

Санкт-Петербургский политехнический университет Петра Великого
Институт машиностроения, материалов и транспорта
Высшая школа автоматизации и робототехники

**ОТЧЁТ О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ
РАБОТЕ**

Система стереозрения на основе ортогонально ориентированных камер

Выполнил
студент гр.3331506/80401 <подпись> М. Д. Пантелейев

Руководитель
старший преподаватель <подпись> А. С. Габриель

Научный консультант <подпись> В. В. Варлашин

«___» _____ 202__ г.

Санкт-Петербург
2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Аналитический обзор систем стереозрения	6
1.1 Стереозрение	6
1.2 Модели сверхширокоугольной камеры	10
1.3 Обоснование выбора ПО	14
1.4 Выводы по разделу	16
2 Система стереозрения	17
2.1 Описание системы стереозрения	17
2.2 Виртуальное моделирование системы	19
2.3 Передача изображений с виртуальных камер для обработки . .	21
2.4 Алгоритм устранения искажений	23
2.5 Выводы по разделу	28
3 Экспериментальное исследование системы стереозрения	30
3.1 Оценка качества устранения искажений	30
3.2 Исследование точности оценки глубины	32
3.3 Выводы по разделу	34
ЗАКЛЮЧЕНИЕ	36
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	37
ПРИЛОЖЕНИЕ А	40

ВВЕДЕНИЕ

За последние годы был совершен существенный прогресс в доступности и точности сенсоров, позволяющих мобильным роботам осуществлять оценку окружающего пространства. Такие информационно-измерительные устройства как лидары, сонары и стереокамеры стали основной опорой для алгоритмов автономной навигации и локализации. Тем не менее в роботах по-прежнему присутствуют телевизионные системы, так как они дают наиболее легко воспринимаемую информацию для оператора в случаях, когда его вмешательство необходимо. Число и расположение видеокамер может быть различным. Бортовые телевизионные системы классифицируются по различным критериям[1]:

- по количеству камер (монохромные или многокамерные).
- по функциональному назначению:
 - а) одиночные камеры;
 - б) PTZ-камеры (англ. Pan-Tilt-Zoom);
 - в) стереокамеры;
 - г) системы кругового обзора;
 - д) комбинированные системы.

Для наиболее эффективного покрытия телевизионной системой максимального объёма окружающего пространства в ней могут использоваться камеры "рыбий глаз" (англ. fisheye-camera), позволяющие одним кадром покрыть угловое поле свыше 180° . Набор таких камер может составлять систему кругового обзора, позволяющую оператору видеть не только в любом направлении, но даже с видом от третьего лица [2].

В случае автономных мобильных роботов телевизионный круговой обзор нужен лишь по необходимости, но при этом может быть весьма дорогостоящими и занимать место в корпусе. При этом в случае наличия пересечений полей зрения камер, входящих в систему, возможно применение принципа стереозрения. Однако значительные радиальные искажения изоб-

ражения, вызванные особенностями используемых объективов, не позволяют использовать известные алгоритмы стереозрения без подготовки.

Реализация системы стереозрения, способной работать в таких конфигурациях, позволит дать существующим роботам новый способ получать информацию об окружении и проектировать будущих роботов с учётом этой возможности.

Целью работы является разработка и исследование точности системы стереозрения, основанной на ортогонально расположенных сверхширокоугольных камерах.

В ходе работы решаются следующие задачи:

- обзор современных систем стереозрения, использующих изображения с широкоугольных камер,
- обзор моделей, применяющихся в моделировании камер с объективами ”рыбий глаз”,
- обоснование выбора программного обеспечения, используемого для разработки и тестирования алгоритма,
- разработка алгоритма устранения искажений fisheye-изображения в области кадра,
- моделирование системы стереозрения,
- определение точности оценки глубины системой.

1 Аналитический обзор систем стереозрения

1.1 Стереозрение

Система стереозрения состоит из двух камер, наблюдающих сцену с разных точек, как изображено на рисунке 1.2 [3]. Фундаментальная основа принципа заключается в предположении, что каждой точке в пространстве соответствует уникальная пара пикселей на снимках с двух камер.

При этом к камерам предъявляются некоторые требования [4]:

- Камеры откалиброваны. Это значит, что известны внутренние (оптические) и внешние (расположение камер в пространстве) параметры камер.
- Ректификация. Подразумевает выравнивание изображения с обеих камер по строкам.
- Ламбертовость поверхностей. Означает независимость освещения наблюдаемых поверхностей от угла зрения.

Таким образом, соблюдение указанных выше требований позволяет использовать следующий геометрический принцип. При наличии двух камеры, как изображено на рисунке 1.2, где C — центр первой камеры, C' — центр второй камеры, точка пространства X проецируется в x на плоскость изображения левой камеры и в x' на плоскость изображения правой камеры. Прообразом точки x на изображении левой камеры является луч xX . Этот луч проецируется на плоскость второй камеры в прямую l' , называемую эпиполярной линией. Образ точки X на плоскости изображения второй камеры обязательно лежит на эпиполярной линии l' .

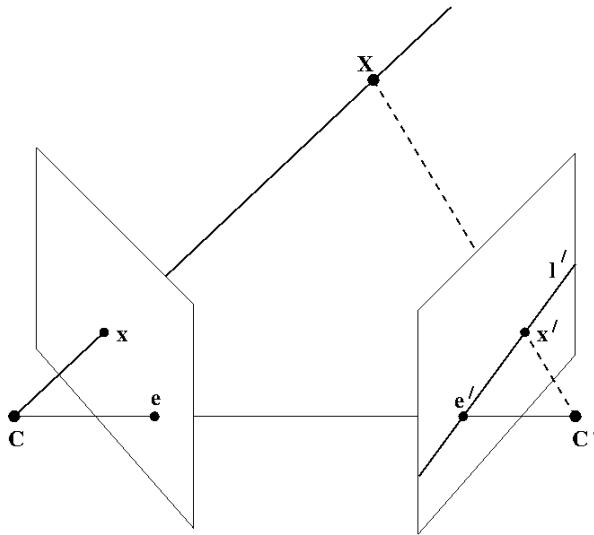


Рисунок 1.1 – Эпиполярная геометрия

В результате каждой точке x на изображении левой камеры соответствует эпиполярная линия l' на изображении правой камеры. При этом соответствие для x на изображении правой камеры может лежать только на соответствующей эпиполярной линии. Аналогично, каждой точке x' на правом изображении соответствует эпиполярная линия l на левом.

Далее с помощью точек x и x' возможно посчитать смещения каждого пикселя одного изображения относительно другого, что позволяет построить карту смещений. Очевидно, что смещения будут подсчитаны только для точек, видимых обеими камерами. Карта смещений же приводится далее либо к облаку точек, либо к карте глубины. Стереосистемы, реализующие этот принцип, называют пассивными. Они являются самыми простыми и часто используются, так как для их изготовления достаточно двух закреплённых камер. Однако пассивные системы опираются целиком на видимый свет, что ограничивает их применимость в условиях с малой освещённостью.

Одну из камер можно заменить источником света, освещаяющим одну или несколько точек поверхности световым лучом или специальным шаблоном освещения. Так как структура шаблона и направление его лучей заранее известны, камера может проводить оценку формы объектов в кадре поискажениям шаблона[5]. Структурная подсветка тоже является распространённым методом, реализованным во многих коммерческих сенсорах, благодаря совмещению высокой производительности и низкой цены.

Совмещение пассивного стереоизрения и структурированной подсветки позволяет улучшить качество стереосопоставления, особенно для поверхностей со слабо выраженной текстурой и при низкой освещённости. Однако инфракрасный узор подсветки весьма ограничен в дальности и не виден при сильном солнечном освещении, что затрудняет использование этих сенсоров на открытых пространствах. Системы с подобной технологией называют активными [6].

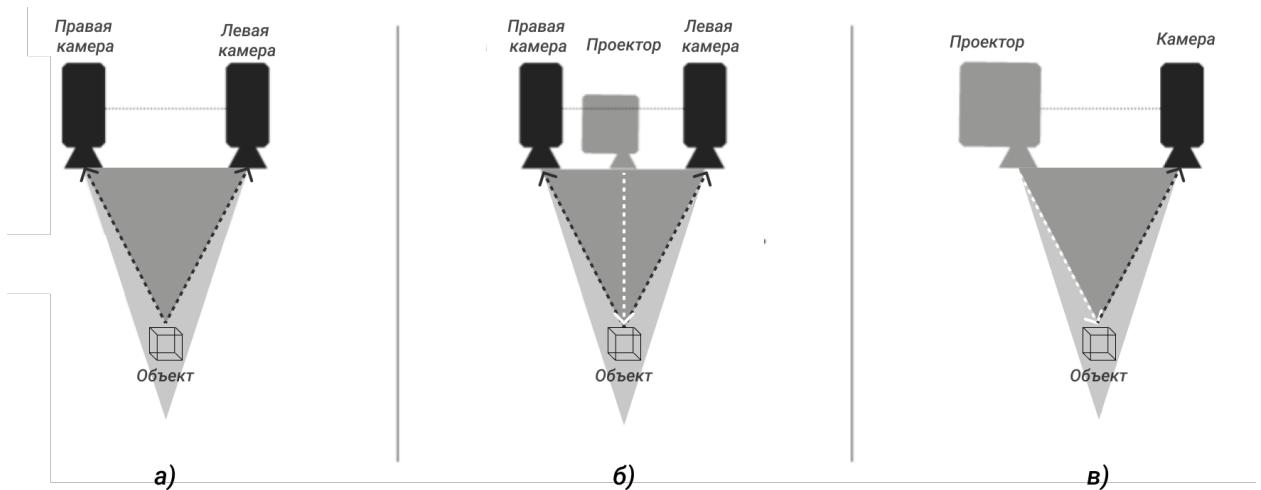


Рисунок 1.2 – а) Пассивная стереосистема; б) Активная стереосистема; в)
Подсветка структурированным светом

Описанные подходы хорошо работают для обычных камер с незначительными искажениями, где выполняется эпиполярная геометрия. В случае же камер со сверхшироким углом обзора в изображение вносятся существенные искажения, которые затрудняют поиск соответствий. На рисунке 1.3 вручную сопоставлены одни и те же точки на фрагментах двух изображений. Видно, что пары признаков больше не лежат на одной горизонтальной прямой. Учитывая распространённость таких камер в робототехнике, задача реализации систем стереоизрения на их основе является актуальной.

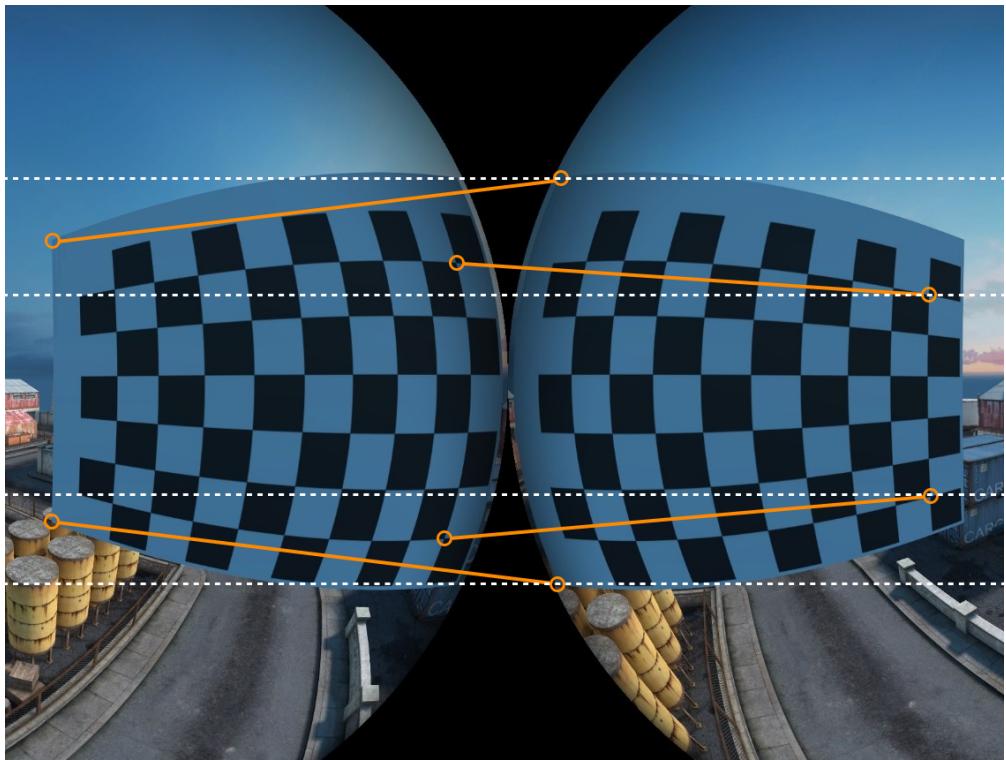


Рисунок 1.3 – Соответствия на снимках с fisheye-камер

Исследователи предложили несколько реализаций систем стереозрения, опирающихся на снимки со сверхширокоугольных камер. Например, с помощью пары таких камер реализована кольцевая область стереозрения с вертикальным полем зрения 65° [7]. Для этого две 245° камеры закреплены на противоположных концах жёсткого стержня. Это позволяет достигнуть панорамного обзора глубины с качеством, достаточным для осуществления автономной навигации и локализации [8], но конструктивно реализуема такая схема расположения камер только для летательных аппаратов.

Рохас и Оиши [9] решили отказаться от типичных для стереозрения этапов устранения искажений и ректификации и извлекать информацию о глубине напрямую по двум снимкам fisheye-камер. Для производства карт глубины используется свёрточная нейронная сеть, что требует существенных вычислительных мощностей - для достижения производительности в реальном времени разработчикам понадобилось использовать компьютер с ЦПУ i7-4770 и ГПУ NVIDIA GTX 1080Ti. В мобильном автономном роботе аналогичный по производительности вычислитель разместить может быть

проблематично. Кроме того, метод описан лишь для случая, когда обе камеры направлены в одном направлении.

Были предложены ортогональные системы, использующие большие площади перекрытия полей зрения объективов "рыбий глаз" для стереоизрения. Чжан и другие разработали особую систему стереоизрения (the special stereo vision system), использующую модуль из четырёх камер с углом зрения 185°

$\times 185^\circ$. Размещены камеры в одной плоскости под углом в 90° , как изображено на рисунке 1.4 [10]. Авторы отразили в работе калибровку разработанной ими системы и устранение искажений, но не проанализировали точность метода. Кроме того, в работе не отражены результаты оценки глубины.



Рисунок 1.4 – Модуль камер

1.2 Модели сверхширокоугольной камеры

Сложности, возникающие при использовании существующих алгоритмов стереоизрения в применении к сверхширокоугольным камерам, связаны с особенностями их оптической системы. Объективы этих камер имеют в своей основе сложную систему линз, схема которой вместе с примером получаемого изображения представлена на рисунке 1.5. Особенности этой системы позволяют достигать углов обзора выше 180° , но также являются причиной aberrации и характерных искажений изображения. Реалистично

моделировать ход лучей в подобных камерах нецелесообразно, поэтому исследователи прибегают к аппроксимациям, называемым моделями камер.

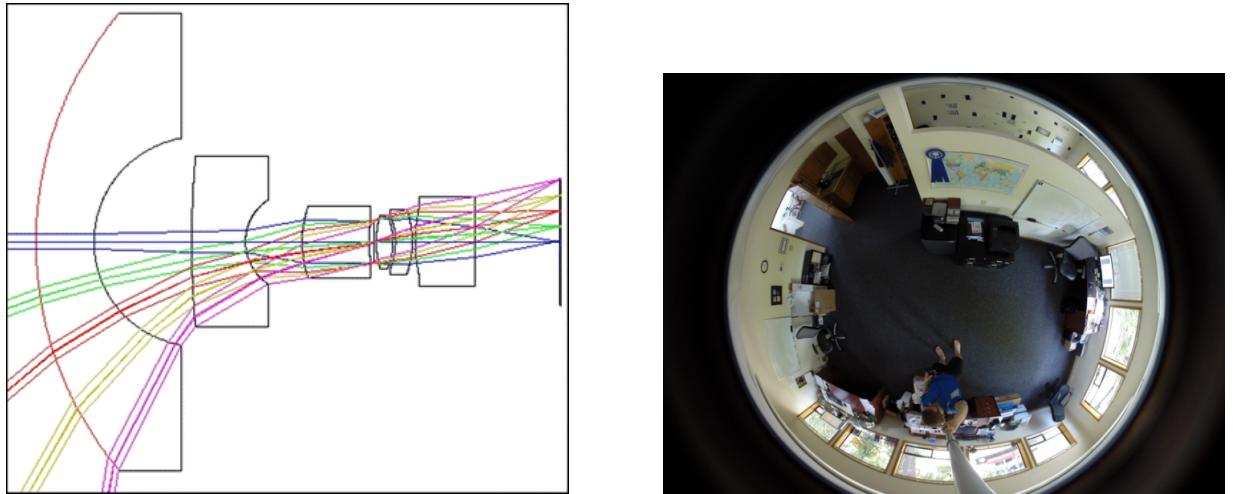


Рисунок 1.5 – Схема хода лучей объектива ”рыбий глаз” (слева), пример изображения (справа)[11]

Как видно по рисунку 1.6, модель проекции для камеры это функция, которая моделирует преобразование из точки трёхмерного пространства в области зрения камеры ($P = [x_c, y_c, z_c]^T$) в точку на плоскости изображения ($p = [u, v]^T$). Единичная полусфера S с центром в точке O_c описывает поле зрения. На ней также лежит точка P_C , являющаяся результатом обратной проекции. Угол θ является углом падения для рассматриваемой точки, а угол ϕ откладывается между положительным направлением оси x и O_ip .

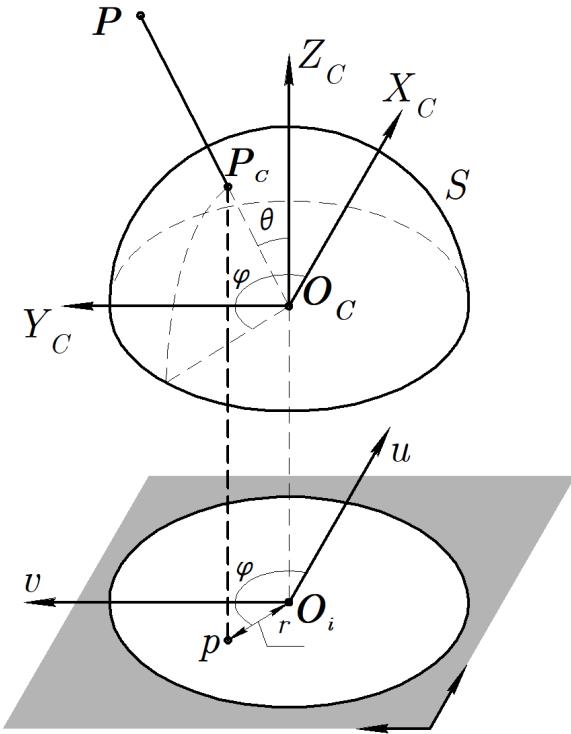


Рисунок 1.6 – Схема проекции точки трёхмерного пространства в точку на изображении

Модели камер включают в себя описания нескольких типов искажений, накладываемых линзой, но в сверхширокоугольных объективах самыми существенными являются радиальные - искажения, проявляющиеся сильнее ближе к краям изображения. Поэтому далее в этой секции модели будут рассматриваться именно с точки зрения описания радиальных искажений.

Перспективная проекция, которая обычно используется в качестве модели ортоскопической камеры, не способна спроектировать широкоугольное пространство на кадр конечного размера. Поэтому при описании и разработке fisheye-объективов опираются на другие виды проекций [12]:

$$r = 2f \tan(\theta/2), \quad (1.1)$$

$$r = f\theta, \quad (1.2)$$

$$r = 2f \sin(\theta/2), \quad (1.3)$$

$$r = f \sin(\theta), \quad (1.4)$$

Но реальные линзы не всегда в точности следуют заданным моделям, к тому же отличия в используемых параметрах усложняют процесс калибровки камер. По этой причине радиальные искажения выгоднее аппроксимировать многочленами [13], например, вида

$$\delta r = k_1 r^3 + k_2 r^5 + k_3 r^7 + \dots + k_n r^{n+2}, \quad (1.5)$$

где δr - радиальное отклонение от идеальной проекции луча; k_i - коэффициенты, описывающие внутренние параметры камеры.

В настоящий момент есть несколько распространённых моделей, аппроксимирующих реальные искажения подобных объективов. Модель Канналы и Брандта [13] для линз с радиально симметричными искажениями реализована в OpenCV и выражает их через угол падения луча света на линзу, а не расстояние от центра изображения до места падения, как это делалось в более ранних моделях. Авторы посчитали, что для описания типичных искажений достаточно пяти членов полинома. Таким образом, указанную модель можно записать следующими уравнениями:

$$\delta r = k_1 \theta + k_2 \theta^3 + k_3 \theta^5 + k_4 \theta^7 + k_5 \theta^9, \quad (1.6)$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \delta r(\theta) \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \end{pmatrix}, \quad (1.7)$$

где θ - угол падения луча, определяемый выбранным типом проекции; ϕ - угол между горизонтом и проекцией падающего луча на плоскость изображения; $r = \sqrt{x^2 + y^2}$ - расстояние от спроектированной точки до центра изображения; f - фокусное расстояние.

Также большое распространение получила модель Скамамуззы [14], которая легла в основу Matlab Omnidirectional Camera Calibration Toolbox. Она связывает точки на изображении с соответствующей им точкой в коор-

динатах камеры фьюк следующим образом

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \lambda \begin{pmatrix} u \\ v \\ a_0 + a_2 r^2 + a_3 r^3 + a_4 r^4 \end{pmatrix}, \quad (1.8)$$

где $a_0 \dots a_4$ - коэффициенты, описывающие внутренние параметры камеры; λ - масштабный коэффициент.

В качестве модели fisheye-камеры в этой работе принята именно модель Скарамуззы, так как она обладает точностью на уровне других [15] и удобным инструментом оценки внутренних параметров камеры.

Существуют и менее распространённые модели, не использующие полиномы для описания искажений. Одной из них является модель двух сфер [15]. Она положение пикселя, проектируя точку сначала на первичную сферу, потом на вторую сферу поменьше, сдвинутую на расстояние ξ , а затем в плоскость изображения камеры-обскуры, сдвинутой на $\frac{\alpha}{1-\alpha}$ относительно центра второй сферы. Модель проекции представлена на рисунке 1.7. Таким образом радиальные искажения можно описать всего двумя параметрами. Модель также интегрирована во многие популярные программы для калибровки камер (Basalt, Kalibr).

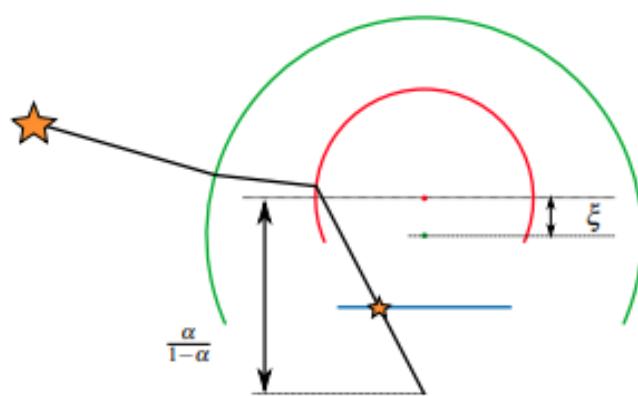


Рисунок 1.7 – Модель двух сфер

1.3 Обоснование выбора ПО

Разработку и первоначальные испытания алгоритма стереовидения целесообразно проводить в виртуальной среде. Это позволяет значительно

упростить разработку, так как уменьшает время на проверку гипотез и расходы на реальное оборудование, особенно в случае неудачных испытаний. Из-за этих факторов виртуальное моделирование в робототехнике приобрело широкое распространение и активно применяется, например, для разработки систем локализации и навигации беспилотного транспорта [16]. Возросшее качество компьютерной графики к тому же позволило моделировать реалистичное окружение, что особенно важно при работе с системами технического зрения.

Требованиями к виртуальной среде является возможность симулировать несколько широкоугольных камер и настраивать их параметры, легко интегрировать алгоритмы технического зрения и создать окружение, приближенное к тому, в котором может работать алгоритм в реальности. На данный момент исследователю доступен широкий выбор программного обеспечения, подходящего для этой задачи. В приложении А представлено сравнение имеющихся предложений по основным изложенным выше требованиям.

По результатам оценки собранные сведения принято решение проводить разработку в симуляторе Unity. Он позволяет подробно настраивать камеру и эмулировать fisheye-объектив, строить реалистичные сцены благодаря свободному импорту моделей, а при программировании в симуляторе можно использовать сторонние программы в виде динамически подключаемых библиотек. По функционалу так же подходит NVIDIA Isaac Sim, но от него пришлось отказаться из-за высоких системных требований и малой изученности продукта.

Разрабатываемое решение должно иметь возможность внедрения в ПО робота, поэтому должно реализовываться на одном из популярных и быстродейственных языков программирования. Учитывая необходимость интеграции с Unity и потребность использовать популярные библиотеки, выбран язык C++. Другим важным фактором является библиотека обработки изображений. В качестве основы для программной части была выбрана библиотека OpenCV, являющаяся стандартом при разработке систем технического зрения.

1.4 Выводы по разделу

Обзор современных моделей сверхширокоугольных камер позволил выбрать наиболее точную и удобную для калибровки. Был осуществлён обзор существующих систем стереозрения, применяющих камеры типа "рыбий глаз" с целью ознакомления с мировым опытом. Принято решение разрабатывать и тестировать предлагаемую систему стереозрения с применением виртуального моделирования.

Для моделирования выбрана среда разработки Unity. Для обработки изображений с камер выбрана библиотека OpenCV для языка программирования C++.

2 Система стереозрения

2.1 Описание системы стереозрения

Предлагаемая система стереозрения состоит из двух камер с объективами типа "рыбий глаз" $\geqslant 180^\circ$, расположенных ортогонально так, что две камеры имеют область пересечения полей зрения. В пространстве, наблюдаемом сразу двумя камерами возможна триангуляция и получение информации об объёме.

Рассмотрим организацию системы на примере робота-доставщика "Рover R3" компании Яндекс, который имеет на борту 4 сверхширокоугольные камеры, размещённые спереди, сзади и по бортам корпуса [17], что соответствует описанию системы. На рисунке 2.1, а показано реальное положение камер робота и их зон перекрытия, в которых может обеспечиваться получение информации о глубине при использовании описываемой системы. Рисунок 2.1, б демонстрирует эквивалентную по горизонтальному покрытию схема при использовании обычных камер с углом обзора 90° .

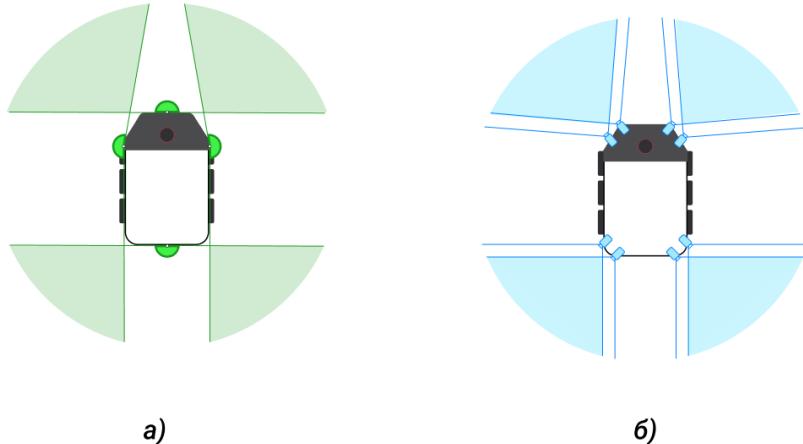


Рисунок 2.1 – Геометрическая модель бинокулярной системы стереозрения

Как можно заметить, системе на основе обычных камер нужно в два раза больше сенсоров, чтобы достичь той же зоны покрытия по горизонтали. Кроме того, традиционная система имеет меньшую зону покрытия по вертикали и не обеспечивает полный панорамный обзор. Все эти факторы делают

систему стереозрения на основе ортогонально расположенных сверхширокоугольных камер более предпочтительной для применения в робототехнике.

Применение существующих алгоритмов стереосопоставления предполагает наличие стереопары, удовлетворяющей условиям, описанным в секции 1.1. Такую стереопару можно получить, введя в систему для каждой сверхширокоугольной камеры виртуальную(ые) камеру-обскуру и направив её в сторону пересечения полей зрения, как если бы это была пара обычных камер. Процесс формирования виртуальной камеры-обскуры и алгоритм устранения искажений более подробно описаны в секции 2.4.

Далее для упрощения рассмотрения системы будет считаться, что оптические оси всех камер находятся в одной плоскости, а на ориентацию виртуальных камер влияет только угол β поворота в этой плоскости.

На рисунке 2.2 изображён простейший вариант системы с двумя камерами под углом 90° , который можно считать частью схемы, представленной на рисунке 2.1, или модуля, изображённого на рисунке 1.4.

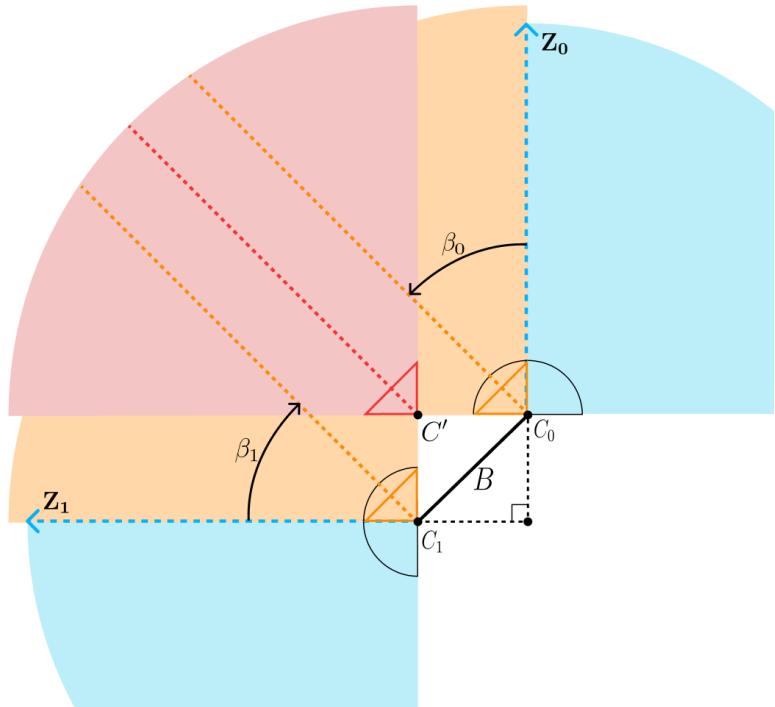


Рисунок 2.2 – Геометрическая модель бинокулярной системы стереозрения

Здесь область пересечения полей зрения камер C_0 и C_1 обозначенная красным. Эта область эквивалентна области пересечения полей зрения двух камер с полями зрения 90° (обозначены оранжевым), повёрнутых на $\beta_0 = \beta_1 = 45^\circ$ в сторону области интереса. Тогда B - база стереопары. Примеры изображений, полученных в такой конфигурации, приведён на рисунке 2.3.



Рисунок 2.3 – Пример исходных изображений с отмеченной областью интереса и снимков виртуальной стереопары

2.2 Виртуальное моделирование системы

Описанная система была смоделирована в среде Unity, её внешний вид представлен на рисунке 2.4. Мир Unity предназначен для базовой проверки работоспособности испытываемого принципа, поэтому не содержит подробной модели какого-либо робота. В нём присутствуют: плоскость земли, кронштейн, на котором сверхширокоугольные камеры закреплены под углом 90° , подвижный калибровочный узор и объекты-цели, предназначенные для оценки расстояния. В силу особенностей работы многих алгоритмов стереоразрезания эти объекты должны сильнотекстурированы [18].



Рисунок 2.4 – Внешний вид сцены в Unity

В Unity создание сцены происходит с использованием встроенных примитивов, импортированных файлов моделей популярных форматов или моделей из магазина ассетов, предлагающего обширную библиотеку объектов и текстур, повторяющих различные реальные объекты. В данной работе использовались как и примитивы для создания простых объектов, так и модели из магазина для имитации препятствий и окружения.

Для моделирования камеры "рыбий глаз" использовался аддон Dome Tools из магазина Unity Asset Store. Он позволяет моделировать сверхширокоугольные объективы с разным углом зрения в эквидистантной проекции [19]. При этом с точки зрения других искажений, не относящихся к моделированию правильной проекции, снимки с этой камеры получаются идеальными. Окно настроек камеры представлено на рисунке 2.5.

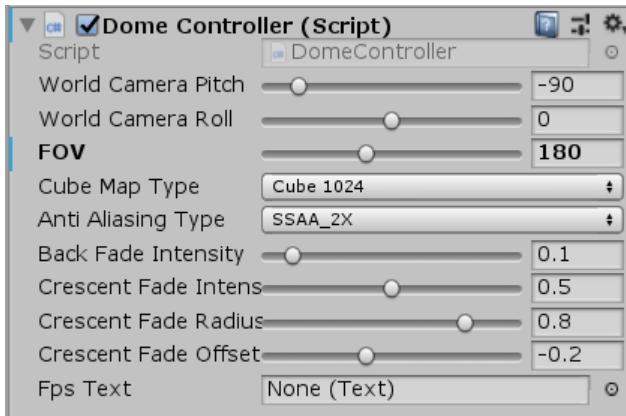


Рисунок 2.5 – Окно настроек виртуальной камеры в Unity

Для виртуальной камеры доступны настройки угла зрения и виньетки по краям изображения, а также различных параметров рендеринга, влияющих на уровень детализации получаемого изображения.

2.3 Передача изображений с виртуальных камер для обработки

Unity в качестве основного языка программирования использует C# и не позволяет импортировать выбранную библиотеку технического зрения OpenCV напрямую. Однако как уже упоминалось в секции 1.3, данная среда позволяет интегрировать плагины, в том числе написанные на других языках программирования, в виде динамически подключаемых библиотек (DLL). Именно этот подход был использован для передачи изображений - была разработана библиотека с функциями для обработки снимков, которая компилировалась отдельно и затем импортировалась в Unity.

Для реализации описанного в этой работе принципа и упрощения разработки в библиотеке были реализованы следующие функции:

- initialize - выделяет память под нужные структуры, заполняет таблицу поиска и создаёт окна для отображения будущих изображений.
- getImages - передаёт изображения в библиотеку и производит обработку изображений в соответствии с аргументами.
- takeScreenshot - производит ту же обработку входного изображения, что и предыдущая функция, но результат сохраняет в файл.
- processImage - передаёт обратно в скрипт уже обработанные изображения для удобного отображения в интерфейсе Unity.

- `terminate` - высвобождает память по завершении работы симуляции.

Обработка, упомянутая в пунктах с функциями `getImages` и `takeScreenshot`, заключается в конвертации цветового пространства изображений (из `RGBA`, используемого в Unity, в `BGR`, используемый в OpenCV), горизонтальном зеркальном отображении (из-за разного положения точки отсчёта координат пикселей) и, наконец, устранив искажений в области интереса. Так как для наилучшей работы стереосопоставления снимки должны быть синхронизированы (то есть получены камерами в один момент времени), обработка производится сразу для двух изображений.

Также написан специальный скрипт, который управляет процессом взаимодействия симуляции и библиотеки. Эта программа имеет окно настроек, которое позволяет задать используемые камеры, элементы управления параметрами и выбрать область интереса с помощью задания углов. Внешний вид этого окна представлен на рисунке 2.6.

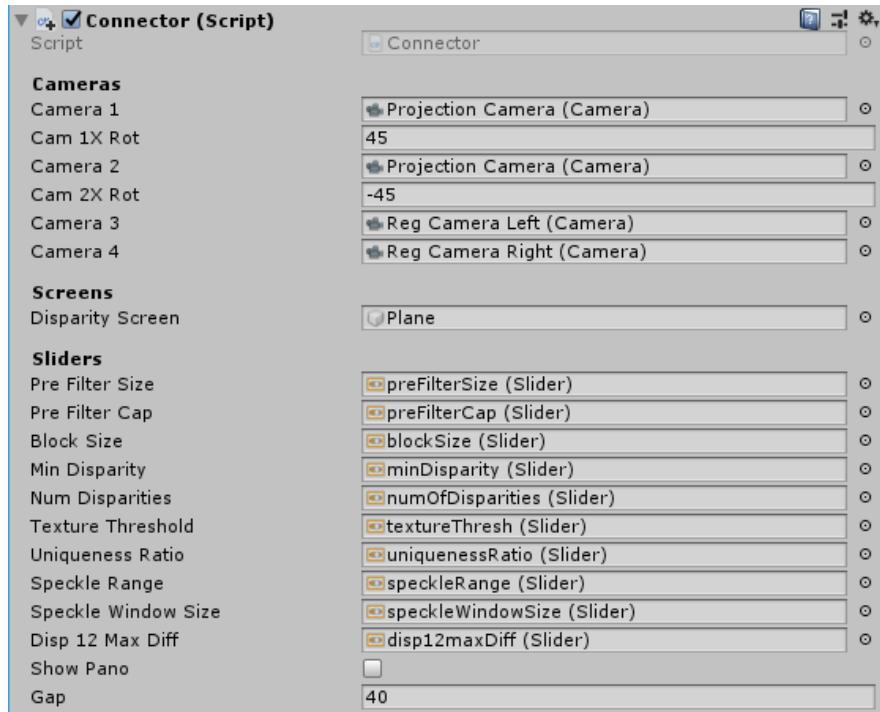


Рисунок 2.6 – Внешний вид окна настроек скрипта

Сразу после запуска симуляции происходит запуск дополнительных вычислительных потоков для обработки кадров с каждой пары камер и передача настроек скриптом в библиотеку для построения таблиц поиска.

Использование нескольких потоков позволяет не блокировать работу симуляции на время обработки изображений. Далее каждое обновление кадра скрипт считывает изображения с камер и помещает их в память соответствующего потока. Поток же работает независимо и обрабатывает каждую следующую пару изображений после готовности предыдущей.

2.4 Алгоритм устранения искажений

Распространённые библиотеки машинного зрения предлагают готовые к использованию классы и функции, позволяющие устранять искажения fisheye-объективов. Например, в составе MATLAB Computer Vision Toolbox присутствует инструмент Camera Calibrator, позволяющий в несколько простых этапов произвести оценку внутренних и внешних параметров камеры. Пользователю надо лишь загрузить набор изображений, содержащих в себе калибровочный узор, выбрать модель камеры (стандартная или сверхширокоугольная) и запустить автоматический процесс калибровки. По завершению процесса программа показывает точность калибровки и может отображать скорректированные изображения. Результаты можно экспорттировать в рабочую область MATLAB для дальнейшей обработки изображений. Интерфейс программы представлен на рисунке 2.7.

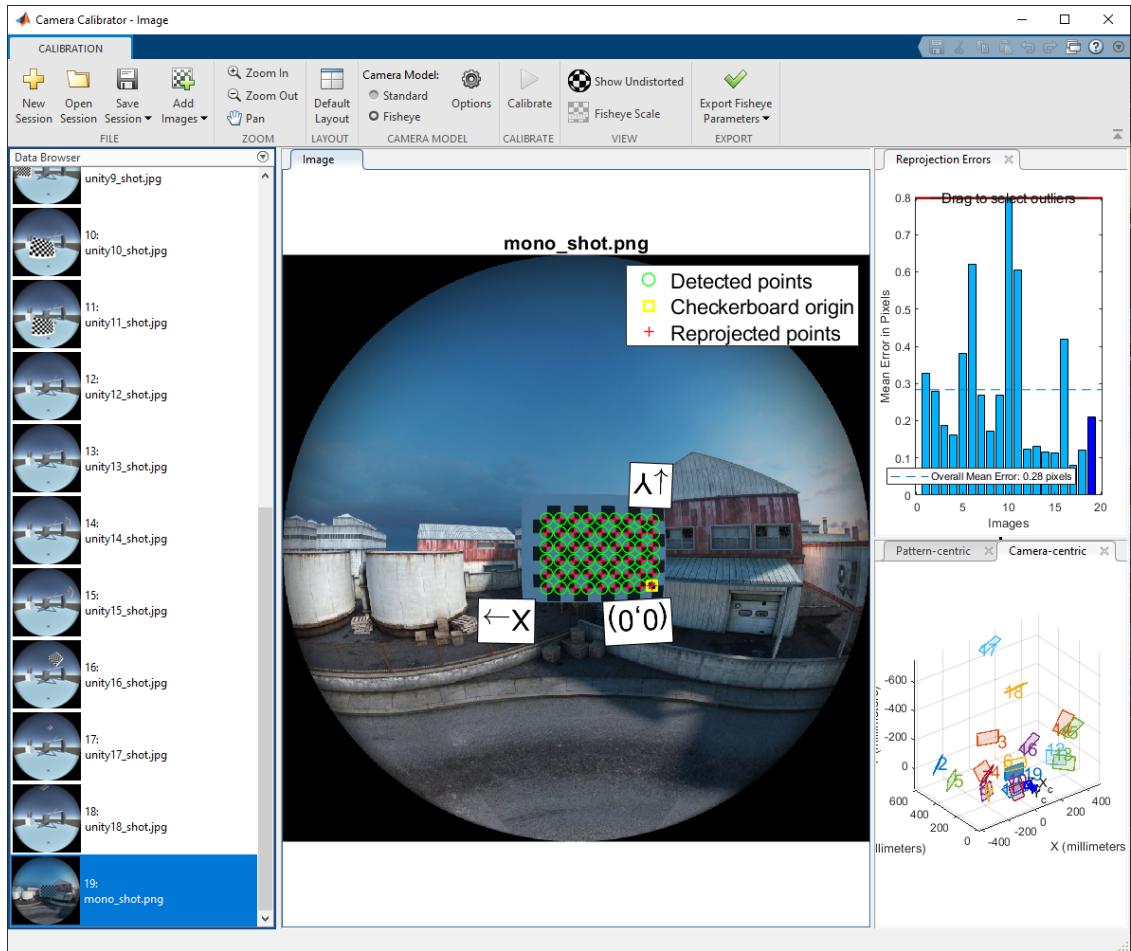


Рисунок 2.7 – Интерфейс приложения MATLAB Camera Calibrator

Однако на практике этот и другие подобные методы весьма ограничены. И MATLAB, и OpenCV устраниют искажения строго в центральной части изображения, не позволяя выбирать желаемое направление обзора. Это ограничивает применимость существующих инструментов только для копланарного расположения камер. Поэтому для реализации предлагаемой системы стереозрения был разработан собственный алгоритм устранения искажений. Схема геометрического принципа, лежащего в основе этого алгоритма, представлена на рисунке 2.8.

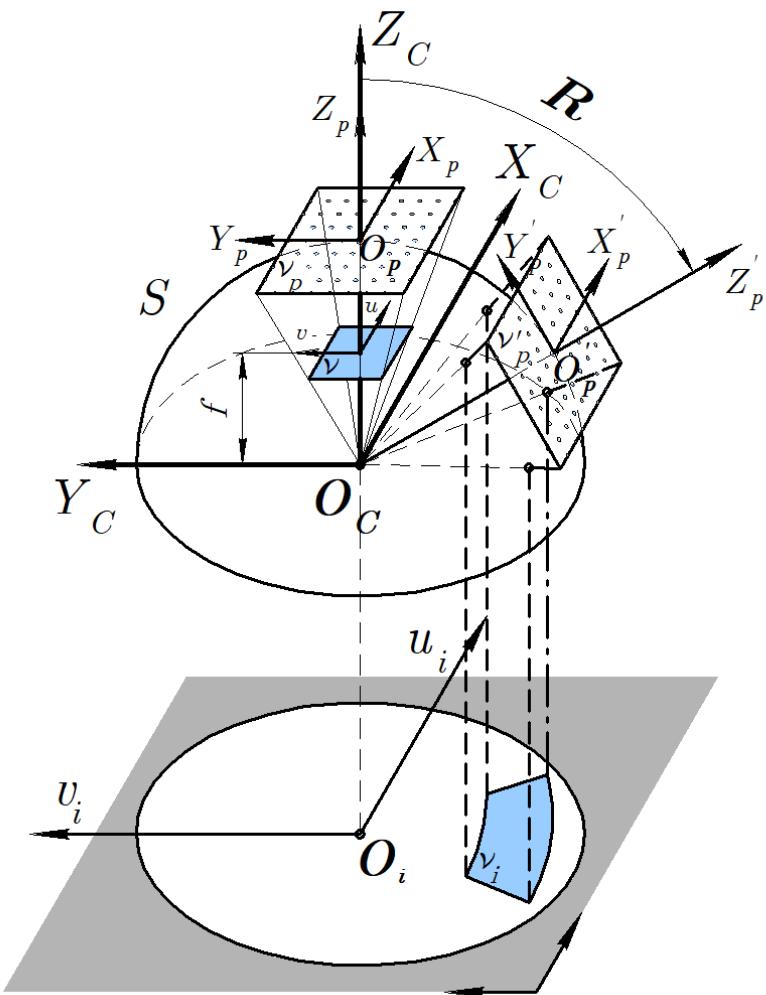


Рисунок 2.8 – Схема принципа устранения искажений

Цель алгоритма - найти, куда на выходном изображении с перспективной проекцией проектируются все пиксели из выбранного участка входного сверхширокоугольного изображения. Сделать это можно, выполнив обратное преобразование (1.8) для каждого пикселя fisheye-снимка и затем прямое преобразование модели камеры-обскуры для получения результирующей проекции. Однако такой подход приведёт к возникновению дефектов из-за несовпадения частот дискретизации двух изображений. Поэтому вместо этого алгоритм сначала выполняет обратное преобразование для каждого пикселя итогового изображения ν , находя таким образом соответствующую

ему точку в системе координат камеры (X_c, Y_c, Z_c)

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} u * z_c / f \\ v * z_c / f \\ z_c \end{bmatrix}, \quad (2.1)$$

где f - фокусное расстояние.

Набор таких точек формирует прямоугольную область ν_p с центром в точке O_p и является плоскостью изображения виртуальной камеры-обскуры с оптической осью Z_p . Поворот точек, входящих в ν_p , с помощью матрицы вращения R образует ν'_p и позволяет таким образом задать направление обзора и ориентацию виртуальной камеры.

$$\begin{bmatrix} x'_p \\ y'_p \\ z'_p \end{bmatrix} = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} R. \quad (2.2)$$

$$R = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & \sin \beta \\ 0 & -\sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma \\ 0 & 1 & 0 \\ \sin \gamma & 0 & \cos \gamma \end{bmatrix}, \quad (2.3)$$

где координаты с индексом p - координаты точек соответствующей плоскости ν в системе камеры, α, β, γ - углы Эйлера, управляющие ориентацией виртуальной камеры-обскуры.

Тогда обратная fisheye-проекция точек из ν'_p позволяют получить область ν_i исходного изображения с искомыми пикселями. Таким образом, зная как проектируется каждая точка из ν в ν_i , возможно обратно перенести информацию о цвете на итоговое изображение.

Так как в используемой модели искажения хода луча считаются центрально симметричными и зависят только от его удаления от центра изображения, рассмотрим ход падающего луча в координатах (Z_c, ρ) , изображённый на рисунке 2.9.

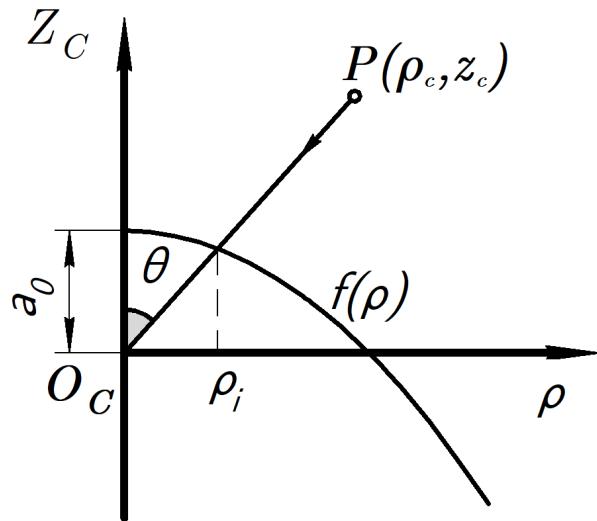


Рисунок 2.9 – Нахождение обратной проекции для используемой модели

Для модели (1.8) обратная проекция записывается как

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \frac{x_c}{\lambda} \\ \frac{y_c}{\lambda} \end{bmatrix}, \quad (2.4)$$

где $\lambda = \rho_c / \rho_i$ - масштабный коэффициент.

Для нахождения ρ_i был применён метод последовательных приближений. Блок-схема алгоритма, реализующего обратную проекцию, изображена на рисунке 2.11. Сравнение изображений инструмента калибровки и описанного алгоритма для центральной области изображения представлено на рисунке 2.10.

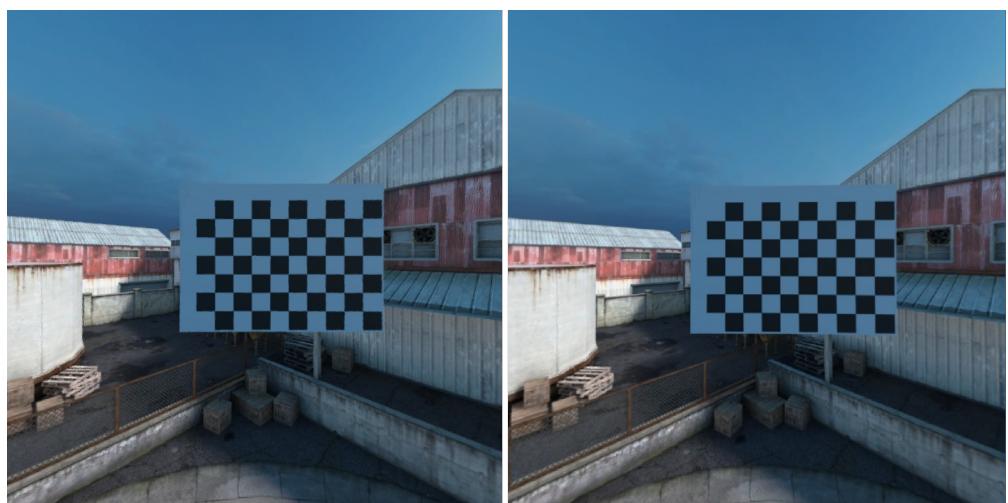


Рисунок 2.10 – Изображения, скорректированные алгоритмом (слева) и MATLAB (справо)

Очевидно, весь процесс преобразования $\nu \rightarrow \nu_i$ требует выполнения существенного количества математических операций, что негативно сказывается на скорости, с которой алгоритм может обрабатывать изображения в реальном времени. Однако при неизменных параметрах модели алгоритм достаточно выполнить лишь один раз, записав результат в таблицу поиска - структуру данных, которая позволяет дальнейшие преобразования проводить по уже известным соотношениям между пикселями. Это позволяет применять алгоритм для устранения искажений в реальном времени.

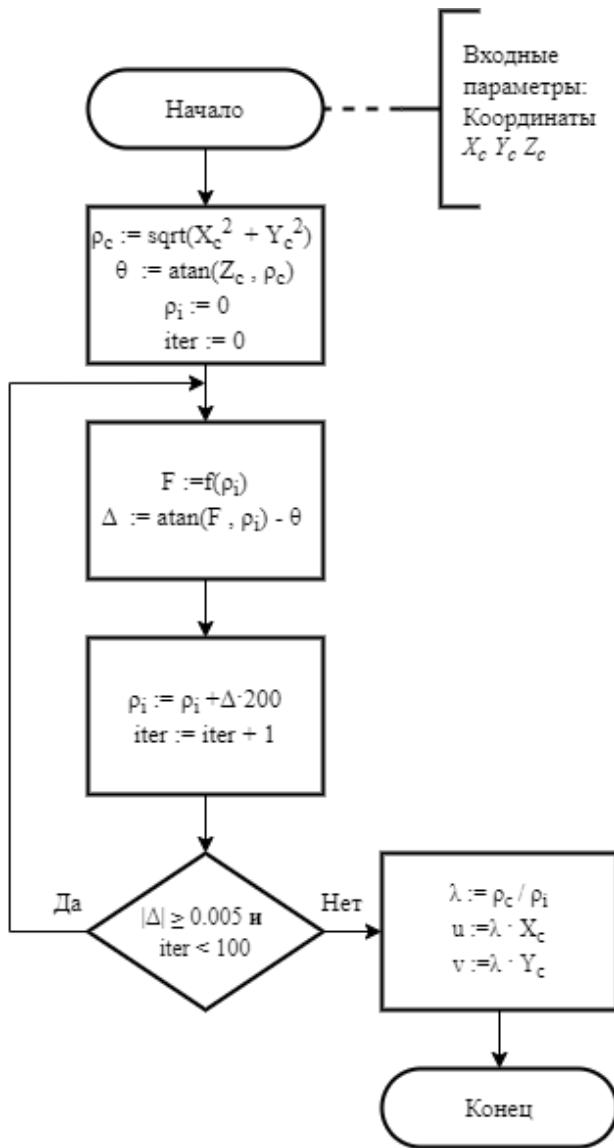


Рисунок 2.11 – Блок-схема алгоритма обратной проекции

2.5 Выводы по разделу

Описан принцип устранения искажений сверхширокоугольных линз с выбором области интереса. Разработан алгоритм нахождения обратной

проекции для fisheye-изображения. Описано устройство системы стереовидения. Разработана её виртуальная модель в среде Unity вместе с алгоритмом передачи изображений с виртуальной камеры в программу обработки изображений. Виртуальная модель системы позволяет перейти к её испытаниям.

3 Экспериментальное исследование системы стереозрения

Описанная в предыдущей главе виртуальная модель системы стереозрения позволяет проводить с ней испытания для оценки работоспособности и сравнения с аналогами в контролируемой среде.

Результатом работы алгоритма стереозрения является карта глубины воспринимаемого пространства. Подобные карты могут использоваться, например, алгоритмами навигации и локализации для построения карты окружения робота. Точность работы этих алгоритмов зависит от того, насколько точно карта глубины передаёт реальную информацию о форме объектов. Оценить эту характеристику для отдельной системы стереозрения проблематично, так как она зависит от множества факторов. Поэтому оценка качества работы системы проведена в сравнении с виртуальной моделью традиционной стереопары.

Виртуальная сцена позволяет разместить сразу несколько объектов в одной точке пространства, таким образом возможно в существующую модель в Unity добавить ещё 2 камеры, совпадающие по параметрам и положению с виртуальными камерами на рисунке 2.2. Разрешение этих камер выбрано исходя из размеров проекции ν_i области интереса на широкоугольном снимке. Для камер "рыбий глаз" с разрешением 1080 * 1080 пикселей она составляет 287482 пикселяй, что аналогично камере с разрешением 540 * 540. В результате получена эталонная стереопара для сравнения.

3.1 Оценка качества устранения искажений

Добавление в виртуальную модель эталонной камеры позволяет оценить качество устранения искажений путём сравнения её снимков со снимками виртуальной камеры-обскуры модуля устранения искажений.

Параметры полинома (1.8) для устранения искажений получены с помощью MATLAB Camera Calibrator и занесены в код библиотеки обработки

изображений. Снимки после устранения искажений и с эталонной камеры показаны на рисунке 3.1.



Рисунок 3.1 – Слева - снимок после устраниния искажений; справа - эталонный снимок.

Визуально снимки очень похожи, однако при более детальном рассмотрении на левом изображении заметна большая резкость, вызванная, вероятно, округлениями чисел в процессе проекции. Также присутствуют малозаметные искажения геометрии. Более явно эти дефекты можно увидеть на разностном изображении, представленном на рисунке 3.2.

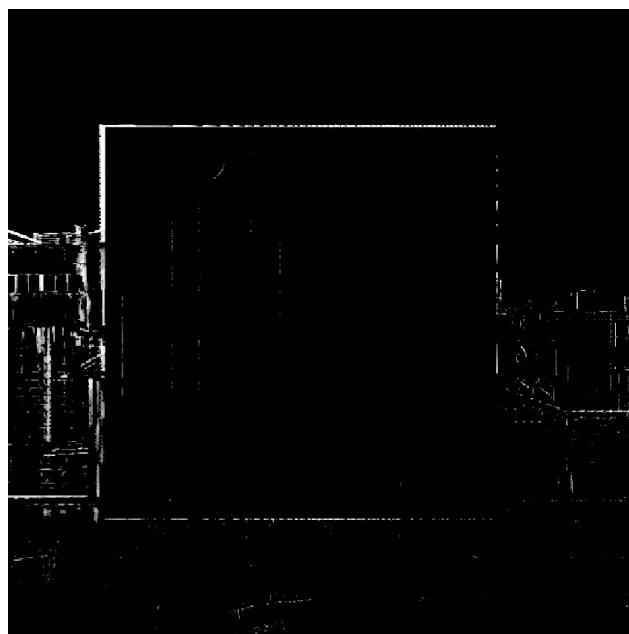


Рисунок 3.2 – Разностное изображение (резкость увеличена). Чем светлее участок, тем сильнее различия

Анализ этого изображения подтверждает различия в резкости обеих картинок и наличие искажений преимущественно в левой части снимка. Тем не менее, искажения достаточно несущественны и проявляются лишь близь краёв исходного изображения, что позволяет считать подобные снимки пригодными к применению в системе стереоизрения.

3.2 Исследование точности оценки глубины

В качестве целевой поверхности выбрана виртуальная плоскость с нанесённой на неё нерегулярной текстурой высокого разрешения. Размеры и положение плоскости относительно камер известно с высокой точностью, что позволяет сравнить результаты стереореконструкции с реальным положением целевого объекта и оценить ошибку. В качестве метрики оценки выбрано среднее квадратичное отклонение положения точек от модели плоскости.

Исследования проведены в MATLAB, снимки получены с помощью виртуальной модели при одинаковых условиях освещённости и постоянных настройках всех компонентов системы. Ширина базы выбрана равной 1 метру. Сначала для обеих стереопар проведена калибровка по снимкам узора шахматной доски с помощью методики [20], что позволило получить внутренние и внешние параметры камер стереопары. Далее происходит построение карты расхождений методом полу-глобального сопоставления [21] и 3D-реконструкция сцены. Результат реконструкции представлен в виде облака точек, изображённого на рисунке 3.3.

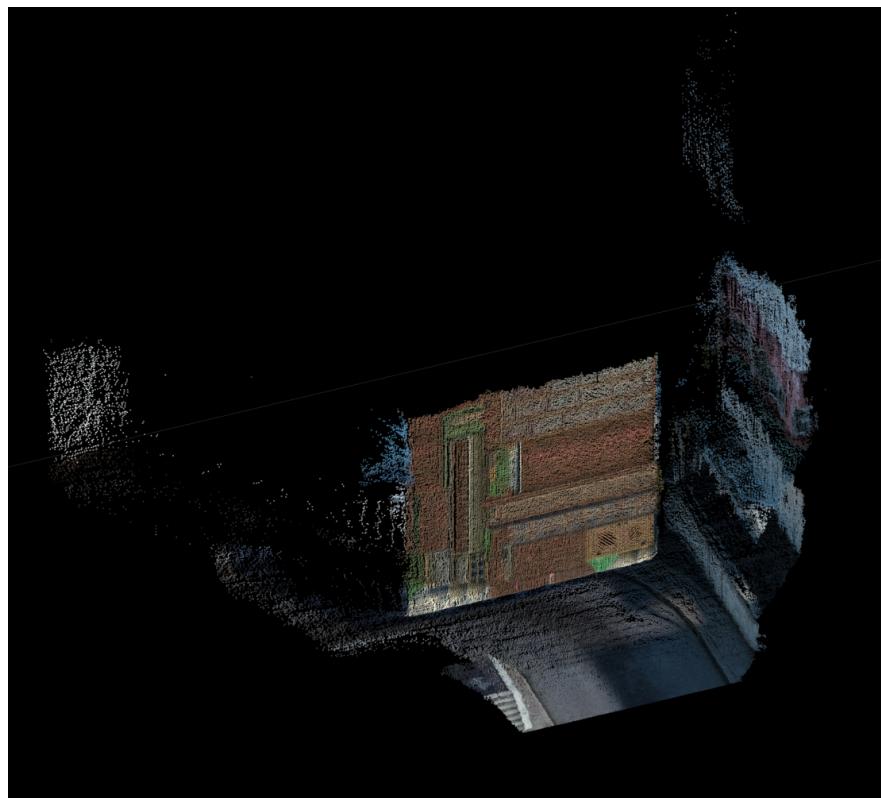


Рисунок 3.3 – Неочищенное облако точек

По заданным параметрам строится модель целевой плоскости, а точки за пределами её окрестности отбрасываются. Затем скрипт перебирает все оставшиеся точки и рассчитывает среднее квадратичное отклонение по длине нормали от точки к плоскости. Описанный алгоритм применён к снимкам целевой плоскости на разных расстояниях с эталонной и исследуемой стереопар. Результаты приведены на графике 3.4.

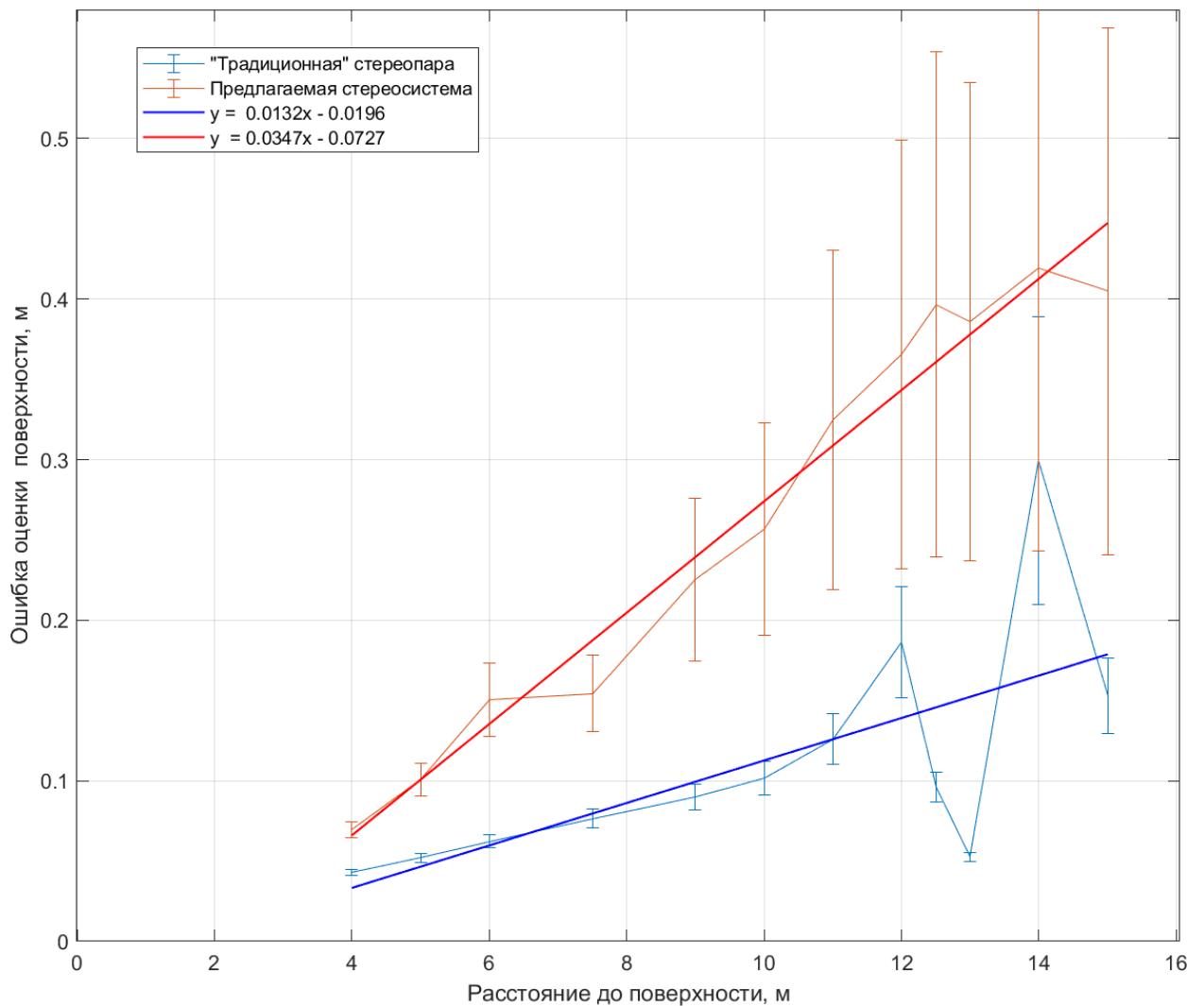


Рисунок 3.4 – График среднеквадратичной ошибки и дисперсии в зависимости от расстояния для эталонной и исследуемой стереопар

Как видно из графика, ошибка оценки поверхности в исследуемой системе в среднем в два раза больше этого показателя для традиционной стереопары. Меньшая точность исследуемой системы связана с влиянием факторов, описанных в предыдущей секции, на алгоритм стереосопоставления. Тем не менее она остаётся в пределах 3% от реального расстояния.

3.3 Выводы по разделу

Исследованы снимки, полученные с помощью виртуальных камер. Выполнено их сравнение с эталонными изображениями, которое продемонстрировало пригодность снимков для использования в системах стереозрения.

В виртуальную среду добавлена традиционная стереопара для сравнения с разработанной системой. Проведён эксперимент по оценке качества

облака точек, полученного с помощью предлагаемого решения. Результаты можно считать удовлетворительными для систем подобного класса.

ЗАКЛЮЧЕНИЕ

В ходе работы рассмотрены существующие системы стереозрения, в особенности использующие информацию со сверхширокоугольных камер. Изучены модели, описывающие подобные камеры. Рассмотрено ПО для виртуального моделирования робототехнических комплексов и систем технического зрения. Выбрано ПО для моделирования и разработки системы стереозрения, использующей объективы "рыбий глаз".

Разработан и описан алгоритм устранения искажений в области интереса и его математическая модель. Описана возможная конфигурация системы стереозрения, рассмотрен принцип работы её фрагмента. Этот фрагмент считать был смоделирован в среде Unity. В процессе размещены и настроены камеры, подготовлены объекты для калибровки и дальнейших испытаний системы. Реализована передача изображений с виртуальных камер для обработки алгоритмами компьютерного зрения. С помощью снимков из виртуальной модели исследовано качество устранения искажений и выполнено сравнение точности оценки глубины предлагаемой системы и варианта с обычными камерами. Результаты позволяют рассмотреть применение системы с реальными камерами и указывают на возможные доработки.

Дальнейшая работа будет сконцентрирована на оптимизации и повышении точности алгоритма устранения искажений, разработке способа автоматической установки параметров системы при разных конфигурациях камер. После этого можно будет приступить к испытаниям качества построения карты и локализации с применением описанной системы сначала с виртуальным, а затем и с реальным роботом.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Варлашин В. В.* Алгоритмическое обеспечение системы кругового обзора для мобильных робототехнических комплексов / Варлашин Виктор Витальевич. — Санкт-петербург, 2018.
2. Real-Time Bird’s Eye Surround View System: An Embedded Perspective / M. Al-Hami [и др.] // Applied Artificial Intelligence. — 2021. — Т. 35, № 10. — С. 765—781.
3. Hartley R. I., Zisserman A. Multiple View Geometry in Computer Vision. — Second. — Cambridge University Press, ISBN: 0521540518, 2004.
4. Вахитов А., Гуревич Л., Павленко Д. Обзор алгоритмов стереозрения // Стохастическая оптимизация в информатике. — 2008. — Т. 4, № 1.
5. Л. Шатиро Д. С. Компьютерное зрение / пер. с англ. А. А. Богуславский. — М. : БИНОМ. Лаборатория знаний, 2015. — 752 с. — ISBN 5-94774-384-1.
6. Khoshelham K., Oude Elberink S. Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications // Sensors (Basel, Switzerland). — 2012. — Дек. — Т. 12. — С. 1437—54.
7. Gao W., Shen S. Dual-fisheye omnidirectional stereo // 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). — 2017. — С. 6715—6722.
8. Autonomous aerial robot using dual-fisheye cameras / W. Gao [и др.] // Journal of Field Robotics. — 2020. — Т. 37, № 4. — С. 497—514.
9. Roxas M., Oishi T. Real-Time Variational Fisheye Stereo without Rectification and Undistortion. — 2019. — arXiv: 1909 . 07545 [cs.RO].
10. The study of calibration and epipolar geometry for the stereo vision system built by fisheye lenses / B. Zhang [и др.] // Intelligent Robots and Computer Vision XXXII: Algorithms and Techniques. Т. 9406 / под ред. J. Röning,

- D. Casasent. — International Society for Optics, Photonics. SPIE, 2015. — C. 167—176. — URL: <https://doi.org/10.1117/12.2079033>.
11. *Technologies T.* Сайт производителя сверхширокоугольных камер Theia. — 2021. — URL: <https://www.theiatech.com/ty180ir> (дата обр. 14.01.2022).
 12. Miyamoto K. Fish Eye Lens // J. Opt. Soc. Am. — 1964. — Авг. — Т. 54, № 8. — С. 1060—1061.
 13. Kannala J., Brandt S. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2006. — Т. 28, № 8. — С. 1335—1340.
 14. Scaramuzza D., Martinelli A., Siegwart R. A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion // Fourth IEEE International Conference on Computer Vision Systems (ICVS'06). — 2006. — С. 45—45.
 15. Usenko V., Demmel N., Cremers D. The Double Sphere Camera Model // CoRR. — 2018. — Т. abs/1807.08957. — arXiv: 1807 . 08957. — URL: <http://arxiv.org/abs/1807.08957>.
 16. On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward / H. Choi [и др.] // Proceedings of the National Academy of Sciences. — 2021. — Т. 118, № 1. — ISSN 0027-8424.
 17. Лавренюк А. Встречаем ровер третьего поколения: история создания робота-курьера Яндекса. — 2021. — URL: <https://habr.com/ru/company/yandex/blog/590997/> (дата обр. 15.01.2022).
 18. Brown M., Burschka D., Hager G. Advances in computational stereo // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2003. — Т. 25, № 8. — С. 993—1008.

19. *VR Z*. A Powerful, Open-source plugin for the Unity game engine that allows anyone to convert their content to the Fulldome planetarium format. — 2021. — URL: <https://zubr.co/work/dome-tools/> (дата обр. 17.01.2022).
20. *Heikkila J., Silven O.* A four-step camera calibration procedure with implicit image correction // Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. — 1997. — C. 1106—1112.
21. *Hirschmuller H.* Accurate and efficient stereo processing by semi-global matching and mutual information // 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). T. 2. — 2005. — 807—814 vol. 2.

ПРИЛОЖЕНИЕ А

Сравнение ПО для симуляции

Название	Симуляция fisheye-камер	Реалистичное моделирование	Интеграция кода	Доступность
Gazebo	Возможна	Возможно	Возможна посредством ROS	Бесплатно
RoboDK	Нет	Затруднено	Нет	От 145€
Webots	Затруднена	Возможно	Возможна	Бесплатно
CoppeliaSim	Затруднена	Затруднено	Возможна	Бесплатно
NVIDIA Isaac Sim	Возможна	Возможно	Возможна	Бесплатно
CARLA	Затруднена	Возможно	Возможна	Бесплатно
Unity	Возможна	Возможно	Возможна	Бесплатно