

Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
ИТМО»

Факультет Программной инженерии и компьютерной техники

Отчёт по лабораторной работе №4

Алгоритмы компьютерной графики

Выполнил: студент группы Р3314

Силинцев В.В.

Преподаватель: Потемин И.С.

Санкт-Петербург 2025

Содержание

Цель работы.....	3
Задание.....	4
Алгоритм разработанного приложения.....	5
Интерфейс пользователя.....	5
Алгоритм работы.....	5
Формула.....	5
Сохранение файла.....	6
Листинг программы.....	7
Результаты работы программы.....	8
Выводы.....	9

Цель работы

Овладеть навыками расчета и визуализации распределения яркости на диффузной сфере, освещенной точечными источниками света.

Задание

- Провести расчет распределения яркости на сфере в пределах заданной области.
- Рекомендуемые пределы значений параметров для расчёта:
 - Размер прямоугольного экрана по высоте (H) и ширине (W) варьируются в диапазоне от 100 до 10000 миллиметров.
 - Разрешение изображения по высоте (Hres) и ширине (Wres) варьируются в диапазоне от 200 до 800 пикселей. Разрешение должно обеспечивать квадратные пиксели.
 - Координаты источников света (x_L, y_L, z_L) [мм] по осям X и Y ± 10000 , по оси Z от 100 до 10000.
 - Координаты наблюдателя: $(0, 0, z_o)$ [мм].
 - Координаты центра сферы: (x_C, y_C, z_C) [мм] по осям X и Y ± 10000 , по оси Z от 100 до 10000. Сфера должна целиком помещаться в область видимости.
 - Сила излучения I_0 варьируются от 0.01 до 10000 Вт/ср.
 - Параметры модели Блинн-Фонга. Выбираются так, чтобы на изображении был виден ярко выраженный пик.
- Написать приложение на Python, формирующее изображение рассчитанного распределения яркости для заданного разрешения с нормировкой (0-255) на максимальное значение яркости.
- Записать сформированное изображение в файл.
- Визуализировать изображение на мониторе.

Алгоритм разработанного приложения

Интерфейс пользователя

Приложение позволяет изменять все основные параметры: размер области, разрешение изображения, координаты источников света и их силу излучения, координаты наблюдателя (по оси z), координаты центра сферы и её радиус, а также коэффициенты диффузного отражения, зеркального отражения и блеска. Ошибки ввода автоматически выводятся пользователю, что предотвращает некорректные расчёты.

Алгоритм работы

Для ввода параметров со стороны пользователя используется библиотека Tkinter для создания интерфейса. При нажатии на кнопку «Рассчитать» вызывается основная функция calculate(), которая с помощью вспомогательных функций проверяет корректность введённых данных. Если все параметры верны, осуществляется расчёт яркости.

Формула

Для расчёта яркости используется модель Блинн-Фонга:

$$I = I_a + \sum_{i=0}^N (I_d^{(i)} + I_s^{(i)}), \text{ где } I_d = k_d \cos(\vec{L}, \vec{N}) i_d, \quad I_s = k_s \cos^\alpha(\vec{N}, \vec{H}) i_s. \text{ (ссылка на источник:}$$

https://kpfu.ru/staff_files/F823558158/Lecture5.pdf).

Сохранение файла

Сформированное изображение можно сохранить на диск с помощью кнопки «Сохранить», которая вызывает функцию `save()`. Поддерживаются форматы PNG, JPEG и BMP.

Листинг программы

Полный исходный код приложения:

<https://github.com/vvlaads/Computer-graphics-4>

Результаты работы программы

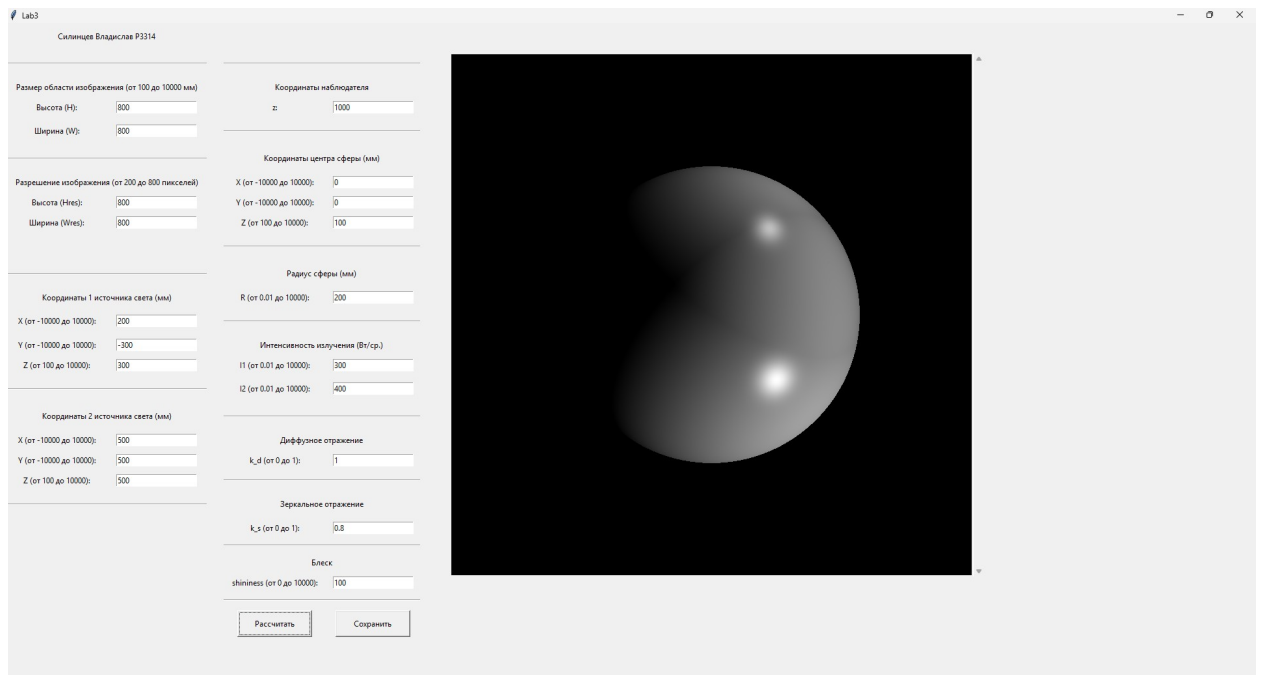


Рисунок 1: Пример работы приложения

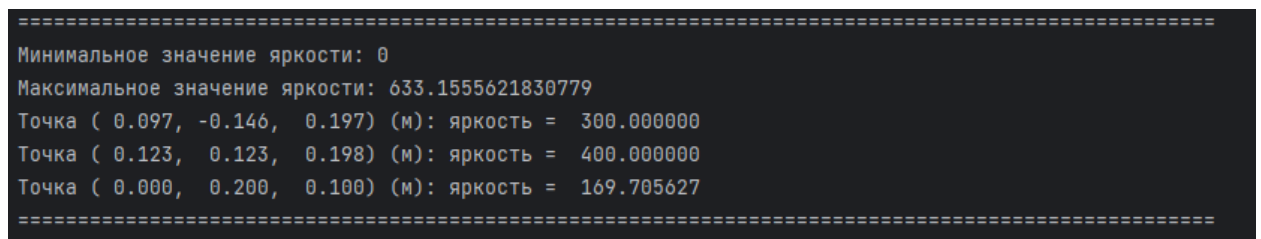


Рисунок 2: Пример работы приложения

Выводы

В ходе выполнения работы была реализована математическая модель расчёта распределения яркости на поверхности сферы с использованием модели Блинн-Фонга. Алгоритм учитывает координаты двух точечных источников света, их интенсивность, геометрические параметры сферы (координаты центра и радиус), положение наблюдателя, а также коэффициенты диффузного и зеркального отражения вместе с показателем блеска. Для каждого пикселя результирующего изображения производится расчёт пересечения луча наблюдателя со сферой, определяется нормаль в точке пересечения и вычисляется освещённость по модели Блинн-Фонга, после чего полученные значения яркости нормируются в диапазон 0–255 для формирования изображения.

Программа включает систему валидации входных параметров, которая проверяет соответствие введённых значений допустимым диапазонам, что предотвращает ошибки в расчётах. Пользовательский интерфейс, разработанный на Tkinter, позволяет интерактивно изменять все параметры сцены и визуализировать результат в реальном времени, а также сохранять итоговое изображение в файл стандартных графических форматов.