

## СОДЕРЖАНИЕ

1	Введение . . . . .	5
2	Обзор алгоритмов стереозрения / моделей fisheye камер / оснащённости роботов соответствующими камерами . . . . .	7
3	Обоснование выбора ПО . . . . .	8

## 1 ВВЕДЕНИЕ

За последние годы был совершён существенный прогресс в доступности и точности сенсоров, позволяющих мобильным роботам осуществлять оценку окружающего пространства. Такие информационно-измерительные устройства как лидары, сонары (и что-нибудь ещё) стали основной опорой алгоритмов для алгоритмов автономной навигации и локализации. Тем не менее в роботах по-прежнему присутствуют оптические системы, так как они дают наиболее читаемую информацию для оператора в случаях, когда его вмешательство необходимо. Существенная часть современных мобильных роботов имеют у себя на борту камеры с широким ( ) или сверхшироким ( ) углами обзора, так как они, хоть и вносят искажения в воспринимаемую картину, позволяют охватить больше окружающего пространства. Набор таких камер может составлять систему кругового обзора [?], позволяющую оператору видеть не только в любом направлении, но даже с видом от третьего лица []. Схема подобной системы приведена на рисунке 1.1.



Рис. 1.1 – Системы кругового обзора

В случае автономных мобильных роботов подобные системы включаются лишь по необходимости, но при этом могут быть весьма дорогостоящими и занимать место в корпусе. Согласно схемам на рисунке 1.1 широкоугольные камеры в системах кругового обзора роботов часто имеют области пересечения их полей зрения, что позволяет проводить оценку глубины / использовать алгоритмы стереозрения / использовать камеры как стереопару. Это даёт роботу вспомогательный (или единственный) источник трёхмерной информации об окружении без дополнительных расходов.

Однако значительные радиальные искажения изображения, вызванные особенностями используемых объективов, и тот факт, что области пересечения обычно расположены ближе к краям изображения, не позволяют использовать известные алгоритмы стереозрения.

Целью работы является разработка и изучение точности алгоритма стереозрения, работающего с изображениями, полученными с использованием широкоугольных и сверхширокоугольных объективов.

В ходе работы решаются следующие задачи:

- Обзор современных алгоритмов стереозрения / алгоритмов калибровки изображений широкоугольных камеры.
- Обоснование выбора программного обеспечения, используемого для разработки и тестирования алгоритма.
- Разработка алгоритма устранения искажений fisheye-объектива и системы стереозрения на его основе.
- Оценка точности оценки глубины в виртуальной среде.

## 2 ОБЗОР АЛГОРИТМОВ СТЕРЕОЗРЕНИЯ / МОДЕЛЕЙ FISHEYE КАМЕР / ОСНАЩЁННОСТИ РОБОТОВ СООТВЕТСТВУЮЩИМИ КАМЕРАМИ

TODO: в этой секции стоит провести обзор. Пока не очень понятно, чего конкретно. Можно наверное и всего понемногу

### 3 ОБОСНОВАНИЕ ВЫБОРА ПО

Разработку и первоначальные испытания алгоритма стереозрения 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 []. Возросшее качество компьютерной графики к тому же позволило моделировать реалистичное окружение, что особенно важно при работе с системами технического зрения.

Для разработки алгоритма, описанного в этой работе, нужна виртуальная среда, в которой можно симулировать несколько широкоугольных камер и настраивать их параметры, легко интегрировать алгоритмы технического зрения и создать окружение, приближенное к тому, в котором будет работать алгоритм. На данный момент исследователю доступен широкий выбор программного обеспечения, подходящего для этой задачи. В таблице 3.1 представлено сравнение имеющихся предложений по основным изложенным выше требованиям.

По результатам оценки собранные сведения было 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 новизны продукта.

Разрабатываемое решение должно иметь возможность внедрения в ПО робота, поэтому должно реализовываться на одном из популярных и быстрых языков программирования. Учитывая необходимость интеграции с Unity и потребность использовать популярные библиотеки, был выбран язык C++. Другим важным фактором является основная библиотека обработки изобра-

Таблица 3.1 – Сравнение ПО для симуляции

Название	Симуляция fisheye-камер	Реалистичное моделирование	Интеграция кода	Доступность
Gazebo	Возможна	Затруднено	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

ражений. В качестве основы для программной части была выбрана библиотека OpenCV, являющаяся стандартом при разработке систем технического зрения. Она доступна к использованию со множеством языков программирования, но наилучшую производительность показывает именно с C++ [1].

### 3 СПИСОК ЛИТЕРАТУРЫ