

## РЕФЕРАТ

Объем РПЗ составляет 53 страниц, содержит 12 иллюстрации, 3 таблицы, и 32 использованных источников.

*Ключевые слова:* дополненная реальность, компьютерное зрение, компьютерная графика, наложение теней, RGBD-изображение, восстановление параметров освещения, глубина точек кадра, карта глубины кадра, сферическая панорама.

# СОДЕРЖАНИЕ

<b>РЕФЕРАТ</b>	<b>5</b>
<b>ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ</b>	<b>8</b>
<b>ВВЕДЕНИЕ</b>	<b>9</b>
<b>1 Аналитический раздел</b>	<b>11</b>
1.1 Анализ предметной области . . . . .	11
1.2 Методы наложения теней в дополненной реальности . . . . .	15
1.2.1 Метод на основе анализа контуров теней . . . . .	15
1.2.2 Метод на основе построения теневых объемов . . . . .	17
1.2.3 Метод с использованием сверточных нейронных сетей и трассировки теневых лучей . . . . .	23
1.3 Сравнение методов . . . . .	27
1.4 Формализация задачи . . . . .	28
<b>2 Конструкторский раздел</b>	<b>30</b>
2.1 Разработанный метод . . . . .	30
2.1.1 Особенности метода . . . . .	30
2.1.2 Обработка изображения . . . . .	31
2.1.3 Поиск положения источников света . . . . .	33
2.1.4 Построение геометрии окружения . . . . .	35
2.1.5 Отрисовка теней . . . . .	37
2.2 Структуры данных . . . . .	38
2.3 Взаимодействие отдельных частей системы . . . . .	38
<b>3 Технологический раздел</b>	<b>41</b>
3.1 Выбор языка и средств программной реализации метода . . . . .	41
3.2 Ограничения метода . . . . .	42
3.3 Формат входных и выходных данных . . . . .	42
3.4 Описание сценариев Unity . . . . .	43
3.5 Демонстрация метода . . . . .	44
3.6 Взаимодействие пользователя с программным обеспечением . . . . .	45
3.6.1 Управление виртуальной камерой . . . . .	46

3.6.2 Управление моделями на сцене . . . . .	47
<b>4 Исследовательский раздел</b>	<b>49</b>
<b>ЗАКЛЮЧЕНИЕ</b>	<b>50</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>53</b>

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

1. ДР – дополненная реальность.
2. ИС – источник света.
3. RGBD (red, green, blue, depth или красный, зеленый, голубой, глубина)  
– тип изображения, в котором каждый пиксель хранит информацию о  
цвете и глубине.

## ВВЕДЕНИЕ

ДР является средством воздействия информационных технологий на окружающую среду [1]. Она позволяет совместить искусственно сгенерированное изображение с изображением реального мира с помощью различных датчиков, получающих информацию об окружающем мире, и специального ПО. Технология ДР предоставляет возможность накладывать элементы виртуальной информации поверх изображений объектов физического мира в реальном времени. Она используется во множестве сфер деятельности человека: медицина, построение анатомических моделей, образование, туризм и других [2].

Важными условиями реалистичного восприятия виртуальных объектов, вписанных в изображение реального мира, является моделирование свойств текстур и оптических свойств поверхности (отражение, пропускание, преломление света) и способности виртуальных объектов отбрасывать тени в соответствии с условиями освещения. Виртуальные объекты должны визуализироваться таким образом, чтобы поведение виртуальных теней, отбрасываемых виртуальными объектами, соответствовало поведению теней от реальных объектов и не вызывало у пользователя дискомфорта при наблюдении смешанного изображения [3].

Для моделирования свойств текстур и оптических свойств поверхности используются изображения высокого разрешения (текстуры), карты нормалей, карты рельефа, карты отражения и т. п. Однако даже при их использовании, если виртуальный объект не отбрасывает тень, это не решает проблему реалистичного восприятия виртуальных объектов, т. к. виртуальный объект будет выделяться на фоне окружения за счет отсутствия взаимодействия с окружающим светом, тем самым вызывать дискомфорт у пользователя. Таким образом, согласование систем освещения реального и виртуального миров является основополагающей [1].

В наложении теней участвуют 3 сущности: ИС, объект, отбрасывающий тень, и объект, на который отбрасывается тень. Соответственно, возможны комбинации:

- ИС может быть виртуальным или реальным;
- объект, отбрасывающий тень, может быть виртуальным или реальным;

- объект, на который отбрасывается тень, может быть виртуальным или реальным;

В данной работе рассматривается случай, при котором ИС является реальным, объект, отбрасывающий тень, является виртуальным и объект, на который отбрасывается тень, является реальным.

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. Поэтому в таких случаях требуется вычислить положения ИС только в начале сессии или при необходимости, когда схема освещения коренным образом изменилась. Такой подход позволит сократить использование вычислительных ресурсов, тратя их только на обработку компьютерной графики и построение геометрии окружения.

**Цель работы** – разработать метод наложения теней в дополненной реальности.

Для достижения поставленной цели нужно решить следующие задачи:

- провести анализ предметной области наложения теней;
- провести обзор существующих методов наложения теней в ДР на основе информации о глубине точек кадра и привести результаты сравнительного анализа;
- разработать и описать собственный метод наложения теней в ДР на основе информации о глубине точек кадра, который будет вычислять положения ИС только в начале сессии или при необходимости;
- разработать программное обеспечение, реализующее описанный метод, и выполнить его тестирование;
- провести исследование результатов разработанного метода при проецировании теней от виртуального объекта на различные поверхности;
- выполнить сравнение результатов работы реализованного метода с результатами, полученными с помощью существующих аналогов.

# 1 Аналитический раздел

В данном разделе изучается предметная область, описываются основные определения ДР, компьютерного зрения и наложения теней, проводится обзор существующих методов получения данных о глубине окружения и способов построения теней на основе информации о глубине точек кадра. Также проводится сравнение методов между собой и формализация задачи в виде IDEF0-диаграммы.

## 1.1 Анализ предметной области

### Дополненная реальность и компьютерное зрение

*Дополненная реальность* – технология интеграции цифровой информации в виде изображений компьютерной графики, текста, видео, и другой информации и изображений объектов действительного (физического) мира в режиме реального времени [2].

*Компьютерное зрение* – теория и технология создания машин, которые могут производить обнаружение, слежение и классификацию объектов [4].

Технология ДР полагается на теорию компьютерного зрения, а точнее на анализ и обработку изображений. В области ДР алгоритмы компьютерного зрения используются для поиска в видеопотоке специальных маркеров или точек интереса. Алгоритмы компьютерного зрения позволяют выделять ключевые особенности на изображении (границы области, углы), производить поиск объектов в реальном времени и многое другое. После нахождения маркеров или точек интереса в видеопотоке и вычислении их местоположения, появляется возможность построения матрицы проекции и позиционирования виртуальных моделей. С помощью матриц проекции и позиционирования можно наложить виртуальный объект на видеопоток таким образом, что будет достигнут эффект присутствия [2].

### Данные об окружении

При наложении теней в качестве исходных данных используются RGBD-изображения [5] [6].

*RGBD-изображение* – изображение, в котором каждый его пиксель помимо информации о цвете хранит информацию и о глубине [8]. Существует

несколько способов получения информации о глубине точек кадра:

- структурированный свет;
- стереокамера;
- время полета (также Time-of-Flight);
- лидар (также LIDAR или LiDAR);

## **Структурированный свет**

Данный способ полагается на проектор света, который проецирует шаблоны на реальные объекты. В качестве шаблонов могут быть точка, линия, полоса определенной толщины и регулярная сетка. Однако первые три шаблона требуют поворота проектора хотя бы по одной оси для сканирования всей области, поэтому обычно используют регулярную сетку, для которой этого не требуется [9].

Для проектора используется инфракрасный свет для минимизации помех. На основе искажения шаблона можно восстановить информацию о глубине с помощью паралакса, зная положение проектора в системе координат камеры. Большинство датчиков структурированного света не работают под прямыми лучами солнца, поскольку они полагаются на проекцию света в сцене. Поэтому они обычно подходят для использования в помещениях [9].

## **Стереокамера**

Данный способ схож со структурированным светом, но отличается использованием еще одной камеры вместо проектора. Зная положения проекций одной и той же точки на изображения камер, и расстояние между камерами, можно вычислить глубину этой точки на основе паралакса. Стереокамеры работают как в помещении, так и на открытом воздухе. Проблемы же у данного способа возникают при анализе глубины точек гладкой окрашенной стены, так как изображения на обеих камерах эквивалентны. [8].

## **Время полета**

Данный способ основан на оценке расстояния от датчика до наблюдаемого объекта путем измерения времени, необходимого для приема датчиком



излучаемого света, отраженного от объекта. Поэтому датчики используют модуляцию излучаемого светового сигнала и сравнивают изменение фазы выпущенного сигнала и отраженного. Концепция практически не отличается от ультразвуковых и радарных датчиков, но здесь в качестве излучаемого сигнала используется свет. При сильном солнечном свете датчик может давать сбои, поэтому данный способ также чаще всего используют в помещениях [8].

## **Лидар**

Данный способ использует идею измерения времени движения света, в течение которого излучаемый свет принимается датчиком, но они полагаются на один или несколько лазерных лучей (концентрированный свет) для измерения глубины точек в сцене. Лидарный датчик полагается на сфокусированные лазерные лучи, которые позволяют проводить измерения расстояния до нескольких километров. Точность измерений обычно не зависит от расстояния, хотя может зависеть от погодных условий: при неблагоприятных условиях, например, густой туман и бурный снегопад, она может падать. Лидарные датчики излучают свет, поэтому они работают в сложных условиях освещения, например, при недостатке освещения. Они подходят для использования в помещениях и на открытом воздухе [8].

## **Построение тени**

Тень состоит из двух частей: полутени и полной тени. Полная тень – центральная, темная, резко отчерченная часть, а полутень (или мягкая тень) – окружающая ее более светлая часть. Полная тень образуется при полном отсутствии света, а полутень освещается частью рассеянного света ИС [10].

Существует несколько способов построения теней:

- использование теневой карты;
- построение теневого объема;
- обратная трассировка лучей.

## **Теневая карта**

Теневая карта – объект, хранящий информацию о затененных пикселях буфера кадра. Основная идея использования теневой карты – построение

карты глубины с точки зрения ИС и сравнение с картой глубины виртуальной камеры: те пиксели, которые видны с точки зрения камеры, но не видны с точки зрения ИС, являются затененными. Этот процесс называется *теневым тестом*. Такой подход не требователен к вычислительным ресурсам, но требователен к памяти, поскольку теневая карта обладает тем же разрешением, что и конечное изображение, и таких карт может быть несколько в зависимости от количества ИС [11].

## **Теневой объем**

Теневой объем – это область пространства, доступ света в которую закрывает объект. Боковые границы теневого объема образуются четырехугольниками, расположенными от границ объекта, направленных на ИС, до некоторой пересекающейся с теневым объемом плоскостью. Границы ближнего к источнику основания четырехугольника образуются так называемыми границами силуэта. Эти границы образуются сторонами треугольников, находящимися на границе затененной и освещенной областей объекта или рядом с ней. В общем случае граница силуэта пролегает по границе между треугольниками, один из которых обращен к источнику света, а другой – в обратную сторону. Чтобы получить теневой объем, нужно найти все границы силуэта и нарисовать четырехугольники для каждой из них. Границы же дальнего основания образуются границами пересечения теневого объема с ближайшей к ИС плоскостью [12].

Освещенность точки определяется следующим образом: от точки местоположения камеры до нее проводится луч, который связан со счетчиком прохождения граней теневого объема. При входе в грань теневого объема счетчик увеличивается, при выходе – уменьшается. В итоге, если счетчик больше нуля, то точка считается затененной, иначе – она освещена.

Такой подход может давать ложные результаты, если камера находится внутри теневого объема. Эта проблема устраняется решением обратной задачи – трассировкой лучей из бесконечности к точке местоположения камеры [12].

## **Обратная трассировка лучей**

Обратная трассировка лучей – это метод оценки освещенности, при котором для каждого пикселя в буфере кадра луч, выпущенный из пикселя,

сталкивается с объектами сцены, чтобы определить отраженный цвет объекта. Луч может быть прослежен дальше, после первого столкновения с некоторым объектом, для создания эффектов, подобных зеркалу или стеклу. Тени с трассировкой лучей оцениваются аналогично оценке отражённого объектом света, но вместо оценки столкновения лучей из буфера кадра со сценой проверка столкновения выполняется для лучей от сцены к источнику света. Если луч от текущего пикселя столкнется с каким-либо объектом до того, как достигнет источника света, текущий пиксель окажется в тени, и, таким образом, его цвет будет темнее по сравнению с основным цветом его материала [11].

## **1.2 Методы наложения теней в дополненной реальности**

Существует несколько методов наложения теней, использующие информацию о глубине точек кадра:

- метод на основе анализа контуров теней ИС;
- метод на основе построения теневых объемов;
- метод с использованием сверточных нейронных сетей и трассировки теневых лучей.

### **1.2.1 Метод на основе анализа контуров теней**

В качестве исходных данных используются RGBD-изображения.

Из исходных данных вычисляется следующий набор данных: точки, соответствующие границам объектов, и точки, соответствующие границам теней от объектов. Предполагается, что имеется прямое соответствие между объектами и их тенями на RGBD-изображении. Количество найденных точек практически не сказывается на точности, но они должны быть расположены равномерно по контуру изображения объекта, и отклонение точек от границы контура не должно быть значительным. Количество точек на контурах объектов и их теней не должно быть меньше 3 [13].

Полученные наборы точек используются для нахождения положений ИС следующим образом.

1. Строится набор плоскостей параллельно поверхности земли.
2. Из точек на границах тени выпускают лучи через сопряженные точки на границах объекта. Эти лучи пересекают ранее построенные плоскости, тем самым итеративно получают точки пересечения лучей с плоскостями.
3. Проверяется характер распределения скоплений этих точек пересечения на каждой плоскости. Скопление точек, полученное данным методом, уплотняется при приближении виртуальной плоскости к ИС. На определенном шаге итераций скопление точек перестанет уплотняться и в дальнейшем будет расширяться. Максимальная плотность скопления свидетельствует о том, что следующая плоскость находится дальше от источника, чем текущая. А источник расположен между этой и следующей плоскостями.
4. После этого определяется центры скоплений этих точек. Зная высоту найденной виртуальной плоскости и центры скоплений, можно восстановить положения ИС. При этом абсолютная ошибка определения координаты ИС по высоте не превышает половины шага, с которым чередуются виртуальные плоскости [13].
5. Далее уточняются координаты ИС с помощью билинейной интерполяции между двумя виртуальными плоскостями с максимальной плотностью точек.

Также стоит упомянуть случай, когда не был найден ни один ИС. Это может происходить, например, если объекты и их тени были некорректно согласованы или видимый край объекта не является краем, создающим тень [13].

Поскольку ограничением данного метода является невозможность восстановить интенсивность и диаграмму излучения ИС, то тип найденного ИС устанавливается точечным, а диаграмма излучения – Ламбертовой [13].

## **Преимущества и недостатки**

Преимущества данного метода являются:

- работа в помещении;

- возможность распознать несколько ИС.

Недостатками данного метода являются:

- зависимость качества его работы от результатов распознавания контуров объектов и теней. Устройство распознавания может распознать только часть контуров, например, распознанная часть тени может не соответствовать обнаруженному контуру объекта.

### **1.2.2 Метод на основе построения теневых объемов**

Данный метод состоит из трех этапов.

1. Обнаружение теней на видео в реальном времени.
2. Построение теневого объема.
3. Отрисовка виртуального объекта и синтез изображения.

В качестве исходных данных используются RGBD-изображения.

В методе текущий, то есть  $k$ -й, кадр видео разделяется на две области: область проекции и новую область.

Предыдущий, то есть  $k - 1$ -й, кадр видео проецируется на  $k$ -й кадр с оцененной позиции камеры и с использованием информации о глубине точек кадра. Область, которая является результатом пересечения двух кадров, называется областью проекции. Новая область – область  $k$ -го кадра, не попавшая в область проекции, то есть разность между областью проекции и  $k$ -м кадром. Далее фильтруются данные глубины кадра от шумов и оценивается положение камеры. Затем происходит обнаружение граней новых теней в новой области, и грани новых теней добавляются к старым.

На основе обнаруженных граней теней строится теновый объем и уточняется с помощью адаптивной стратегии выборки [14] для достижения плавных эффектов отбрасывания теней.

В итоге происходит синтез и отображение конечного изображения с виртуальным объектом.

## Обнаружение теней на видео в реальном времени

Поиск теней происходит в пространстве изображения.

Большая часть теней распределена по всей области проекции. В этой области грани тени кадра  $k$  могут быть непосредственно спроецированы из кадра  $k - 1$ . Чтобы устранить ошибки проецирования, вызванные зашумленными параметрами камеры и данными о глубине, происходит оптимизация первоначально спроецированных граней тени в области проецирования. Грани тени в новой области обнаруживаются и объединяются с гранями теней из области проекции путем обеспечения плавности перехода между гранями теней из области проекции и гранями теней из новой области [15].

### Область проекции

Для оптимизации контуров теней используется алгоритм Кэнни. К полученным результатам применяют фильтрацию среднего сдвига чтобы сгладить контуры теней и избежать влияния мягких теней и шума изображения. Затем пиксель на грани тени, для которого отношение интенсивности двух сторон его направления градиента превышает заданный порог (0.65 от темной стороны к светлой стороне), идентифицируется как находящийся на краю тени. Далее происходит объединение пикселей изначального контура теней с пикселями контуров теней, полученных после оптимизации, путем поиска пикселей граней теней в небольшой окрестности (3 на 3 пикселя) каждого пикселя изначального контура теней. Если в окрестности появляется пиксель контура тени после оптимизации, то он заменяет пиксель изначального контура тени [15].

### Новая область

Изображение представляет собой попиксельное произведение освещенности и коэффициента отражения:

$$I(x) = R(x)L(x), \quad (1.1)$$

где  $I(x)$  – наблюдаемый цвет RGB в пикселе  $x$ , а  $L(x)$  и  $R(x)$  – освещенность и коэффициент отражения (альбедо). Для пикселей  $p$  и  $q$ , которые находятся в теневой и не теневой областях соответственно и на небольшом

расстоянии от пикселя  $x$  (3 на 3 пикселя), интенсивности изображения следующие:

$$\begin{aligned} I(p) &= R(p)(L_{sun} \cos(\theta) + L_{sky}) \\ I(q) &= R(q)L_{sky}, \end{aligned} \quad (1.2)$$

где  $L_{sun}$  – интенсивность солнечного света,  $L_{sky}$  – интенсивность рассеянного света, то есть неба.

Так как пиксели  $p$  и  $q$  примыкают к границе тени, то предполагается, что они имеют одинаковую отражательную способность, то есть  $R(p) \approx R(q)$ , из чего следует, что:

$$\frac{I(p)}{I(q)} \approx \frac{L_{sun} \cos(\theta) + L_{sky}}{L_{sky}} = 1 + \cos(\theta) \frac{L_{sun}}{L_{sky}}, \quad (1.3)$$

где  $\theta$  – угол падения между направлением солнечного света и нормалью к поверхности в пикселе  $p$ .

Для пикселей с известными нормальными направлениями или плоских пикселей, то есть пикселей без информации о глубине, вычисляется отношение  $r = \frac{L_{sun}}{L_{sky}}$ . Для каждого пикселя  $x$  на гранях тени в области проекции вычисляется  $r(x)$ . Затем вычисляется усредненное соотношение по всем пикселям на гранях тени в области проекции и обозначается как  $r_s$ .

В новой области грани тени вычисляются теми же методами, что и в области проекции, но каждая точка грани подвергается анализу: используется пространственное расстояние между пикселем  $y$  и краем тени из области проекции в качестве веса  $w(y)$ , который управляет балансом двух объектов, который определяется как нормализованное среднее расстояние между  $y$  и  $t$  ближайшими пикселями на гранях тени из области проекции. На основе чего вычисляется вероятность пребывания точки  $y$  в тени [15]:

$$\begin{cases} p(y) = (1 - w(y))(1 - S(y)) + w(y)M(y), \\ S(y) = \frac{r(y)}{(r_s + r(y))}, \\ M(y) = \frac{|T(p) - T(q)|}{\max(T(p), T(q))}, \\ T(y) = \frac{H(y)}{V(y) + 0.01}, \end{cases} \quad (1.4)$$

где  $p(y)$  – вероятность пребывания точки  $y$  в тени,  $S(y)$  – отношение сходства  $r(y)$  и  $r_s$ ,  $T(y)$  – отношение оттенка  $H(y)$  и значения  $V(y)$  точки  $y$  в пространстве HSV [16],  $M(y)$  – разница соотношений  $T(p)$  и  $T(q)$ .

## Объединение граней тени

Перед объединением граней теней новой области и области проекции вершины граней области проекции сортируются по глубине. Затем порядок граничных пикселей используется для соединения вершин. Чтобы избежать чрезмерных соединений, используется согласованность в направлениях градиента для уточнения связанных ребер [15].

Далее выбираются множества соседних вершин  $N_j(x)$ , где  $j$  – порядковый номер вершины. Далее выбирается набор кандидатов  $C_j(x)$ , где  $C_j(x_i)$  – набор пикселей граней тени в новой области в окрестности  $N_j(x)$ . Для каждого пикселя  $x$  в окрестности  $N_j(x)$  вычисляется среднее направление градиента  $g_x$  по  $N_j(x)$ . Затем для каждого пикселя  $x_i$  в  $C_j(x_i)$  вычисляется направление градиента  $g_{x_i}$ . Угловая разница выражается так:

$$d(x_i) = \frac{(g_x - g_{x_i})^2}{g_{x_i}^2}. \quad (1.5)$$

Если  $d(x_i) < \alpha$  ( $\alpha$  – некоторое пороговое значение),  $x_i$  идентифицируется как край тени.

## Построение теневого объема

### Приближенное построение

Для построения приближенного теневого объема помимо граней тени используется также информация о направлении падения солнечного света, полученная на основе местоположения пользователя и времени суток при съемке. Для каждого пикселя на краю тени испускается луч вдоль направления солнечного света, длина луча на практике равна максимально возможной величине. Если закрывающий солнце объект виден не полностью, отбрасываемая тень обычно обрезается по крайней мере на одну границу изображения. Соответственно, теневой объем объекта будет естественным образом усечен трехмерными плоскими поверхностями границ изображения. Для полностью видимого объекта, отбрасывающего тень, используется информация о его геометрии для усечения объема тени. Благодаря такому подходу поверхность



теневого объема грубо аппроксимируется серией параллелограммов [15].

### Уточненное построение

Для более гладкого затенения трехмерных объектов точки тени заменяются на непрерывные кривые. Делается это с помощью кривых Безье. Из-за сложной конфигурации наружных теней прямая подгонка Безье приводит к большим ошибкам. Вместо этого края тени разделяются на кусочно-гладкие сегменты с использованием дискретной кривизны. Затем множества точек границ тени делятся на сегменты. Для каждого сегмента первая точка, а именно  $P_0$ , и последняя точка, а именно  $P_2$ , устанавливаются в качестве контрольных точек. Затем, используя координаты других точек на отрезке, оптимальная точка, а именно  $P_1$ , получается с помощью подгонки кривой Безье методом наименьших квадратов [15]:

$$B(t) = \sum_{k=0}^2 p_k B_{k,1}(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2. \quad (1.6)$$

При повторной выборке подогнанных кривых количество выбранных точек должно быть достаточным для создания плавных краев тени и как можно меньшим для поддержания эффективности. Для достижения этой цели используется адаптивная стратегия выборки. Для каждого двухмерного сегмента  $l$ , обладающего  $n$  пикселями, обозначается расстояние между первой и последней точкой сегмента  $P_{2D}^1$  и  $P_{2D}^n$  соответственно в виде  $d(P_{2D}^1, P_{2D}^n)$ , и их трехмерное расстояние  $d(P_{3D}^1, P_{3D}^n)$ . Их соотношение  $\frac{d(P_{2D}^1, P_{2D}^n)}{d(P_{3D}^1, P_{3D}^n)}$  представляет трехмерное расстояние, соответствующее единице пикселя на изображении. Чем меньше соотношение, тем больше точек следует отобрать. Далее, когда края тени находятся далеко от камеры, количество точек выборки следует уменьшить. В противном случае следует его увеличить. Наконец, трехмерное расстояние также определяет количество точек выборки [15].

Конечное количество точек выборки, которое обозначается как  $m$ , выражается как:

$$m = \max\left(\frac{d(P_{2D}^1, P_{2D}^n)}{d(P_{3D}^1, P_{3D}^n)} \cdot \frac{d(P_{3D}^1, P_{3D}^n)}{\frac{1}{n} \sum_{i=1}^n d_i}, \epsilon\right), \quad (1.7)$$

где  $d_i$  - глубина  $i$ -й точки и  $\epsilon$  - минимальное количество точек выборки, которое экспериментально установлено равным  $2n$  [15].

Схема метода изображена на рисунке 1.1.

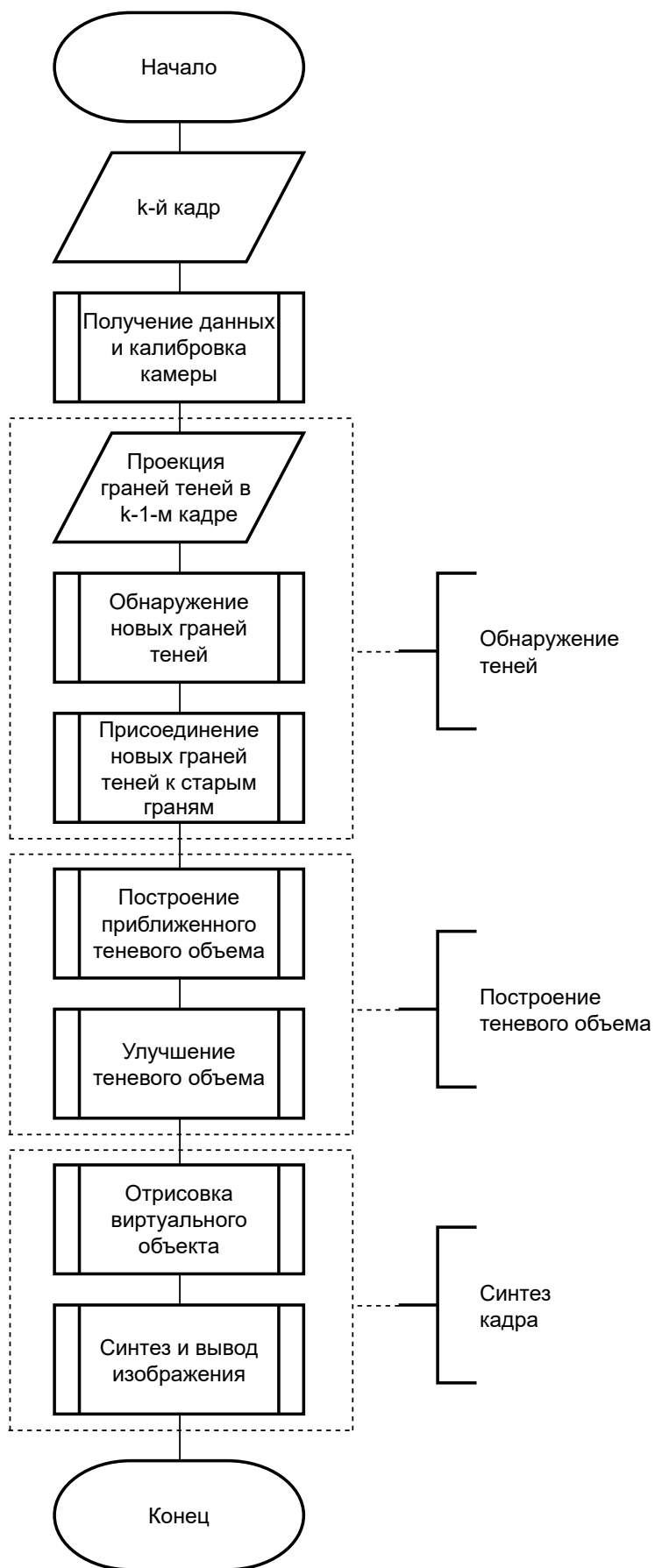


Рисунок 1.1 – Схема метода на основе построения теневого объема

## Преимущества и недостатки

Преимущества:

- моделирует систему освещения вне помещения;
- моделирует взаимодействие между тенями реальных и виртуальных объектов.

Недостатки:

- не позволяет моделировать освещение в помещениях;
- работает только со статическими тенями: динамические тени не распознаются, поскольку новые грани тени добавляются к граням предыдущего кадра путем обеспечения согласованности направления градиента края;
- плохо работает с мягкими тенями.

### 1.2.3 Метод с использованием сверточных нейронных сетей и трассировки теневых лучей

Суть метода – определить координаты ИС по теням, отбрасываемым объектами. Он основан на предположении, что для небольших ИС тень объекта является изображением центральной проекции этого объекта на поверхности «пола». Следовательно, зная координаты сопряженных точек границ теней и границ объектов, отбрасывающих эти тени, можно восстановить центральную проекцию и найти положение источника света [6].

Однако найти сопряженные точки – задача нетривиальная, особенно для сложных сцен, когда есть много теней от разных ИС, и когда тени не проецируются на плоскую поверхность. Метод основан на формировании пучков лучей, исходящих из точек на границе тени. В этом случае предполагается, что среди пучков лучей, испускаемых из тени к объекту, будет хотя бы один, идущий в направлении источника света. Эти лучи формируются из точек, полученных после определения контуров объектов и теней. В качестве контура объекта рассматривается не только его геометрический контур, но и световой контур, то есть граница света и тени на самом освещаемом объекте. Группа лучей, исходящих из разных точек тени на разные точки объекта,

может сформировать каустику, которая будет находиться вблизи источника света. Центр перетяжки этой каустики в пространстве сцены соответствует положению источника света. Поэтому основная задача метода – найти группу лучей, формирующих каустику [6].

В качестве исходных данных используется RGBD-изображение сцены, не требующее калибровки по реальным значениям яркости.

Данный метод состоит из двух этапов.

1. Обучение сверточной нейронной сети для определения границ объектов и теневых областей RGBD-изображений, полученных устройством ДР.
2. Использование алгоритмов машинного зрения для определения положения источников освещения в сцене.

Более подробно второй этап метода выглядит так.

1. Определяются все теневые области на изображении.
2. Идентифицируются объекты, отбрасывающие тени, и определяются границы объектов, включая световые границы в областях освещаемой и теневой частей объекта. Точки этих границ формируются и сохраняются.
3. Формируются облака точек вероятного пересечения лучей, исходящих из разных точек тени и объекта. Образуются пары несопряженных лучей, то есть лучи должны исходить из разных точек через разные точки одного объекта. Поскольку фактическое пересечение таких лучей невозможно, выполняется поиск точки на отрезке с минимальным расстоянием, соединяющим две эти прямые. Точки позади объекта или за пределами области определения сцены отбрасываются.
4. Точки, полученные в результате пересечения траекторий лучей, помечаются номером объекта, через который прошел луч. Эта маркировка позволяет сортировать сформированные лучи.
5. Происходит анализ областей скоплений точек, которые принимаются за положение источников света. Для каждой области координаты ИС усредняются, и средняя точка берется за точку положения ИС.

6. Для найденных точек проверяется правильность нахождения координат ИС. Для этого от источника света на границе тени испускаются лучи и оценивается отклонение координат соответствующих точек от ближайших точек границ объекта. Если отклонение находится в пределах допуска, то найденная точка принимается за центральную точку ИС, в противном случае источник света считается ложным и отклоняется. Кроме того, близкорасположенные ИС, найденные для различных объектов, объединяются в один ИС [6].

## **Определение контуров теней и объектов**

Изображения в оттенках серого и цветные изображения могут содержать значительный шум, заключающийся в случайных вариациях яркости или цветов точек изображения. Поэтому для определения контуров объектов и теней необходимо сперва устранить шум изображения, для чего используются различные методы фильтрации и алгоритмы компьютерного зрения. Для этого используются алгоритмы Кэнни [17] для обнаружения границ изображения, затем размытие по Гауссу [18] и операция наращивания [19] для устранения шума на границах изображения. Чтобы оставить только контуры границ, используется алгоритм скелетизации [20], который уменьшает бинарные объекты до ширины одной точки изображения.

После определения всех контуров объектов и теней на изображении необходимо найти соответствие между ними. В первую очередь строятся регионы интересов [21] области контуров, и если они соприкасаются, то есть имеют общие границы, то с большой вероятностью контур тени соответствует контуру объекта [6].

Кроме того, используется еще один метод сопоставления контуров, заключающийся в использовании функции, вычисляющей и сравнивающей по заданным регионам интересов «моменты» контуров изображений объектов и теней сцены. Далее в функцию сравнения контуров подаются полученные значения и возвращается метрика, показывающая сходство. Чем ниже результат на выходе функции (чем ближе она к нулю), тем больше соответствие и тем вероятнее, что сравниваемые контуры тени и объекта имеют одно происхождение, то есть тень была сформирована данным объектом [6].

## Формирование лучей

После того, как были определены все необходимые координаты на исходном изображении контуров объектов и их теней, начинается процесс формирования лучей. Они формируются с заданным шагом по контуру, например, исходя из соображения, что на контуре изображения и тени не должно быть больше 10 или 20 точек [6]. Исходя из этого на контурах выбираются точки с соответствующим шагом, через которые затем выпускаются лучи, и вычисляются точки, находящиеся на минимальном расстоянии между этими лучами, то есть точки перетяжки лучей. Вычисление точек перетяжки основывается на методе наименьших квадратов, что является стандартным подходом в регрессионном анализе для аппроксимации решения переопределенных систем путем минимизации суммы квадратов, полученных в результатах каждого отдельного уравнения [22].

Далее определяется максимальная плотность точек перетяжки. По найденным точкам в области наибольшей плотности вычисляются моменты и находится средняя точка.

Необходимо отметить, что если в процессе поиска координат источников света использовались два или более объектов сцены, то найденные облака точек, имеющие максимальную плотность и характеризующие источники света от разных групп объектов – теней, можно объединять в один общий источник света, имеющий конечный размер. Это объединение можно делать только в том случае, если облака точек были порождены различными объектами, поскольку один объект, формирующий разные тени, не может создать один источник [6].

## Преимущества и недостатки

Преимуществами данного метода являются:

- моделирует систему освещения внутри помещения;
- возможность распознать сложные тени и несколько ИС.

Недостатками данного метода являются:

- высокие требования к производительности системы [23].

### 1.3 Сравнение методов

В качестве критериев сравнения методов были предложены следующие:

- восстановление нескольких ИС;
- работа метода в помещении;
- работа метода вне помещения;
- динамическая смена окружения;
- пересчет положения ИС.

Динамическая смена окружения подразумевает, что метод способен восстановить систему освещения при смене окружения во время своей работы (переход в другое помещение и т. п.). Пересчет положения ИС подразумевает, что метод вычисляет положения ИС в каждом кадре во время сессии, даже если система освещения не меняется, что избыточно, как ранее было сказано.

В таблице 1.1 приведено сравнение методов.

Таблица 1.1 – Сравнение методов наложения теней в ДР

	Восстано- вление несколь- ких ИС	Работа метода в помеще- нии	Работа метода вне поме- щения	Динамич- еская смена окруже- ния	Пересчет положе- ния ИС
Метод на основе анализа контуров теней ИС	+	+	-	+	-
Метод на основе построения тене- вых объемов	-	-	+	+	-
Метод с исполь- зованием сверточ- ных нейронных се- тей и трассировки теневого лучей	+	+	-	+	-

Из результатов сравнения видно, что вышеописанные методы подходят для разных случаев наложения теней в дополненной реальности, но ни один из них не соответствует последнему критерию, из чего следует необходимость разработки собственного метода наложения теней в дополненной реальности, который ему соответствует, и для тех условий, когда схема осещения окружения в основном не изменяется.

## 1.4 Формализация задачи

На рисунке 1.2 представлена формализация задачи в виде IDEF0-диаграммы.



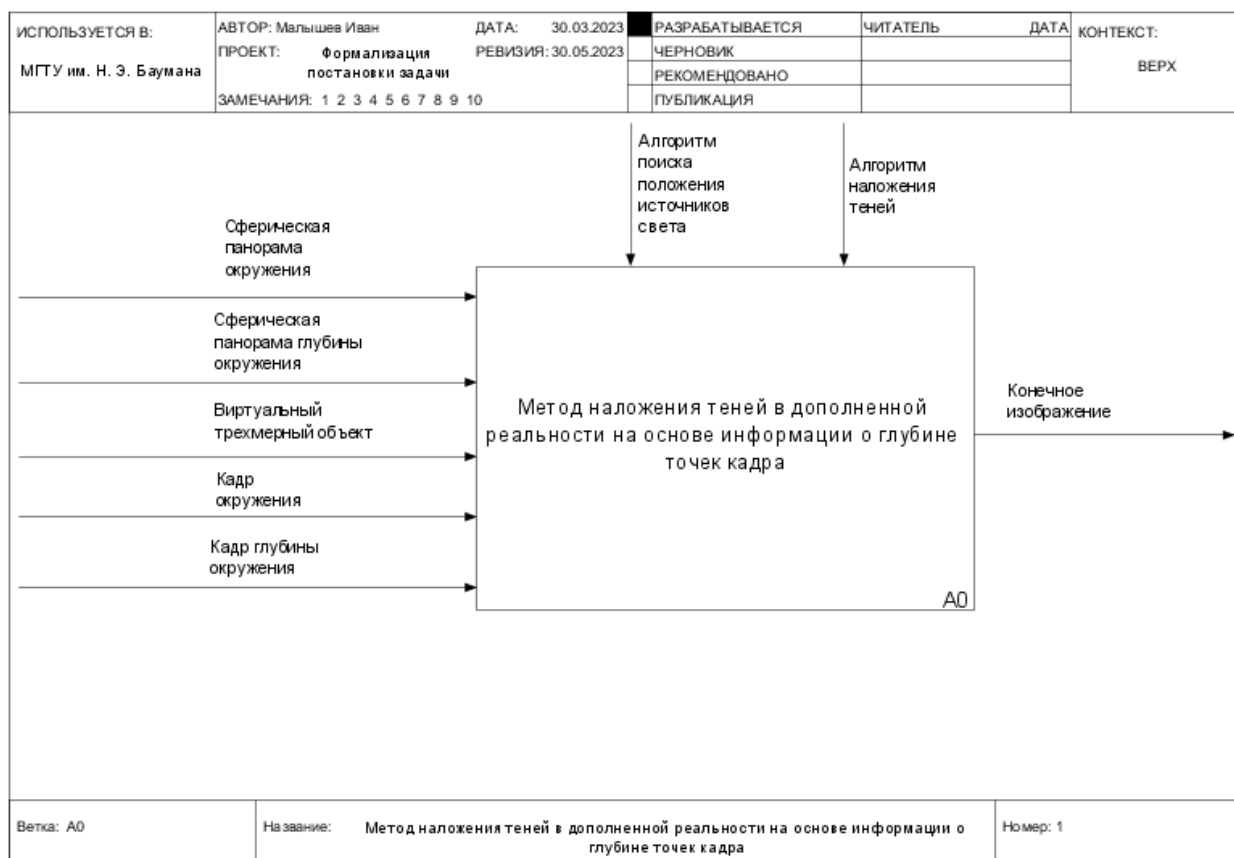


Рисунок 1.2 – Формализация задачи в виде IDEF0-диаграммы

## Вывод

В данном разделе были изучены предметная область, описаны основные определения ДР, компьютерного зрения и наложения теней, проведен обзор существующих методов получения данных о глубине окружения и способов построения теней на основе информации о глубине точек кадра. Также было проведено сравнение методов между собой и была формализованна задача в виде IDEF0-диаграммы.

## 2 Конструкторский раздел

В данном разделе описывается разработанный метод и его особенности; формулируются и описываются ключевые шаги метода в виде схем алгоритмов; описываются структуры данных, используемые в алгоритмах, и взаимодействие отдельных частей системы.

### 2.1 Разработанный метод

Разработанный метод основывается на методе, анализирующем гистограмму изображения окружения, которому не требуется вычислять положения ИС в каждом кадре, но в отличие от него учитывает глубину окружения [1].

Метод состоит из 4 основных этапов:

- обработка изображения;
- поиск положения источников света;
- построение геометрии окружения;
- отрисовка теней.

#### 2.1.1 Особенности метода

В этом методе в качестве исходных данных используется информация о глобальном освещении и глубине окружающей среды и любых источниках света, присутствующих вокруг пользователя, из сферической панорамы окружения и сферической панорамы глубины окружения. Исходные изображения должны обладать следующим свойством: они должны быть всенаправленными, т.е. для каждого направления пространства имеется пиксель, представляющий это направление.

Существует 2 подхода для получения сферической панорамы:

- использование всеракурсной камеры;
- создание из набора фотографий всего окружения и их компоновка с помощью стороннего ПО.

Также стоит уточнить, что первые 2 этапа метода (обработка изображения и поиск положения источников света) обрабатываются в начале сессии или по необходимости, например, при смене освещения окружения, а остальные 2 (построение геометрии окружения, отрисовка теней) обрабатываются в каждом кадре.

### 2.1.2 Обработка изображения

Полученная сферическая панорама конвертируется из цветного изображения в изображение в оттенках серого для более простого применения порогового значения к значениям цвета пикселей, поскольку в изображении в оттенках серого они варьируются только от 0 до 255. Стоит отметить, что не все цветные изображения после конвертирования в изображение в оттенках серого имеют пиксели, которые варьируются от 0 до 255. Например, слишком яркие или слишком тусклые изображения.

Далее проверяется следующее условие:

$$\frac{\max(\text{PixelValue})}{\text{average}(\text{PixelValue})} \geq 1.5, \quad (2.1)$$

где  $\text{PixelValue}$  – значение пикселя,  $\max(\text{PixelValue})$  – максимальное значение пикселя,  $\text{average}(\text{PixelValue})$  – среднее значение пикселя.

Если это условие верно, то это означает, что разница между максимальным значением и средним значением пикселей достаточно, чтобы утверждать, что существует видимая разница между окружающим светом и возможным точечным светом. В ином случае возможно ошибочное отождествление окружающего света с точечным светом, что приводит, во-первых, к слишком высокой плотности белых пикселей и, во-вторых, к неточному расположению света на этапе поиска положения источников света [1].

Далее происходит анализ гистограммы яркости изображения [24]. Вычисляется минимальный процент спектра яркости (начиная с 0), который покрывает не менее 98% пикселей, после чего оставшуюся часть спектра берут для обнаружения областей, где могут располагаться ИС. Этот шаг позволяет убрать ненужные области изображения посредством затемнения и рассматривать только те области изображения, которые соответствуют реальным ИС. Полученный результат обрабатывается медианным фильтром размытия в качестве метода шумоподавления, поскольку возможно наличие некоторых

областей, которые соответствуют отражениям объектов в окружающей среде [1].

Схема алгоритма изображена на рисунке 2.1

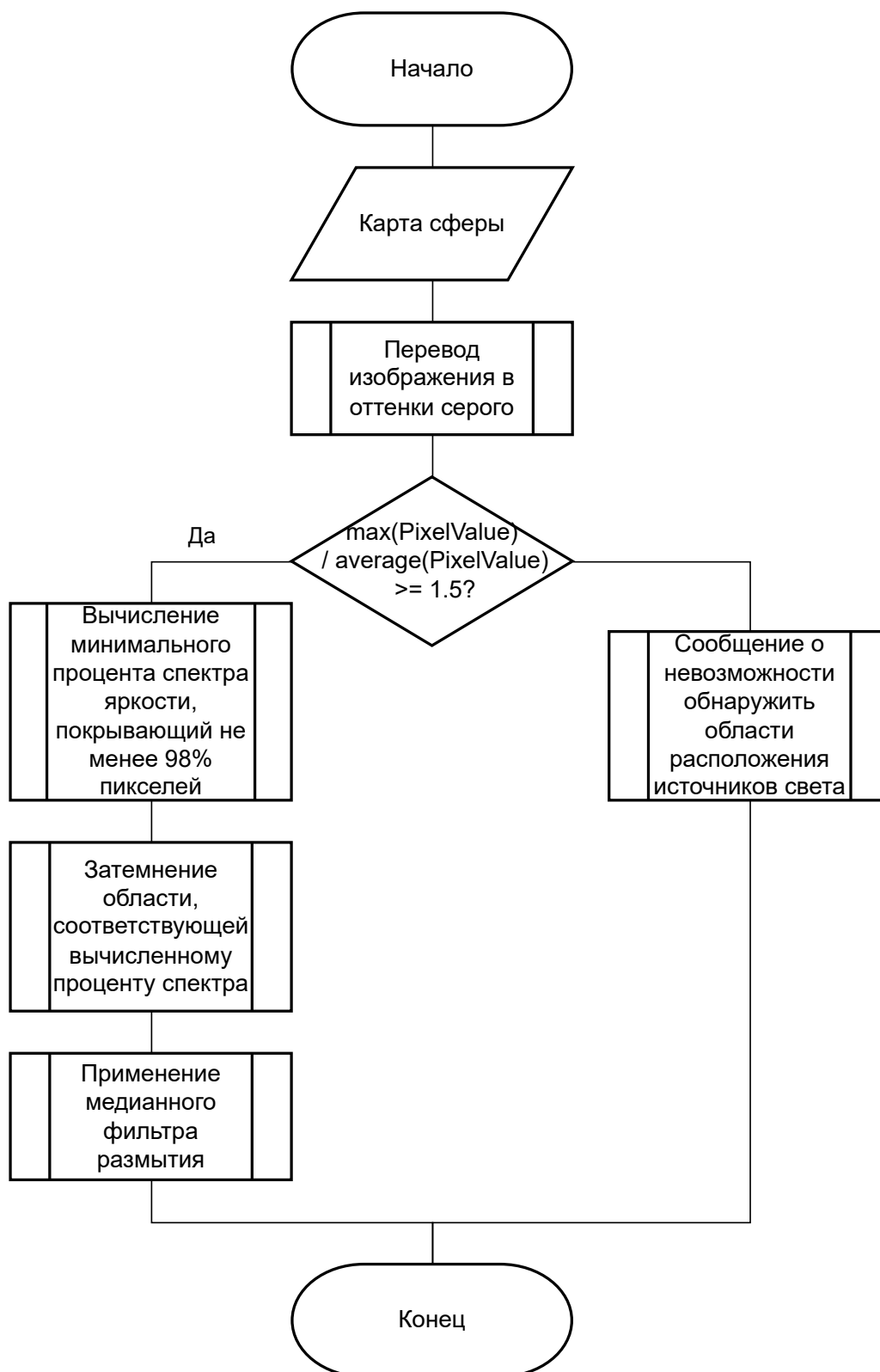


Рисунок 2.1 – Схема алгоритма обработки изображения

### 2.1.3 Поиск положения источников света

У полученного изображения вычисляются контуры ИС. Из полученных контуров определяются моменты контуров, из которых можно получить центроиды каждого ИС.

Моменты изображения представляют собой средневзвешенное значение интенсивности пикселей изображения, то есть это суммарная характеристика контура, рассчитанная интегрированием (суммированием) всех пикселей контура. Все что необходимо – это вычислить сумму интенсивностей всех пикселей и получить на выходе значение [6].

Из координат центроида на изображении окружения  $(x_{pixel}, y_{pixel})$  можно получить зенит  $\theta$  и азимут  $\phi$  ИС на сферической карте следующим образом. Координата  $x_{pixel}$  пропорциональна зениту, а координата  $y_{pixel}$  пропорциональна азимуту, из чего следуют соотношения [1]:

$$\begin{aligned} \frac{x_{pixel}}{\theta} &= \frac{\text{width}}{2\pi}, \\ \frac{\text{height} - y_{pixel}}{\phi} &= \frac{\text{height}}{\pi}, \end{aligned} \quad (2.2)$$

где width, height – ширина и высота исходного изображения окружения соответственно. Стоит отметить, что координата  $r$  у ИС равна средней глубине внутри прямоугольника, который описывает контур ИС.

Зная значения координат ИС в сферических координатах, можно их перевести в декартову систему координат:

$$\begin{aligned} x &= r \sin \theta \cos \phi, \\ y &= r \sin \theta \sin \phi, \\ z &= r \cos \theta. \end{aligned} \quad (2.3)$$

Таким образом, становится известно положение ИС в трехмерном пространстве.

Схема алгоритма изображена на рисунке 2.2

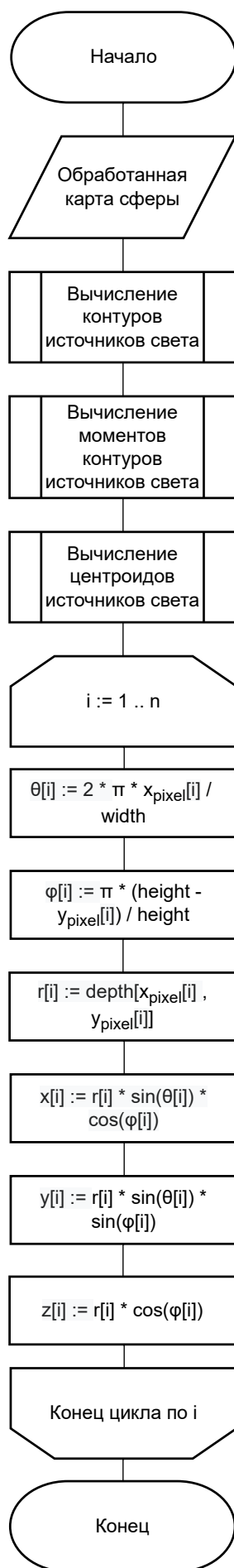


Рисунок 2.2 – Схема алгоритма поиска положения ИС

### 2.1.4 Построение геометрии окружения

Алгоритм построение сетки, отражающей геометрию окружения, состоит из двух этапов [25]:

- строится плоская прямоугольная сетка, количество вершин которой по длине и ширине определяется размерами входной карты глубины;
- каждая вершина, сопоставленная с точкой карты глубины, смещается на основе повторно спроецированного значения глубины.

Схема алгоритма изображена на рисунке 2.3

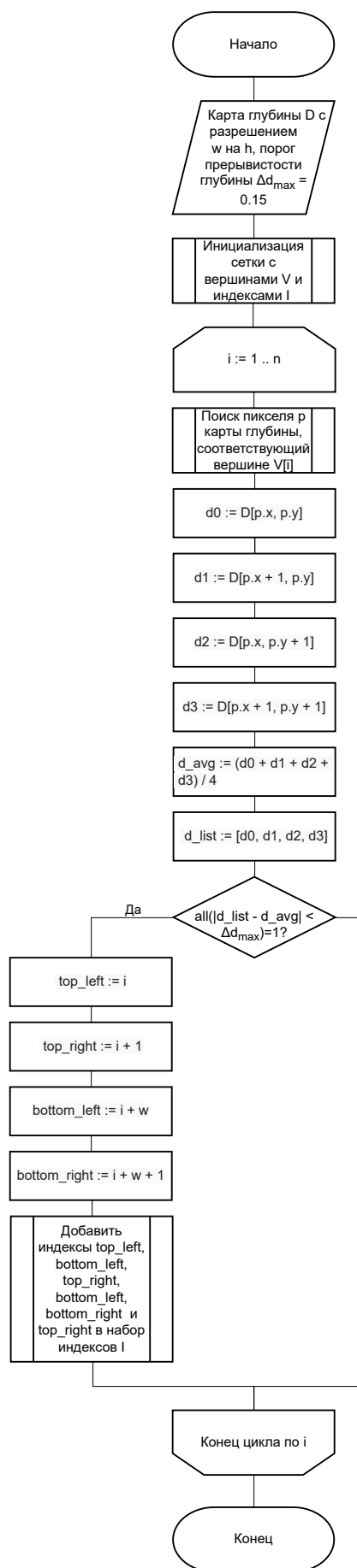


Рисунок 2.3 – Схема алгоритма построения геометрии окружения в реальном времени



Функция **all** проверяет булево выражения для каждого элемента некоторого списка. Если для всех элементов списка выражение истинно, то функция возвращает 1, иначе – 0. Значения глубины, используемые на данном этапе, нормализованы.

### 2.1.5 Отрисовка теней

Отрисовка теней происходит по алгоритмам машинной графики. Зная местоположение каждого ИС, расставляются их виртуальные аналоги. Как описано в предыдущем подразделе, поверх реальных объектов строится полигональная сетка, на которую падает тень от виртуального объекта, освещаемого расставленными ИС. В итоге происходит синтез и отображение виртуальной сцены с виртуальным объектом, отбрасывающим тень от «реальных» ИС.

Схема алгоритма изображена на рисунке 2.4



Рисунок 2.4 – Схема алгоритма отрисовки теней

## 2.2 Структуры данных

В таблице 2.1 приведены структуры данных, используемые в алгоритмах.

Таблица 2.1 – Структуры данных, используемые в алгоритмах

Данные	Представление
Точка трехмерного пространства	Координаты X, Y, Z
Вектор	Точка трехмерного пространства
Мировая система координат (МСК)	Точка трехмерного пространства с тремя ортонормированными векторами
Локальная система координат (ЛСК)	Точка трехмерного пространства относительно МСК с тремя ортонормированными векторами
Вершина	Точка трехмерного пространства
Полигон	Массив из трех индексов списка вершин
Полигональная сетка	Список полигонов
Карта сферы	Растровое изображение
Карта глубины	Растровое черно-белое изображение
Камера	Пространство обзора (ЛСК); ширина и высота виртуального экрана; отступы от центра ЛСК по оси Z ближней и дальней плоскости
Источник света	Точка в трехмерном пространстве; параметры источника света

## 2.3 Взаимодействие отдельных частей системы

На рисунке 2.5 представлена схема взаимодействия отдельных частей системы.

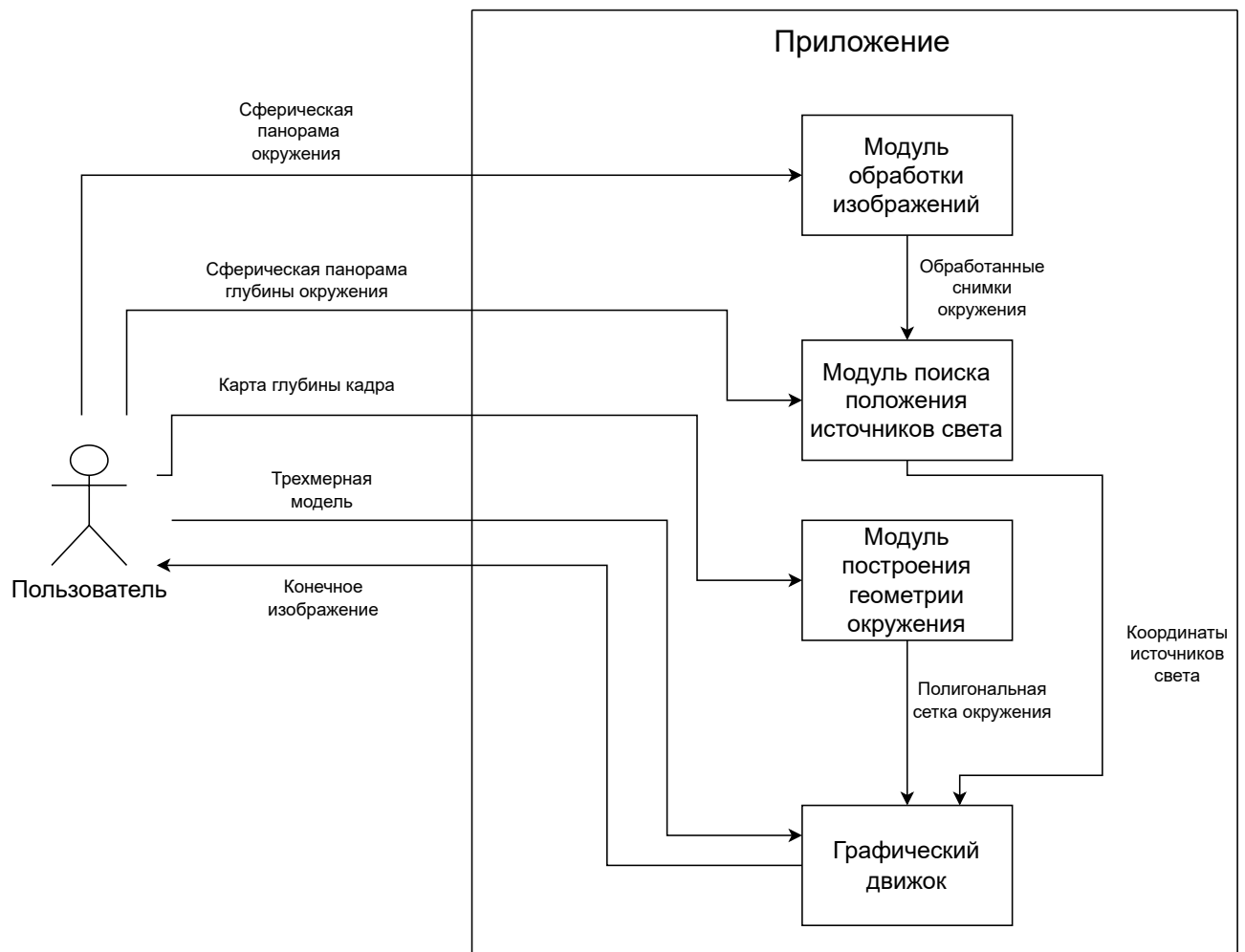


Рисунок 2.5 – Схема взаимодействия отдельных частей системы

На рисунке 2.6 представлена IDEF0-диаграмма описанного метода.

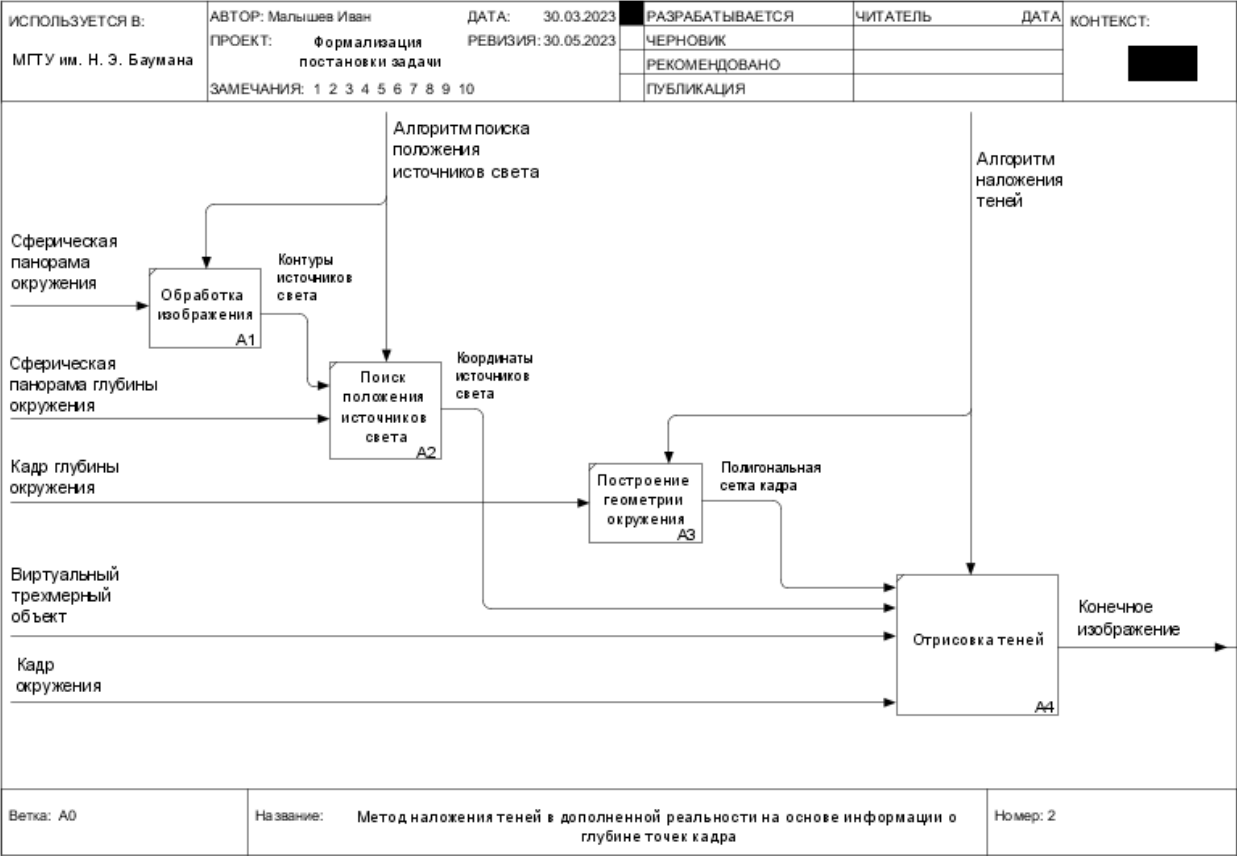


Рисунок 2.6 – IDEF0-диаграмма описанного метода

Вывод

В данном разделе был описан разработанный метод и особенности разработанного метода; были сформулированы и описаны ключевые шаги метода в виде схем алгоритмов. Также были описаны структуры данных, используемые в алгоритмах, и взаимодействие отдельных частей системы.

### 3 Технологический раздел

В данном разделе обосновывается выбор языка и средств программной реализации метода наложения теней в дополненной реальности; описывается разработанное программное обеспечение, реализующее описанный метод, и проводится его тестирование. Также описывается формат входных и выходных данных и взаимодействие пользователя с программным обеспечением.

#### 3.1 Выбор языка и средств программной реализации метода

Для обработки ИС, виртуальных теней, трехмерных моделей и прочих элементов машинной графики был выбран игровой движок Unity [26], поскольку он кроссплатформенный, обладает бесплатной версией, функционала которого достаточно для реализации метода, обширной документацией и множеством обучающего материала, что позволяет легко его освоить.

В качестве языка программирования были выбраны C# [27], поскольку скрипты для Unity используют этот язык. Также этот язык обладает удобным синтаксисом, управляемым кодом и сборщиком мусора, благодаря этому не нужно заботиться об утечках памяти, об указателях и о некоторых базовых структурах и алгоритмах – все это уже реализовано. Это позволит ускорить разработку и отладку кода.

В качестве среды разработки (IDE) была выбрана Visual Studio [28], обладающая интеллектуальными подсказками, инструментами анализа, отладки и тестирования кода, поставляющаяся вместе с языком C# и пакетным менеджером NuGet [29], который позволяет в случае чего дополнить недостающий функционал языка или игрового движка.

Для анализа и обработки изображений (анализ гистограммы, поиск контуров и т. д.) была выбрана библиотека OpenCV [30], а точнее обертка над ней OpenCVSharp [31], учитывая ранее выбранный язык. Для захвата данных об окружении была выбрана 3D-камера Kinect [32], поскольку к ней прилагается ПО, которое позволяет использовать ее вместе с Unity.

## 3.2 Ограничения метода

Так как в реализации метода используется Kinect, ограничения накладываются используемой 3D-камерой:

- минимальное расстояние от точки съемки окружения должно быть не менее 0.8 метров;
- максимальное расстояние от точки съемки окружения должно быть не более 4 метров.

Соблюдая эти условия, Kinect способен захватить информацию о глубине окружения. В ином случае значения глубины, полученные с Kinect, будут равны нулю.

Также стоит учитывать следующие ограничения метода:

- частота изменения системы освещения минимальна или равна нулю;
- отношение максимального значения яркости к среднему значению яркости сферической панорамы окружения должно быть не менее 1.5;
- любой ИС на сферической панораме окружения интерпретируется как точечный ИС с белым свечением;
- полигональная сетка окружения не учитывает свойства поверхности, такие как прозрачность и альбедо.

## 3.3 Формат входных и выходных данных

В таблице 3.1 приведено описание формата входных данных.

Таблица 3.1 – 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 с разрешением 640 на 480 пикселей
Кадр глубины окружения	Изображение в оттенках серого с разрешением 640 на 480 пикселей

Ограничения на входные данные:

- обе сферические панорамы должны быть с одинаковым разрешением;
- оба кадра окружения должны быть с одинаковым разрешением.

Выходными данными является цветное изображение с разрешением 640 на 480 пикселей.

### 3.4 Описание сценариев Unity

Сценарий Unity представляет собой файл с расширением **.cs**, в котором написан некоторый код на языке C#.

В разработанном программном обеспечении были реализованы следующие сценарии Unity:

- SavePano, SaveDepthPano предназначены для загрузки сферической панорамы окружения, загрузки сферической панорамы глубины окружения соответственно;
- EnvDataFields предназначен для хранения загруженных панорам и координат найденных ИС;

- LightPosCalc представляет собой алгоритм вычисления положения ИС;
- KinectManager и KinectWrapper предназначены для работы с Kinect, а также KinectManager отвечает за построение полигональной сетки окружения;
- SceneManager отвечает за расстановку ИС по найденным координатам, а также за выход в меню;
- ExamplesManager и ExampleManager отвечают за управление моделями на сцене;
- CameraMove и CameraRotate предназначены для перемещения и поворота виртуальной камеры соответственно.

### 3.5 Демонстрация метода

На рисунке 3.1 представлено сравнение поведения теней от реального и виртуального объекта.



Рисунок 3.1 – Сравнение поведения теней от реального и виртуального объекта

Как видно из результата сравнения, направления теней от виртуального объекта и реального совпадают, различия лишь заключаются в отсутствии мягких теней у виртуального объекта.



### 3.6 Взаимодействие пользователя с программным обеспечением

На рисунке 3.2 представлен начальный экран реализованного ПО.



Рисунок 3.2 – Начальный экран реализованного ПО

Для загрузки данных об окружении пользователю необходимо загрузить сферическую панораму окружения или директорию с набором изображений, из которых можно составить сферическую панораму, и сферическую панораму глубины окружения или директорию с набором изображений, из которых можно составить сферическую панораму глубины, для чего на начальном экране присутствуют соответствующие кнопки.

Для демонстрации метода присутствует кнопка «Демонстрация метода», которая загружает сцену, где демонстрируется восстановленная система освещения и взаимодействие виртуального объекта с реальными. Эта кнопка не будет действовать до тех пор, пока не будут загружены данные об окружении и данные о глубине окружения.

На рисунке 3.3 представлен пример работы реализованного метода.



Рисунок 3.3 – Пример работы реализованного метода

В верхнем правом углу отображается статус работы Kinect. Если при взаимодействии с Kinect возникла ошибка, то в этом поле отображается текст ошибки.

Для возврата в меню нужно нажать клавишу **Escape**.

### 3.6.1 Управление виртуальной камерой

Т. к. у Kinect нет GPS-модуля и гироскопа, то синхронизировать положение виртуальной камеры с положением Kinect в помещении требуется пользователю. Для этого в верхнем левом углу отображаются данные о позиции и повороте виртуальной камеры. Значения координат отображаются в сантиметрах, углы поворота – в градусах. Чтобы изменить координаты виртуальной камеры, используются следующие клавиши:

- W – увеличивает значение координаты Z;
- A – увеличивает значение координаты X;
- S – уменьшает значение координаты Z;
- D – уменьшает значение координаты X;
- Space – увеличивает значение координаты Y;
- Shift – уменьшает значение координаты Y.

Для изменения поворота виртуальной камеры используются следующие клавиши:

- **R** и **F** для поворота вверх и вниз соответственно;
- **T** и **G** для поворота влево и вправо соответственно.

Нулевая точка соответствует точке, в которой происходила съемка панорам окружения. Нулевые значения поворота камеры соответствуют направлению в центре панорамы.

### 3.6.2 Управление моделями на сцене

В разработанном программном обеспечении существует система управления трехмерными моделями. После загрузки панорам и вычисления положения источников света загружается сцена, где пользователю доступно создание, удаление и управление положением и поворотом заранее predetermined набора моделей, представленные на рисунке 3.4.



Рисунок 3.4 – 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.

#### Выбор модели на сцене

Выбор модели, имеющейся на сцене, осуществляется с помощью ЛКМ: нужно навести курсор на некоторую модель на сцене и нажать ЛКМ.

Чтобы сбросить выбранную модель, достаточно нажать ПКМ в любой части экрана.

## Удаление моделей

Удаление моделей происходит с помощью клавиши **Delete**: нужно выбрать модель на сцене и после нажать клавишу **Delete**.

## Управление моделями

Для управления выбранной моделью на сцене 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.

## Вывод

В данном разделе был обоснован выбор языка и средств программной реализации метода наложения теней в дополненной реальности; описаны ограничения метода и формат входных и выходных данных. Также было продемонстрировано разработанное программное обеспечение, реализующее описанный метод, проведена проверка его работоспособности и описано взаимодействие пользователя с программным обеспечением.

## 4 Исследовательский раздел

## ЗАКЛЮЧЕНИЕ

## 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. Технологии дополненной реальности [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/tehnologii-dopolnennoy-realnosti/viewer> (Дата обращения 30.09.2022).
3. Эффективный метод восстановления освещения в системах смешанной реальности с использованием HDR-изображения трехмерной сцены. Т. 29 / Н. Богданов [и др.]. — 2019. — С. 26—31.
4. Компьютерное зрение [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/kompyuternoe-zrenie> (Дата обращения 5.10.2022).
5. *Debevec P.* Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. — 2008. — С. 1—10.
6. М. И. Сорокин, Д. Д. Жданов, И. В. Валиев, "Восстановление положения источников освещения сцены в системах смешанной реальности с использованием сверточных нейронных сетей и трассировки теневых лучей" // Препринты ИПМ им. М. В. Келдыша – 2021 – С. 18.
7. Использование HDR-технологии для фотографической фиксации процессуальных действий [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/ispolzovanie-hdr-tehnologii-dlya-fotograficheskoy-fiksatsii-protsessualnyh-deystviy/viewer> (Дата обращения 8.10.2022).
8. *Lopes A., Souza R., Pedrini H.* A Survey on RGB-D Datasets // arXiv preprint arXiv:2201.05761. — 2022.
9. A comparative survey on invisible structured light [Электронный ресурс]. — Режим доступа: [https://fofi.pagesperso-orange.fr/Downloads/Fofi\\_EI2004.pdf](https://fofi.pagesperso-orange.fr/Downloads/Fofi_EI2004.pdf) (Дата обращения 10.10.2022).
10. *Роджерс Д.* Алгоритмические основы машинной графики. — Рипол Классик, 1989.

11. Programming vertex, geometry, and pixel shaders / W. F. Engel [и др.]. — Cengage Delmar Learning, 2008.
12. Вольф Д. OpenGL 4. Язык шейдеров. Книга рецептов / пер. с англ. А. Н. Киселева // М.: ДМК Пресс – 2015 – С. 368.
13. А. Ф. Лемешев, Д. Д. Жданов, Б. Х. Барладян, "Метод восстановления параметров освещения в системах смешанной реальности", Препринты ИПМ им. М. В. Келдыша – 2021 – С. 17.
14. *Thomasian A.* Chapter 8 - Database parallelism, big data and analytics, deep learning / под ред. А. Thomasian. — Morgan Kaufmann, 2022. — С. 385—491.
15. Simulating shadow interactions for outdoor augmented reality with RGBD data / H. Wei [и др.] // IEEE Access. — 2019. — Т. 7. — С. 75292—75304.
16. Color image segmentation: advances and prospects / H.-D. Cheng [и др.]. — Elsevier, 2001.
17. Canny, J., A Computational Approach To Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence – 1986.
18. Shapiro, L. G. & Stockman, G. C. Computer Vision. // Prentice Hall – 2001.
19. Jean Serra, Image Analysis and Mathematical Morphology, 1982.
20. Abeysinghe, Sasakthi. Segmentation-free skeletonization of grayscale volumes for shape understanding/ Abeysinghe, Sasakthi; Baker, Matthew; Chiu, Wah; Ju, Tao // Conf. Shape Modeling and Applications – 2008.
21. Ron Brinkmann (1999). The Art and Science of Digital Compositing. Morgan Kaufmann. С. 184.
22. Линник Ю. В. Метод наименьших квадратов и основы математико-статистической теории обработки наблюдений. // 2-е изд. – М. – 1962.
23. О некоторых методах визуализации динамических 3D моделей [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/o-nekotoryh-metodah-vizualizatsii-dinamicheskikh-3d-modeley> (Дата обращения 24.11.2022).



24. Способы обработки гистограммы телевизионного изображения [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/sposoby-obrabotki-gistogrammy-teplovizionnogo-izobrazheniya> (Дата обращения 15.10.2022).
25. DepthLab: Real-time 3D interaction with depth maps for mobile augmented reality / R. Du [и др.] // Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology. — 2020. — С. 829—843.
26. Unity, платформа для разработки в реальном времени [Электронный ресурс]. — Режим доступа: <https://unity.com> (Дата обращения 14.04.2023).
27. Документация по C# [Электронный ресурс]. — Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения 14.04.2023).
28. Visual Studio [Электронный ресурс]. — Режим доступа: <https://visualstudio.microsoft.com/ru/> (дата обращения 14.04.2023).
29. NuGet [Электронный ресурс]. — Режим доступа: <https://www.nuget.org/> (дата обращения 14.04.2023).
30. OpenCV. — Режим доступа: <https://opencv.org/> (дата обращения 17.04.2023).
31. OpenCVSharp. — Режим доступа: [https://shimat.github.io/opencvsharp\\_docs/](https://shimat.github.io/opencvsharp_docs/) (дата обращения 17.04.2023).
32. Создание прототипа компьютерного бесконтактного компьютерного интерфейса в Unity 3D [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/sozдание-prototipa-kompyuternogo-beskontaktnogo-kompyuternogo-interfeysa-v-unity-3d/viewer> (дата обращения 17.04.2023).