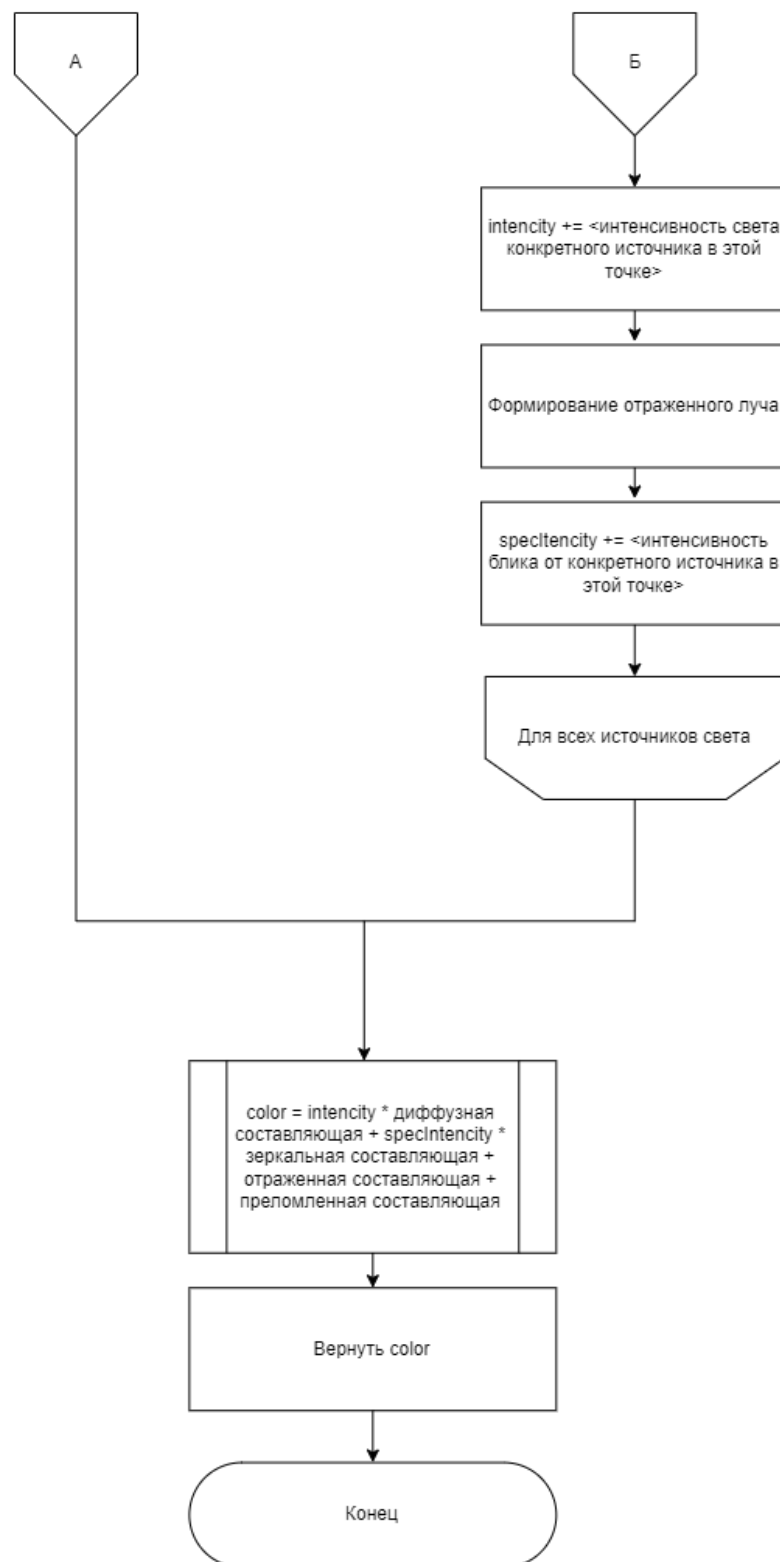


Рисунок 2.4 – схема алгоритма, часть 2

3. Технологическая часть

3.1. Выбор технологии программирования

Структурный и объектно-ориентированный подходы являются основными технологиями в программировании. Так как данная задача представляет собой сложную систему, было принято решение об использовании объектно-ориентированного подхода. Кроме того, в выбранном подходе есть принципы инкапсуляции, полиморфизма и наследования, которые позволяют создавать более гибкие и широко расширяемые системы в отличие от структурного подхода.

3.2. Выбор языка программирования

При написании программного продукта было решено задействовать язык C++. Этот выбор обусловлен следующими факторами:

- высокая вычислительная производительность, которая позволяет сократить время синтеза изображения;
- поддержка объектно-ориентированного программирования, что позволяет пользоваться шаблонами и паттернами проектирования;
- доступность – существует большое количество учебной литературы и статей по C++ на различных языках;
- по сравнению с C, в C++ больше готового функционала;
- Данный язык преподавался в рамках курса «Объектно-ориентированного программирования»

3.3. Выбор среды разработки программирования

Средой разработки была выбрана среда QtCreator, что обусловлено следующими причинами:

- поддержка расширения QtDesign, которая позволяет создавать красивый и надежный интерфейс;

- хороший и удобный отладчик, позволяющий легко выявлять ошибки в коде программного продукта;
- Основы разработки в данной среде преподавались еще в рамках курса «Программирования на Си»;
- хорошая совместимость с C++, так как фреймворк Qt написан на этом языке программирования

3.4. Примеры работы программы

На рисунках 3.1 – 3.3 приведены примеры работы программы.

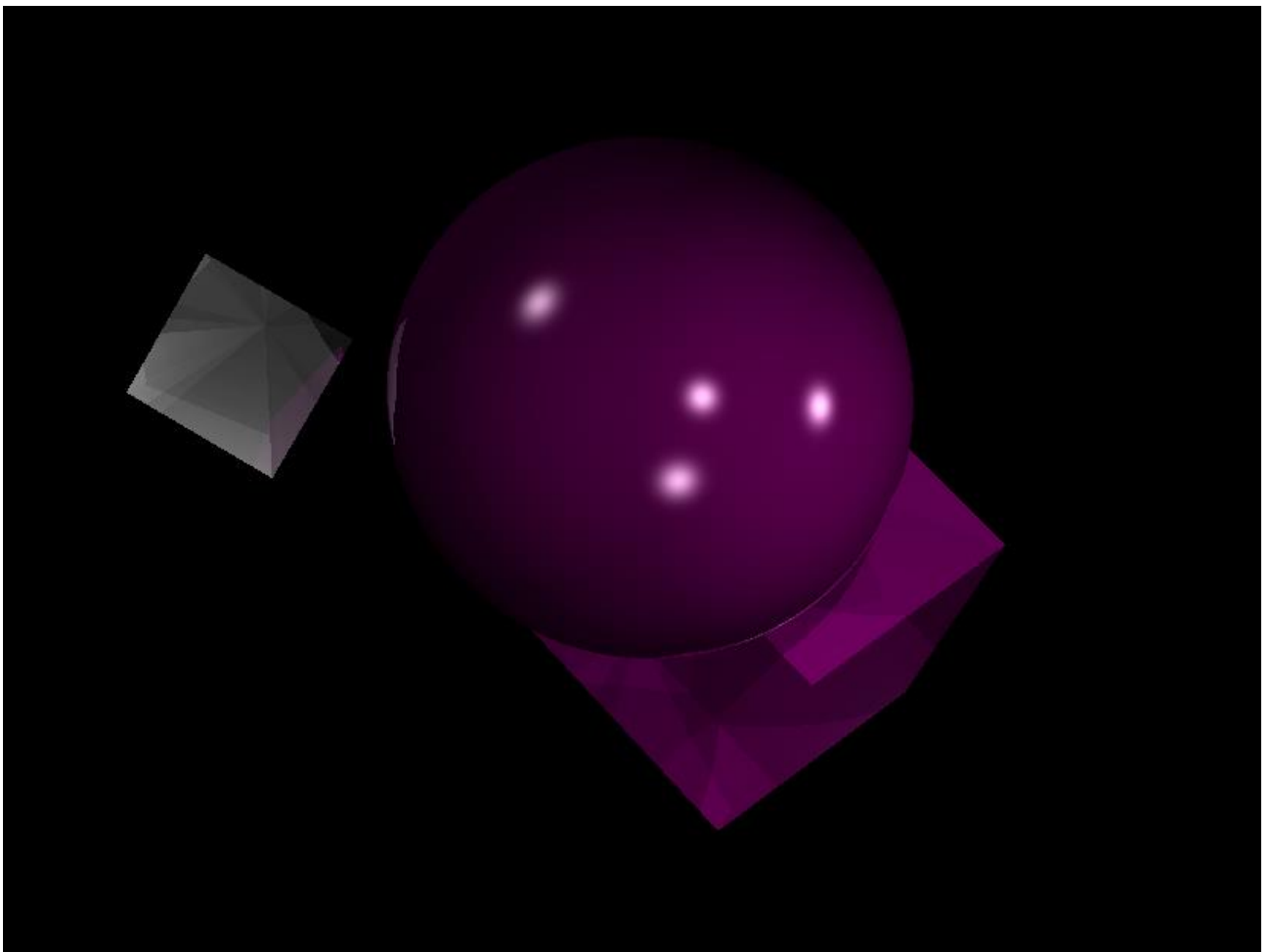


Рисунок 3.1 – сфера с коэффициентом прозрачности 0

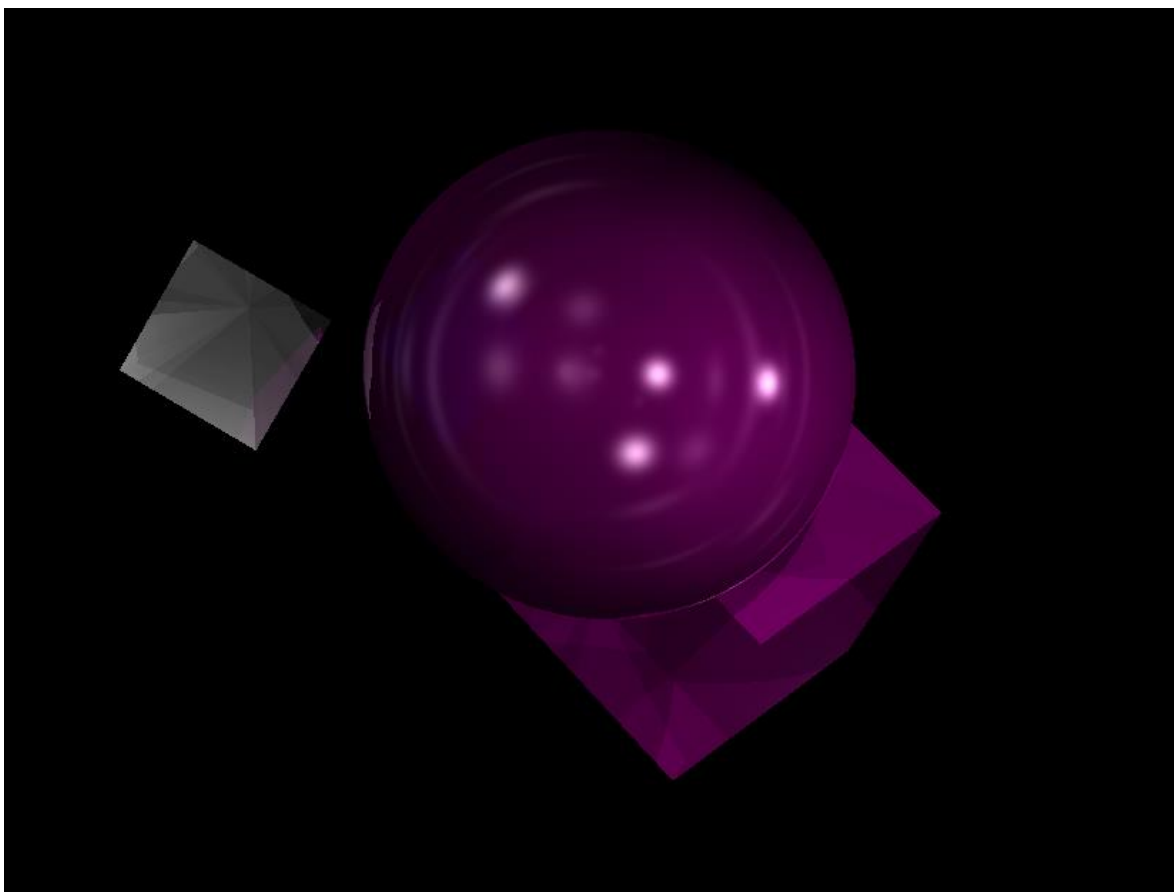


Рисунок 3.2 – сфера с коэффициентом прозрачности 0.2

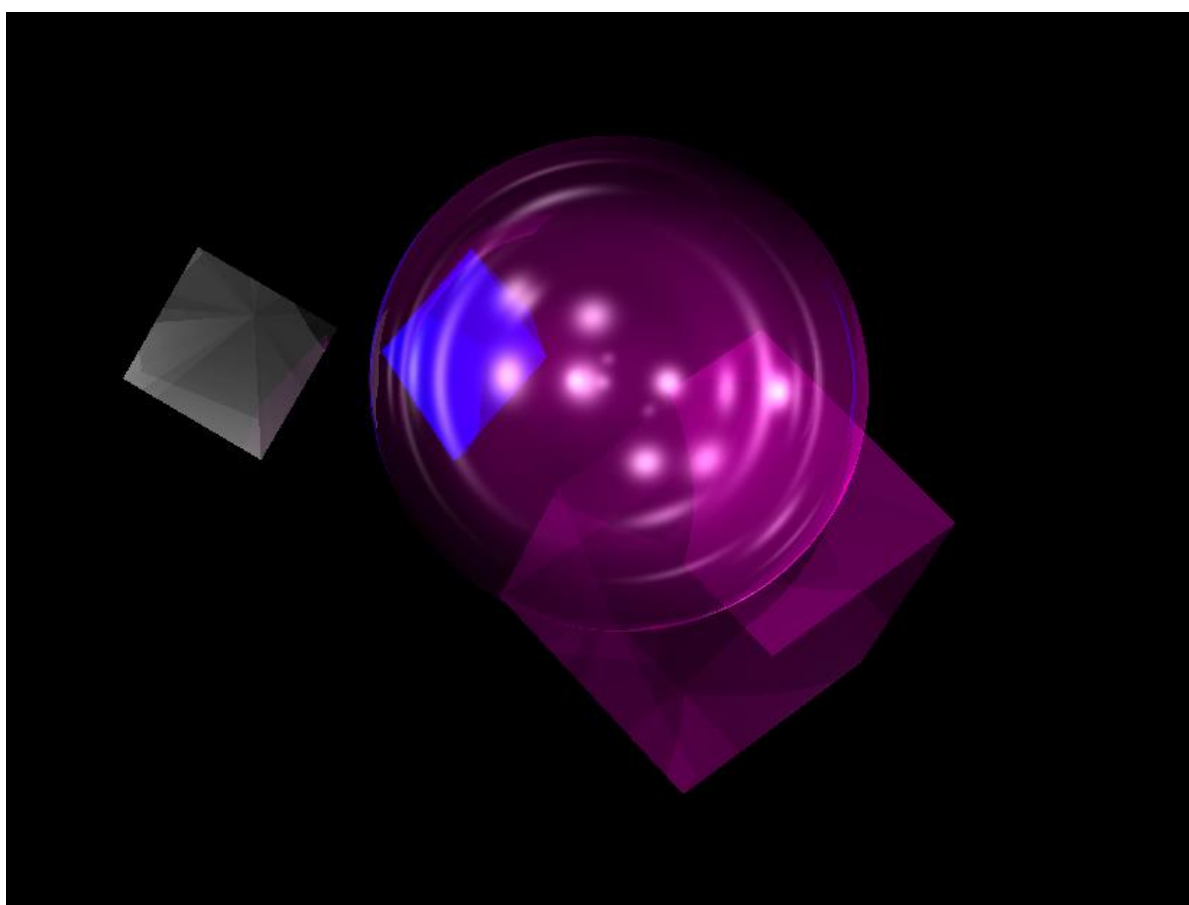


Рисунок 3.3 – сфера с коэффициентом прозрачности 0.9

Заключение

Во время выполнения технологической практики была начата разработка программного обеспечения для визуализации геометрических объектов за полупрозрачной сферой, решены все поставленные задачи.

Были рассмотрены способы задания моделей объекта, алгоритмы синтеза изображения и модели освещения. Были проанализированы их достоинства и недостатки и выбраны наиболее подходящие для решения поставленных задач.

Был разработан программный продукт, с помощью которого генерируется изображение с использованием алгоритма обратной трассировки лучей.

Данная работа помогла глубоко изучить возможности Qt Creator и получить более узкоспециализированные знания в области машинной графики.

Список литературы

1. Документация библиотеки Qt (Электронный ресурс). Режим доступа: <https://doc.qt.io> (дата обращения: 01.07.2022).
2. Документация языка C++ (Электронный ресурс). Режим доступа: <https://docs.microsoft.com/ru-ru/cpp/cpp/?view=msvc-170> (дата обращения: 01.07.2022).
3. Д. Роджерс «Алгоритмические основы машинной графики». Москва-Мир, 1989. с. 437.
4. Дебелов В.А., Валеев Т.Ф., Курс лекций по дисциплине «Инженерная и компьютерная графика».
5. Куров А.В., Курс лекций по дисциплине «Компьютерная графика».
6. Трассировка лучей в реальном времени. [Электронный ресурс]. Режим доступа: <https://www.ixbt.com/3dv/directx-raytracing.html> (дата обращения: 15.07.2022).
7. Простые модели освещения. [Электронный ресурс]. Режим доступа: <http://grafika.me/node/344> (дата обращения: 16.07.2022).