

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Основная часть	5
1.1 Формальная постановка задачи	5
1.2 Этапы метода	5
1.3 Ограничения метода	11
1.3.1 Ограничения на входные и выходные данные	11
1.4 Разработанное программное обеспечение	12
1.4.1 Требования к вычислительной системе	13
1.4.2 Основные модули	13
1.4.3 Руководство пользователя	13
1.4.4 Примеры работы	14
ЗАКЛЮЧЕНИЕ	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	17

ВВЕДЕНИЕ

Во время выполнения выпускной квалификационной работы был разработан метод наложение теней в дополненной реальности на основе информации о глубине точек кадра.

Зачастую методы наложения теней в дополненной реальности вычисляют положения ИС в каждом кадре, что требует немало вычислительных ресурсов [1]. В тех случаях, когда схема освещения окружения в основном не изменятся, это избыточно. Разработанный метод предназначен для таких случаев и требует вычислить положения ИС только в начале сессии или при необходимости, когда схема освещения коренным образом изменилась, что позволяет сократить использование вычислительных ресурсов, тратя их только на обработку компьютерной графики и построение геометрии окружения.

Цель работы – реализовать программное обеспечение, демонстрирующее практическую осуществимость спроектированного в ходе выполнения выпускной квалификационной работы метода.

Для достижения поставленной цели нужно решить следующие задачи:

- формализовать поставленную задачу в виде IDEF0-диаграммы;
- описать этапы реализуемого метода;
- ввести ограничения метода и на входные и выходные данные;
- разработать программное обеспечение для данного метода.

1 Основная часть

1.1 Формальная постановка задачи

На рисунке 1.1 представлена формализация задачи в виде IDEF0-диаграммы.

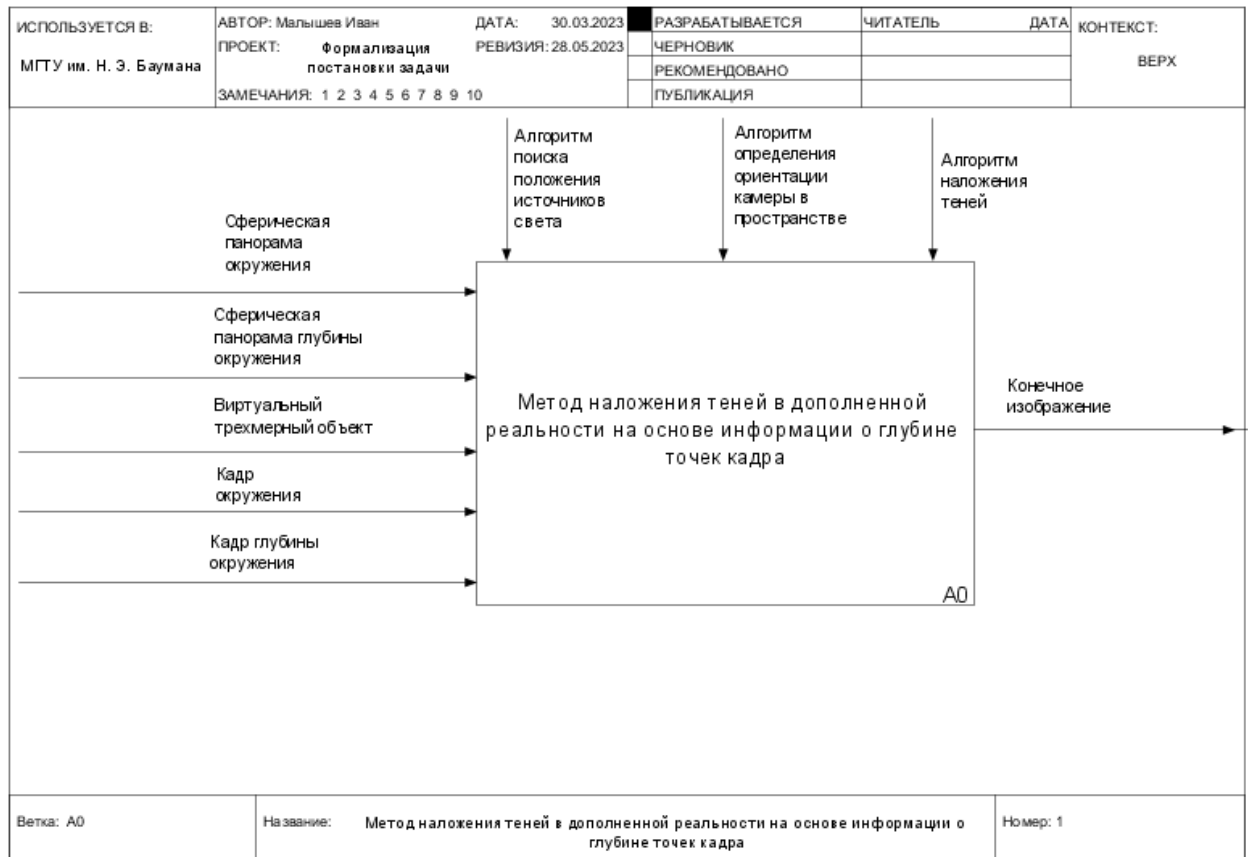


Рисунок 1.1 – Формализация задачи в виде IDEF0-диаграммы

1.2 Этапы метода

На рисунке 1.2 представлена IDEF0-диаграмма описанного метода.

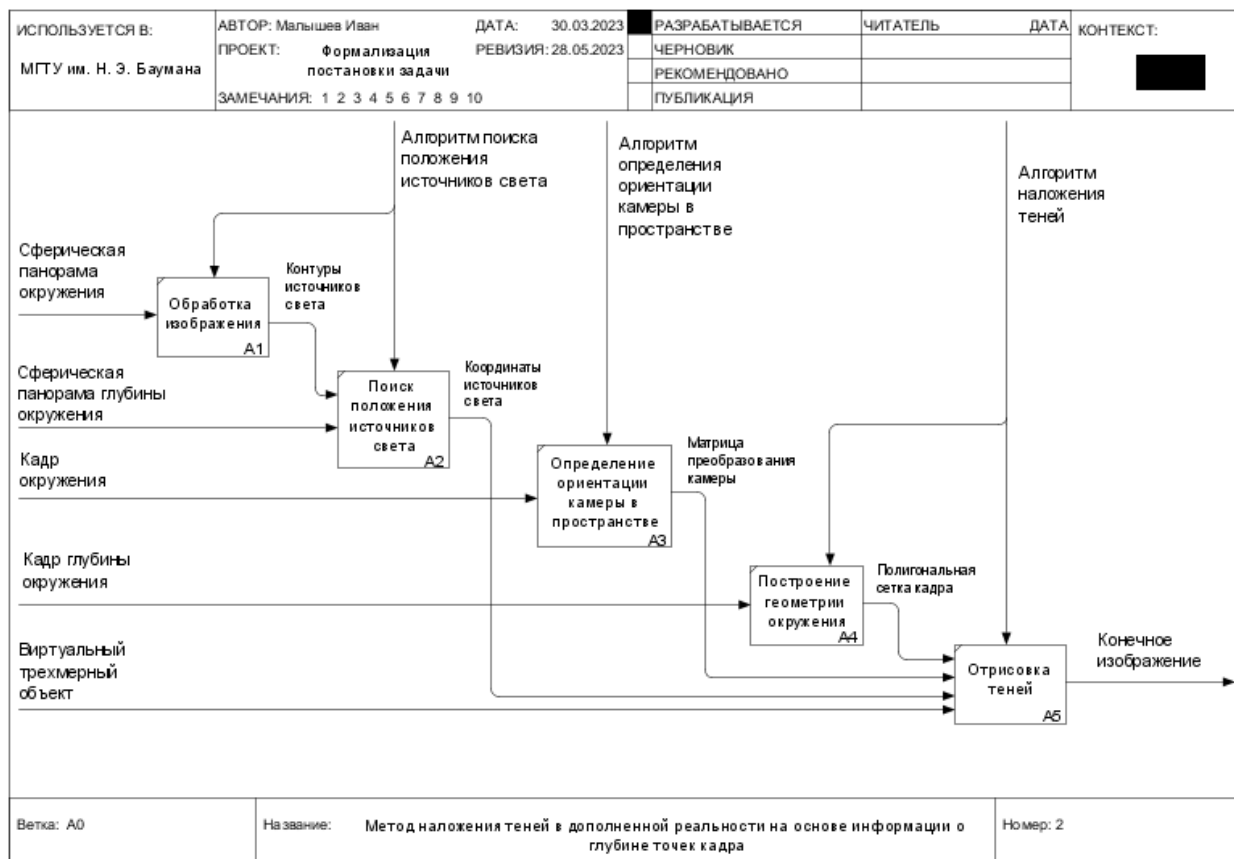


Рисунок 1.2 – IDEF0-диаграмма описанного метода

На рисунках 1.3-1.7 представлены схемы этапов разработанного метода.

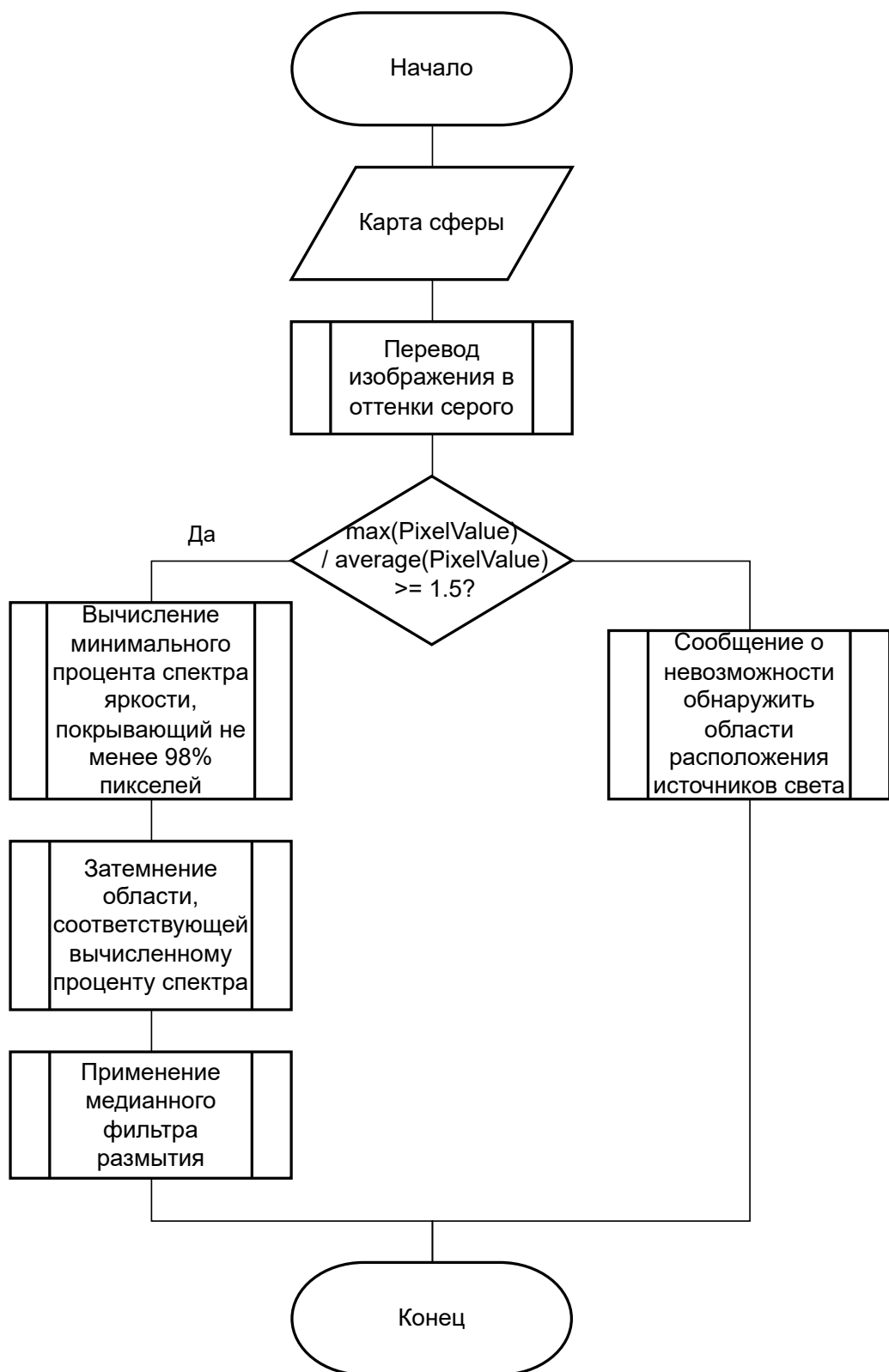


Рисунок 1.3 – Схема алгоритма обработки изображения

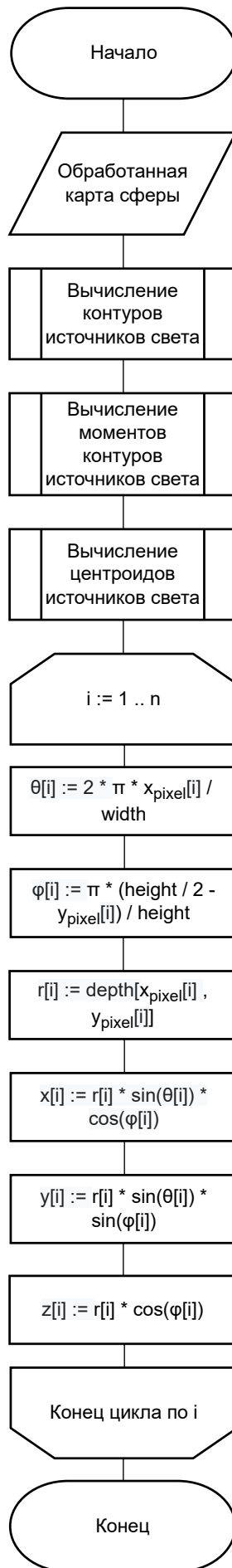


Рисунок 1.4 – Схема алгоритма поиска положения источников света

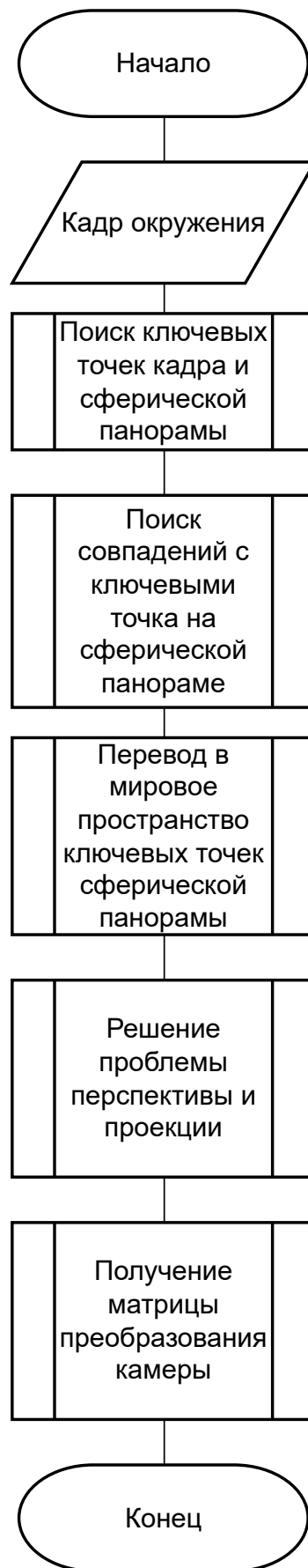


Рисунок 1.5 – Схема алгоритма определения ориентации камеры в пространстве

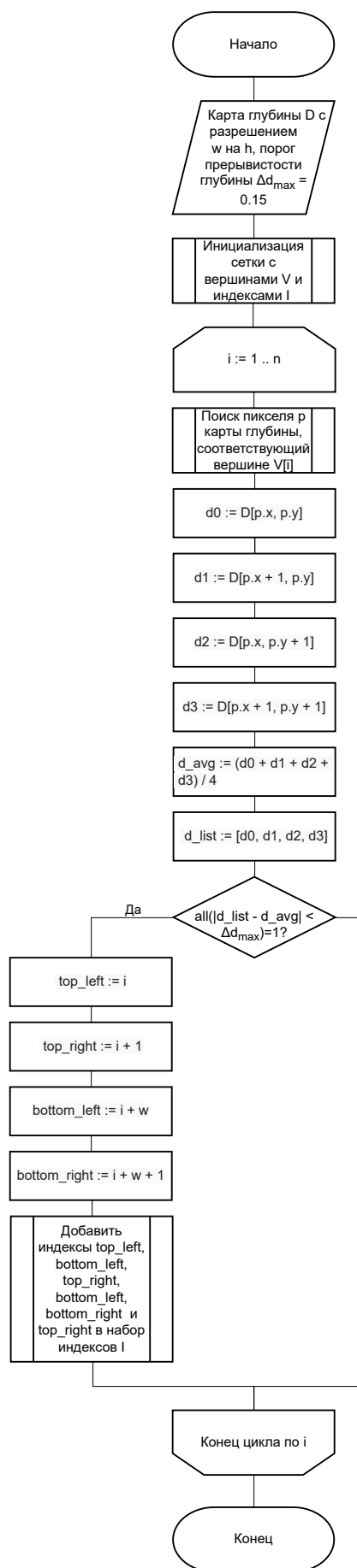


Рисунок 1.6 – Схема алгоритма построения геометрии окружения в реальном времени

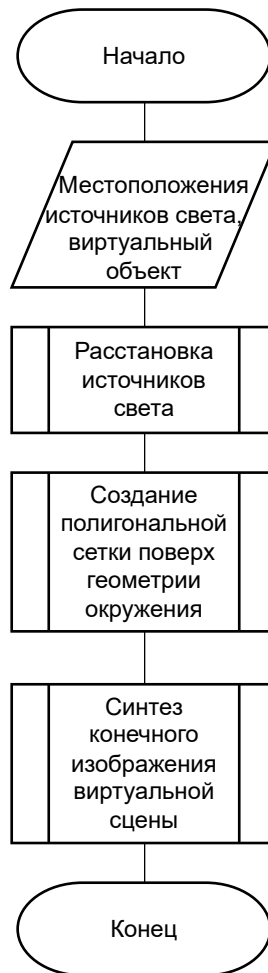


Рисунок 1.7 – Схема алгоритма отрисовки теней

1.3 Ограничения метода

Разработанный метод имеет следующие ограничения:

- метод применим только для работы в помещении;
- полигональная сетка окружения не учитывает свойства поверхности, такие как отражение поверхностью света и прозрачность.

Первое ограничение связано с тем, что на улице сложно однозначно определить положение источников света из-за рассеянного света [1].

1.3.1 Ограничения на входные и выходные данные

В таблице 1.1 приведено описание формата входных данных.

Таблица 1.1 – Описание входных данных

Данные	Формат
Сферическая панорама окружения	Цветное изображение формата PNG с глубиной цвета 32 бит
Сферическая панорама глубины окружения	Изображение в оттенках серого формата PNG с глубиной цвета 64 бит
Виртуальный трехмерный объект	Файл формата prefab
Кадр окружения	Цветное изображение
Кадр глубины окружения	Изображение в оттенках серого

Ограничения на входные данные:

- обе сферические панорамы должны быть с одинаковым разрешением;
- оба кадра окружения должны быть с одинаковым разрешением;
- минимальное расстояние от точки съемки окружения должно соответствовать минимальному расстоянию распознавания глубины используемой 3D-камеры;
- максимальное расстояние от точки съемки окружения должно соответствовать максимальному расстоянию распознавания глубины используемой 3D-камеры.

Выходными данными является цветное изображение, разрешение которого соответствует разрешению кадров окружения.

1.4 Разработанное программное обеспечение

Разработанное программного обеспечения представляет собой проект Unity [2], который состоит из набора сценариев на языке C# [3]. В качестве 3D-камеры используется Kinect [4].

1.4.1 Требования к вычислительной системе

Для работы реализованного программного обеспечения требуется:

- 3D-камера Kinect и Kinect SDK версии 1.8;
- Поддержка Visual C++ 2015;
- Поддержка .NET Standard 2.1.

Для работы с проектом реализованного программного обеспечения требуется Unity версии 2021.3.23f1.

1.4.2 Основные модули

Далее представлены основные модули разработанного программного обеспечения:

- LightPosCalc – предназначен для вычисления положения источников света;
- KinectManager – предназначен для взаимодействия с Kinect (получения кадра окружения и кадра глубины окружения);
- ExamplesManager – предназначен для работы с набором доступных моделей.

1.4.3 Руководство пользователя

В разработанном программном обеспечении существует система управления трехмерными моделями. После загрузки панорма и вычисления положения источников света загружается сцена, где пользователю доступно создание, удаление и управление положением и поворотом заранее predetermined набора моделей, представленные на рисунке 1.8.



Рисунок 1.8 – Набор доступных моделей

Создание моделей

Создание моделей производится посредством наведения курсора на некоторую часть экрана и нажатием клавиши на клавиатуре **1**, **2** или **3**, что добавляет на сцену модели параллелепипеда, кота и пальмы соответственно. Они создаются от камеры на некотором расстоянии, значение которого задается конфигурационным файлом.

Удаление моделей

Удаление моделей происходит с помощью клавиши **Delete**: нужно навести курсор на некоторую модель на сцене, нажать ЛКМ и нажать клавишу **Delete**.

Управление моделями

Для управления моделью нужно нажать ЛКМ на некоторую модель на сцене, после чего пользователю доступно:

- перемещение модели вдоль мировой оси X (влево-право) клавишами **A** и **D** соответственно;
- перемещение модели вдоль мировой оси Z (вперед-назад) клавишами **W** и **S** соответственно;
- перемещение модели вдоль мировой оси Y (вверх-вниз) клавишами **Space** и **Shift** соответственно.
- поворот модели вокруг собственной оси X по часовой стрелке и против часовой стрелки клавишами **R** и **F** соответственно;
- поворот модели вокруг собственной оси Y по часовой стрелке и против часовой стрелки клавишами **T** и **G** соответственно;
- поворот модели вокруг собственной оси Z по часовой стрелке и против часовой стрелки клавишами **Y** и **H** соответственно.

1.4.4 Примеры работы

На рисунке 1.9 представлен графический интерфейс разработанного программного обеспечения.



Рисунок 1.9 – Графический интерфейс разработанного программного обеспечения

На рисунке 1.10 представлен пример работы разработанного программного обеспечения.



Рисунок 1.10 – Пример работы

ЗАКЛЮЧЕНИЕ

В ходе выполнения преддипломной практики было разработано программное обеспечение, демонстрирующее практическую осуществимость спроектированного в ходе выполнения выпускной квалификационной работы метода наложение теней в дополненной реальности на основе информации о глубине точек кадра.

Были описаны этапы метода, ограничения, основные модули разработанного программного обеспечения. Было написано руководство пользователя и приведены примеры работы программного обеспечения.

Таким образом, цель работы была достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Osti F., Santi G. M., Caligiana G.* Real time shadow mapping for augmented reality photorealistic rendering // Applied Sciences. — 2019. — Т. 9, № 11. — С. 2225.
2. Unity, платформа для разработки в реальном времени [Электронный ресурс]. — Режим доступа: <https://unity.com> (Дата обращения 14.04.2023).
3. Документация по C# [Электронный ресурс]. — Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения 20.05.2023).
4. Создание прототипа компьютерного бесконтактного компьютерного интерфейса в Unity 3D [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/sozdanie-prototipa-kompyuternogo-beskontaktnogo-kompyuternogo-interfeysa-v-unity-3d/viewer> (дата обращения 21.05.2023).