

**Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО  
ITMO University**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
GRADUATION THESIS**

**Исследование алгоритмов одновременной локализации и построения карты в задачах  
мобильной робототехники**

**Обучающийся / Student** Курчавый Виталий Вячеславович

**Факультет/институт/кластер/ Faculty/Institute/Cluster** факультет систем управления и  
робототехники

**Группа/Group** R3438

**Направление подготовки/ Subject area** 15.03.06 Мехатроника и робототехника

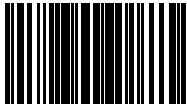
**Образовательная программа / Educational program** Робототехника 2020

**Язык реализации ОП / Language of the educational program** Русский

**Квалификация/ Degree level** Бакалавр

**Руководитель ВКР/ Thesis supervisor** Шаветов Сергей Васильевич, доцент, кандидат  
технических наук, Университет ИТМО, факультет систем управления и робототехники,  
доцент (квалификационная категория "ординарный доцент")

Обучающийся/Student

Документ подписан	
Курчавый Виталий Вячеславович	
22.05.2024	

(эл. подпись/ signature)

Курчавый  
Виталий  
Вячеславович

(Фамилия И.О./ name  
and surname)

Руководитель ВКР/  
Thesis supervisor

Документ подписан	
Шаветов Сергей Васильевич	
16.05.2024	

(эл. подпись/ signature)

Шаветов Сергей  
Васильевич

(Фамилия И.О./ name  
and surname)

**Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО  
ITMO University**

**ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ /  
OBJECTIVES FOR A GRADUATION THESIS**

**Обучающийся / Student** Курчавый Виталий Вячеславович

**Факультет/институт/кластер/ Faculty/Institute/Cluster** факультет систем управления и робототехники

**Группа/Group** R3438

**Направление подготовки/ Subject area** 15.03.06 Мехатроника и робототехника

**Образовательная программа / Educational program** Робототехника 2020

**Язык реализации ОП / Language of the educational program** Русский

**Квалификация/ Degree level** Бакалавр

**Тема ВКР/ Thesis topic** Исследование алгоритмов одновременной локализации и построения карты в задачах мобильной робототехники

**Руководитель ВКР/ Thesis supervisor** Шаветов Сергей Васильевич, доцент, кандидат технических наук, Университет ИТМО, факультет систем управления и робототехники, доцент (квалификационная категория "ординарный доцент")

**Характеристика темы ВКР / Description of thesis subject (topic)**

**Тема в области фундаментальных исследований / Subject of fundamental research:** нет / not

**Тема в области прикладных исследований / Subject of applied research:** да / yes

**Основные вопросы, подлежащие разработке / Key issues to be analyzed**

Техническое задание: требуется провести исследование алгоритмов одновременной локализации и построения карты при управлении мобильным роботом в складском помещении. Размеры складского мобильного робота: 0,5 x 0,5 x 0,5 м, размеры складского помещения: 20 x 20 м.

Исходные данные к работе:

1. Цифровая видеокамера: угол обзора 90°, разрешение не менее 1280 x 720 пикселей.
2. Лазерный дальномер: дальность сканирования до 10 м.
3. RGB-D камера: глубина измерения до 10 м, разрешение не менее 1280 x 720 пикселей.

Цель работы: проведение исследования алгоритмов одновременной локализации и построения карты при управлении мобильным роботом в складском помещении.

Для достижения цели работы необходимо решить следующие задачи:

1. Провести аналитический обзор существующих алгоритмов одновременной локализации и построения карты.
2. Выполнить создание модели мобильного робота и карты склада согласно требованиям

технического задания.

3. Провести численное моделирование исследуемых алгоритмов.
4. Выполнить анализ результатов моделирования, сравнение точности и быстродействия алгоритмов.

Содержание работы:

1. Аналитический обзор существующих решений
2. Подготовка среды моделирования для проведения исследования
3. Проведение численного моделирования алгоритмов
4. Заключение
5. Список литературы

**Форма представления материалов ВКР / Format(s) of thesis materials:**

Пояснительная записка ВКР

Презентация PowerPoint или PDF

**Дата выдачи задания / Assignment issued on:** 05.02.2024

**Срок представления готовой ВКР / Deadline for final edition of the thesis** 30.05.2024

**СОГЛАСОВАНО / AGREED:**

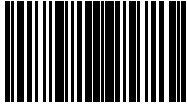
Руководитель ВКР/  
Thesis supervisor

Документ подписан	
Шаветов Сергей Васильевич	
20.02.2024	

(эл. подпись)

Шаветов Сергей  
Васильевич

Задание принял к  
исполнению/ Objectives  
assumed BY

Документ подписан	
Курчавый Виталий Вячеславович	
20.02.2024	

(эл. подпись)

Курчавый  
Виталий  
Вячеславович

Руководитель ОП/ Head  
of educational program

Документ подписан	
Бобцов Алексей Алексеевич	
06.05.2024	

(эл. подпись)

Бобцов Алексей  
Алексеевич

**Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО  
ITMO University**

**АННОТАЦИЯ  
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ  
SUMMARY OF A GRADUATION THESIS**

**Обучающийся / Student** Курчавый Виталий Вячеславович  
**Факультет/институт/кластер/ Faculty/Institute/Cluster** факультет систем управления и робототехники  
**Группа/Group** R3438  
**Направление подготовки/ Subject area** 15.03.06 Мехатроника и робототехника  
**Образовательная программа / Educational program** Робототехника 2020  
**Язык реализации ОП / Language of the educational program** Русский  
**Квалификация/ Degree level** Бакалавр  
**Тема ВКР/ Thesis topic** Исследование алгоритмов одновременной локализации и построения карты в задачах мобильной робототехники  
**Руководитель ВКР/ Thesis supervisor** Шаветов Сергей Васильевич, доцент, кандидат технических наук, Университет ИТМО, факультет систем управления и робототехники, доцент (квалификационная категория "ординарный доцент")

**ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ  
DESCRIPTION OF THE GRADUATION THESIS**

**Цель исследования / Research goal**

Проведение исследования алгоритмов одновременной локализации и построения карты при управлении мобильным роботом в складском помещении.

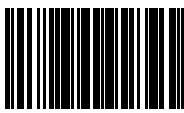
**Задачи, решаемые в ВКР / Research tasks**

1. Выполнение аналитического обзора алгоритмов одновременной локализации и построения карты в задачах мобильной робототехники. 2. Выполнение экспериментального исследования алгоритмов одновременной локализации и построения карты в задачах мобильной робототехники. 3. Выполнение анализа результатов экспериментального исследования.

**Краткая характеристика полученных результатов / Short summary of results/findings**

В результате выполнения работы были исследованы различные алгоритмы одновременной локализации и построения карты на основе сравнения результатов моделирования алгоритмов при управлении мобильным роботом в складском помещении.

Обучающийся/Student

Документ подписан	
Курчавый Виталий Вячеславович	
22.05.2024	

Курчавый  
Виталий  
Вячеславович

Руководитель ВКР/  
Thesis supervisor

(эл. подпись/ signature)

Документ подписан	
Шаветов Сергей Васильевич	
16.05.2024	

(эл. подпись/ signature)

(Фамилия И.О./ name  
and surname)

Шаветов Сергей  
Васильевич

(Фамилия И.О./ name  
and surname)

## СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ .....	7
ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	8
ВВЕДЕНИЕ.....	10
1 АНАЛИТИЧЕСКИЙ ОБЗОР АЛГОРИТМОВ ОДНОВРЕМЕННОЙ ЛОКАЛИЗАЦИИ И ПОСТРОЕНИЯ КАРТЫ.....	12
1.1 АЛГОРИТМЫ НА ОСНОВЕ КАМЕР .....	13
1.2 АЛГОРИТМЫ НА ОСНОВЕ ЛИДАРОВ.....	16
1.3 АЛГОРИТМЫ НА ОСНОВЕ КАМЕР ГЛУБИНЫ.....	19
2 ПОСТАНОВКА ЗАДАЧИ.....	24
3 ИССЛЕДОВАНИЕ КОНКРЕТНЫХ РЕАЛИЗАЦИЙ .....	25
3.1 GMAPPING.....	25
3.2 НЕСТОР SLAM.....	26
3.3 CARTOGRAPHER.....	27
3.4 ORB-SLAM2 .....	30
4 ОПИСАНИЕ МОДЕЛИРОВАНИЯ.....	33
4.1 ИСПОЛЬЗУЕМОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ .....	33
4.2 ОПИСАНИЕ ВЫБРАННЫХ ОБЪЕКТОВ ДЛЯ МОДЕЛИРОВАНИЯ..	35
4.3 ОПИСАНИЕ ПРОЦЕССА МОДЕЛИРОВАНИЯ.....	36
5 РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ.....	39
ЗАКЛЮЧЕНИЕ .....	40
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	41

## **СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ**

2D – Two-Dimensional

3D – Three-Dimensional

BoW – Bag of Words

CPU – Central Processing Unit

Фпс – Фреймы в секунду (Frames Per Second)

GPL – General Public License

GPS – Global Positioning System

GPU – Graphics Processing Unit

IMU – Inertial Measurement Unit

ORB – Oriented FAST and Rotated BRIEF

PTAM – Parallel Tracking and Mapping

RBPF – Rao-Blackwellized Particle Filter

ROS – Robot Operating System

SIFT – Scale-Invariant Feature Transform

SLAM – Simultaneous Localization and Mapping

SURF – Speeded-Up Robust Features

TSDF – Truncated Signed Distance Function

Кг – Килограм

Лидар – Light Detection and Ranging

Мм – Миллиметр

М/с – Метр в секунду

Об/мин – Обороты в минуту

Ч – час

Гб – гигабайт

## ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

DTAM (Dense Tracking and Mapping) – метод SLAM

Google – американская компания, специализирующаяся на интернет-сервисах и продуктах

Iterative Closest Point Algorithm (Итеративный алгоритм ближайших точек) – алгоритм для минимизации разницы между двумя наборами точек путём нахождения ближайших точек и итеративного преобразования одного набора к другому

LSD – Large-Scale Direct - метод визуальной одометрии

Microsoft – американская технологическая компания

Microsoft Kinect – камера глубины

MONO-SLAM (Monocular SLAM) – самый первый визуальный SLAM

ORB-SLAM (Oriented FAST and Rotated BRIEF SLAM) – метод SLAM

SVO (Semi-Direct Visual Odometry) – метод визуальной одометрии, который использует комбинацию прямых и непрямых подходов для отслеживания движения камеры

Визуальная одометрия – метод вычисления положения и ориентации камеры путём анализа последовательных изображений, полученных с камеры

Воксель – элемент трехмерного пространства, аналогичный пикселю в двумерной графике; используется для представления объектов или пространств в 3D

Карта местности – цифровое представление окружающей среды

Монокулярная система – система, использующая одну камеру для решения задач SLAM

Стереосистема – система, использующая две камеры для решения задач SLAM

Алгоритм Лукаса – Канаде – метод отслеживания оптического потока в изображениях



Фильтр Калмана – математический алгоритм для оценивания состояния динамической системы на основе последовательных шумных измерений

Фотометрическая ошибка – разница в интенсивностях пикселей между реальным изображением и его предсказанием в методах визуальной одометрии и SLAM

## **ВВЕДЕНИЕ**

В современной мире автоматизация играет ключевую роль в увеличении эффективности и оптимизации процессов. Благодаря замене ручного труда трудом роботов уменьшается время выполнения операций, повышается их качество, уменьшаются затраты на их выполнение.

В том числе и складские операции автоматизируются с помощью мобильных роботов. Роботы могут быстро и точно перемещать товары, ускоряя процессы перемещения товаров. Также мобильные роботы способствуют уменьшению операционных затрат, их не нужно обучать, им не нужно платить зарплату, они меньше ошибаются, быстро оптимизируются под изменившуюся среду. Автоматизация улучшает условия труда, так как сотрудники предприятия освобождаются от рутинной работы

Чтобы перемещаться внутри склада роботам необходимо точно определять свое местоположение, а также строить карту окружающей среды в реальном времени.

Цель данной дипломной работы заключается в исследовании различных алгоритмов, применяемых для одновременной локализации и построения карты на примере управления роботом в складском помещении.

В данной работе будут рассмотрены основные методы и подходы к решению задач одновременной локализации и построения карты, их точность и эффективность в условиях складской среды.

Исследование предполагает теоретический анализ основных алгоритмов одновременной локализации и построения карты. А также анализ практического применения в реальных условиях складской среды на основе моделирования. В работе будет произведено сравнение эффективности различных алгоритмов с учетом особенностей работы на складе, таких как наличие стеллажей, препятствия на пути движения робота.

В конце будут выявлены наиболее надежные и подходящие алгоритмы одновременной локализации и построения карты для мобильных роботов, эксплуатируемых на небольших складах.

# **1 АНАЛИТИЧЕСКИЙ ОБЗОР АЛГОРИТМОВ ОДНОВРЕМЕННОЙ ЛОКАЛИЗАЦИИ И ПОСТРОЕНИЯ КАРТЫ**

Изначально поставленная задача формулировалось для решения задачи автономного управления мобильным роботом [2]. Чтобы написать программу управления для автономного робота, нужно чтобы на вход программе подавались данные, к примеру показатели датчиков, на основе которых будет определяться входное воздействие. Часто робот запоминает карту местности, которая в дальнейшем используется для его навигации. Системы, реализующие алгоритм построения карты, можно выделить в два типа: работающие внутри помещения и снаружи. Если система строит карту вне помещения, то она может использовать GPS, но могут накладываться проблемы с освещением и шумом датчиков. В работе будут рассматриваться алгоритмы для помещений, так как целью является выявление наиболее подходящего алгоритма для складского робота. Система, реализующая алгоритм построения карты, принимает на вход показатели таких датчиков, как камеры, лидары, камер глубины и др., чтобы обрабатывать эти данные, оценивать положение робота и конструировать карту местности.

Часто алгоритмы классифицируются на основе датчиков, которые они используют для решения задачи. Таким образом, выделяют алгоритмы, которые используют камеры в качестве датчиков, алгоритмы, которые используют лазерный сканер или камеру глубины. Алгоритмы, которые используют камеры, в свою очередь делятся на стерео, которые используют две камеры, которые часто располагаются подобно глазам человека, и монокулярные, которые используют одну камеру.

Формально, исследователи разделяют задачу построения карты и локализации на 5 следующих этапов:

- Инициализация
- Слежение
- Переопределение

- Глобальное построение карты [1].

Различные алгоритмы используют разную методологию на каждом этапе. От выбранной методологии зависит эффективность и сложность алгоритма. В данной работе будут рассмотрены общие методологии и подходы для большинства современных алгоритмов.

Под картой местности чаще всего понимается карта занятости: ячейки карты могут быть двух типов, белые, означающие свободную клетку и черные, означающие занятую клетку, к примеру стеной или стеллажом.

### **1.1 Алгоритмы на основе камер**

Рассмотрим алгоритмы, использующие камеры. Камеры обязательно должны быть откалиброваны, чтобы все параметры были известны. На этапе инициализации определяется глобальная система координат для того, чтобы в будущем оценивать положения камеры. На этапе слежения используются соседние фреймы камеры и делается вывод о перемещении робота, далее имеющаяся карта сравнивается с видимой на данный момент, чтобы оценить положение робота, на основе метода оценки с помощью  $n$  точек [3]. В некоторых алгоритмах также происходит переопределение позиции, в случае если слежение не удалось, данный этап является преимущественным, так как делают локализацию более эффективной, к примеру решает проблему с похищением робота [2]. На следующем этапе происходит оптимизация глобальной карты. Выявляются циклы, чтобы измерить ошибку оценки и преобразовать соответствующим образом карту.

Есть два подхода к задаче слежения: метод на основе признаков и прямой метод. Первый метод определяет определенные признак на фотографиях в разные моменты времени и на основе этого определяет местоположение робота, минимизируя геометрическую ошибку. Он очень энергозатратный, часто невозможно выполнить в реальном времени, а также теряется информация, так как используются только определенные части фреймов. Ещё

один минус этого алгоритма заключается в том, что камера может попасть в точку, из которой не видно ни одного ориентира. Второй метод минимизирует фотометрическую ошибку и исключает недостатки первого алгоритма. Но имеет свои. Главные из них: он использует метод градиентного спуска, из-за чего есть вероятность найти локальный минимум, а не глобальный, требует большого количество пикселей для расчета, предполагается постоянный показатель светимости.

Самый ранний алгоритм, использующий одну камеру и строящий карту местности – MonoSLAM, придуманный в 2007 году. В нем используется расширенный фильтр Калмана для оптимизации. Можно использовать ограниченное количество ориентиров. Самый главный недостаток этого алгоритма — это низкая производительность [5].

В алгоритме PTAM производительность усовершенствовалось путем разделения вычисления алгоритма слежения и алгоритма построения в отдельные потоки на CPU. Также вместо фильтра Калмана использовались методы нелинейной оптимизации. PTAM – это первый алгоритм, который использует регулировку связки [6]. Инициализация происходит с помощью алгоритма пяти точек [7]. Алгоритм считает фотографию, ключевым фреймом, если фотография сильно отличается от остальных ключевых фреймов. На основе этих ключевых фреймов строится слежение за позицией робота. Ключевые фреймы оцениваются с помощью триангуляции и регулировки связки. Ключевое преимущество PTAM перед MonoSLAM заключается в том, что в первом алгоритме может использоваться большое количество точек, что способствует повышению точности.

ORB-SLAM является расширенным PTAM, помимо поиска цикла в этом алгоритме реализована графовая оптимизация, что еще более сильно повышает точность, также, как и в PTAM есть регулировка связки. Инициализация карты может происходить за счет начального движения, если используется монокулярная система и на основе данных с двух камер, если

используются стерео система. В качестве признаков использует бинарные дескрипторы, которые могут быть быстро вычислены. Алгоритм минимизирует функцию потерь Хьюбера. Также для детекции цикла используется метод мешок слов [8, 22].

Примером алгоритма, который не использует метода поиска признаков, а напрямую работает с пикселями: DTAM. В данном алгоритме инициализация карты происходит на основе данных с двух камер. Расстояние до каждого пикселя также измеряется на основе данных с камер. Также использует графовую оптимизацию. Преимуществом использования данного алгоритма является возможность отсутствия на карте существенных признаков, что позволяет адаптироваться под любую среду. Обладает высокими показателями точности, но требует большого количества памяти. Также, периодически возникают сложности с инициализацией карты, если использовать монокулярную систему. Алгоритм является сильно зависимым от освещенности, как и большинство алгоритмов, используемых камеры.

Еще один алгоритм, который обходится без детекции признаков – это LSD-SLAM. В основе этого алгоритма лежит полу плотная визуальная одометрия [9]. Также присутствует графовая оптимизация с семью степенями свободы. Сначала каждому пикселю присваивают случайное значение глубины. Движение робота оценивается с помощью вида, сгенерированного сконструированной картой. Как и DTAM LSD-SLAM пытается минимизировать фотометрическую ошибку. Для минимизации используется алгоритм Гаусса-Ньютона. Вместо того, чтобы оценивать каждый пиксель алгоритм концентрируется на местах высоким градиентом, к примеру на углах и текстурах. Как и в других алгоритмах с графовой оптимизацией мир представляется некоторым количеством ключевых фреймов [10].

SVO – полупрямой метод визуальной одометрии. Отслеживание траектории выполняется с помощью метода выделения признаков, а глобальное построение карты с помощью прямого метода. В основе лежит

алгоритм Лукаса – Канаде. Движение камеры оценивается на основе минимизации фотометрической ошибки вблизи признаков. Таким образом SVO комбинирует в себе два подхода. За счет этого SVO является более робастным, чем алгоритмы, где слежение и построение карты решаются только с помощью метода поиска признаков. Поскольку расчет ведется только для пикселей, около признака, то данный алгоритм является более эффективным по времени и памяти, чем алгоритмы, основанные только на прямых методах, к примеру DTAM и LSD-SLAM [12].

Следующий алгоритм – DSO. В переводе означает прямая разряженная одометрия. Данный алгоритм минимизирует фотометрическую и геометрическую ошибки. С помощью алгоритма Лукаса – Канаде подобно SVO. Также в алгоритме реализовано отслеживание цикла [13]. Однако DSO – это визуальная одометрия, а не метод построения карты местности и локализации, так как алгоритм строит только локальную карту. Методы построения карты с прямым расчетом могут классифицироваться в зависимости от карты плотности, так SVO и DSO разряженную плотность, LSD-SLAM умеренную, а DTAM – плотную [1, 12, 13].

## **1.2 Алгоритмы на основе лидаров**

Slam использующие лидары - это технология, которая измеряет расстояния путем освещения цели лазерным лучом и затем анализирует отраженный лазер. Эта технология предоставляет высокоразрешённые 3D данные, что делает её особенно подходящей для задач SLAM.

Опишем подробнее работу лидара. Он излучает лазерные лучи и измеряет время, которое требуется свету, чтобы отразиться от объектов. Рассчитывая время пролёта, сенсор определяет расстояние до каждой точки в окружающей среде, создавая облако точек — 3D представление окружающего пространства. Обработка сырых данных облака точек необходима для точного SLAM. Ключевые шаги обработки облака точек включают:



- понижение выборки: уменьшение количества точек в облаке для повышения эффективности вычислений при сохранении важных особенностей.

- фильтрация шума: удаление ошибочных точек, вызванных шумом датчиков или внешними условиями.

- сегментация: идентификация и разделение различных объектов или областей внутри облака точек.

Из облака точек извлекаются признаки для идентификации ключевых точек или областей, которые можно отслеживать на протяжении нескольких сканов. Эти признаки могут включать геометрические формы, края или определённые паттерны внутри облака точек. Различают следующие признаки: плоские поверхности внутри облака точек, резких края или углы, которые являются характерными и стабильными. Сопоставление признаков между последовательными сканами позволяет системе оценивать относительное движение мобильного робота.

Оценка одометрии включает вычисление приращения движения лидара между сканами. Она производится с помощью следующих методов. Первый – это широко используемый итеративный алгоритм ближайшей точки, который, который минимизирует расстояние между соответствующими точками в последовательных облаках точек для оценки преобразования (линейного перемещения и вращения) между точками. Также используется алгоритм нормального распределения трансформаций, который представляет из себя вероятностный подход, использующий сеточное представление облака точек, и сопоставляет эти сетки между сканами для оценки движения [41]. Поиск цикла критичен для исправления накопленной ошибки при построении карты. Когда система повторно посещает ранее построенную область, она обнаруживает цикл путем сопоставления текущего облака точек с сохранёнными точками. Техники детекции циклов включают в себя два метода. Первый метод – это глобальная регистрация, который сравнивает текущее

облако точек с глобальной картой для идентификации совпадений. Вторым методом – это представление карты в виде графа позиций. Обнаружение цикла добавляет новые рёбра в граф, и используются техники оптимизации для минимизации ошибок [43]. Часто используются следующие методы оптимизации графа:

- Метод Гаусса – Ньютона
- Алгоритм Левенберга – Марквардта
- Ceres solver [49]
- Обычная графовая оптимизация [50]
- Постепенное увеличение плавности и построения карты [42]
- Разные виды многоуровневой оптимизации
- Функция потерь Хьюбера
- Функция потерь Коши

Основными преимуществами лидара являются высокая точность, сенсоры предоставляют точные измерения расстояний, что приводит к высоко точным картам и локализации, надёжность, эффективность в различных условиях окружающей среды, включая слабое освещение или плохую видимость, где другие сенсоры, например такие, как камеры, могут не справляться. Также системы, основанные на лидаре формируют детализированные 3D представления окружающей среды, полезные для приложений, требующих информации о глубине. Основным минусом таких систем является их высокая стоимость, а также, нестабильная работа в условиях дождя и тумана [44]. На Рисунок 1 приведена примерная схема работы алгоритмов с лидаром.

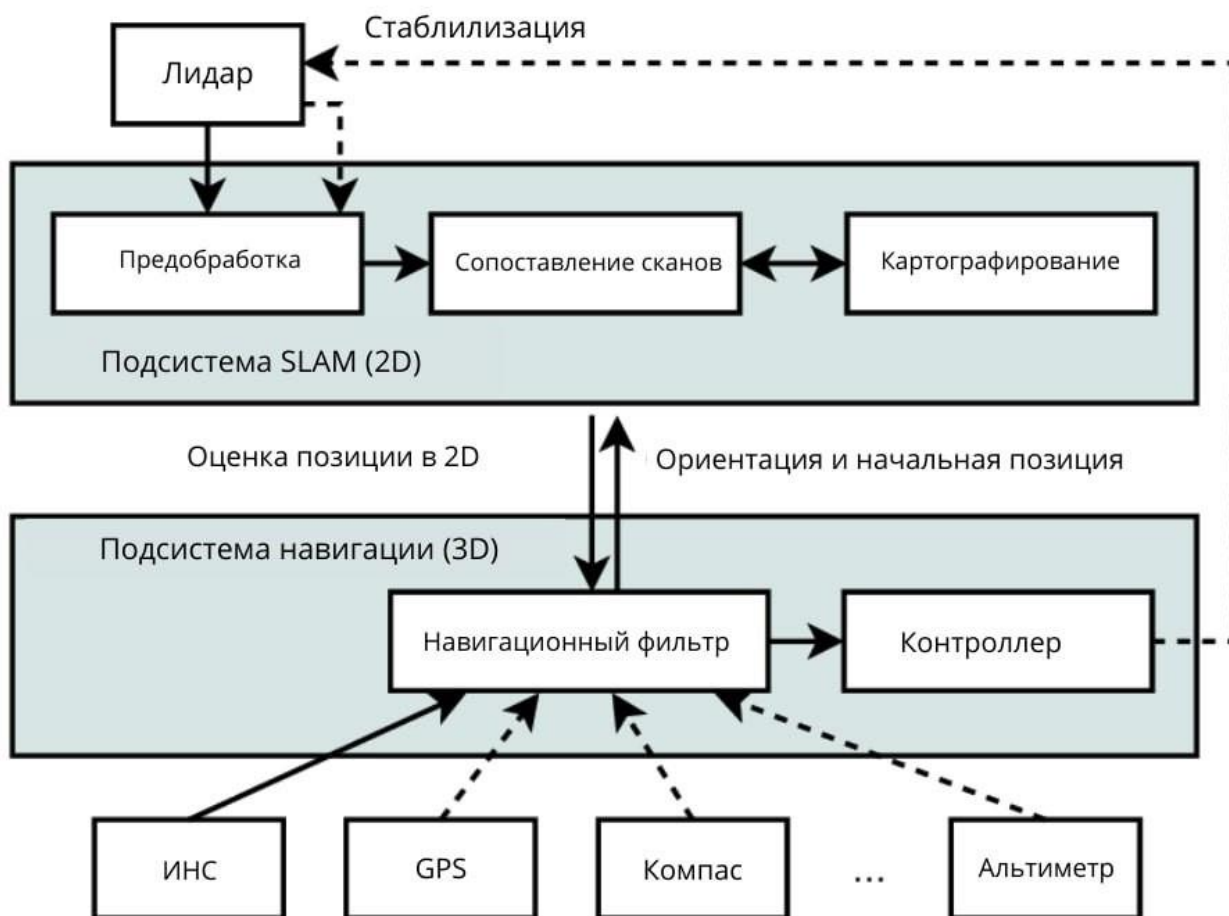


Рисунок 1 — Схема работы SLAM с лидаром

### 1.3 Алгоритмы на основе камер глубины

За последнюю декаду камеры глубины получили широкое использование в алгоритмах построения карты. А также они стали гораздо дешевле, что повышает частоту использования, к примеру Microsoft Kinect [14]. Большинство современных алгоритмов, использующих камеру глубины и вообще камеры основаны на архитектуре PTAM, где задачи отслеживания и локального построения карты решаются параллельно.

Алгоритмы на основе камер глубины гораздо проще, чем на основе моно и стерео камер, так как на вход алгоритму, помимо фотографии подается еще и карта глубины (расстояние до каждого пикселя на фотографии), а не только цветная фотография. Что помогает избежать многих манипуляций по расчету

расстоянию в стерео и моно системах, в результате чего уменьшается вероятность ошибки и повышается эффективность и робастность алгоритма. Алгоритмы на основе камер глубины так же, как и на основе обычных камер, могут быть разделены на три типа на основе методов отслеживания положения робота: прямой метод, непрямой (на основе признаков) метод, гибридный метод, где используются оба метода.

Для начала рассмотрим прямые методы.

Kinect Fusion – алгоритм, строящий 3D карту местности, предложенный исследователям из Microsoft [15]. Вначале алгоритм получает данные из камеры глубины, которые получены с камеры, использующей структурированный свет или время полета технологии для оценки расстояния. Пространство сохраняется в качестве карты вокселей – 3D пикселей в формате TSDF. Позиция робота оценивается с помощью итеративного алгоритма ближайших точек. В дальнейшем TSDF информация используется для построения карты [16, 17]. Kinect Fusion реализована на GPU, чтобы добиться развертывания алгоритма в реальном времени [1].

Следующий рассматриваемый алгоритм – RGBDSLAM. Обладает высокой точностью и робастностью. Основанный на минимизации функции потерь, названной RGBDTAM. Функция потерь включает в себя фотометрическую ошибку в точках, которые принадлежат углам Кэнни, которые детектируются с помощью оператора Кэнни [18]. Информация о карте также хранится в формате TSDF.

IDRGBO – алгоритм построения карты, основанный на методе прямого отслеживания, который не исследует как можно большее количество пикселей, а берет только 24 точки. При этом по точности не уступает другим прямым алгоритмам. Основываясь на матрице информации, алгоритм выбирает точки, которые максимизирует детерминант матрицы и на основе их оценивает положение робота. Также использует итеративный алгоритм ближайших точек. Умеет распознавать циклы. Преимуществами алгоритма являются

высокая точность, робастность, детализированная карта и успешная распознаваемость циклов. Минусы: высокая нагрузка на CPU и большое потребление объема памяти [19].

Рассмотри теперь не прямые методы, которые сначала извлекают признаки из фотографий, а затем минимизируют геометрическую ошибку. Обычно используется три типа геометрических ошибок: 2D от точки к точке, 3D от точки к точке, 3D от точки к плоскости [20].

RGB-D SLAM – алгоритм предложенный и протестированный в 1012 году. Сначала извлекаются ключевые точки и дескрипторы (например, SIFT, SURF, ORB), извлекаемые из цветных изображений для визуального сопоставления. Затем используется глубинная информация для улучшения сопоставления признаков и разрешения неоднозначностей. Оценка местоположения производится с помощью итеративного алгоритма ближайших точек. Минимизируется 3D ошибка от точки к точке. Затем строится карта местности, которая представляет из себя воксельную сетку. После использования визуальные и геометрические признаки для обнаружения повторных посещений ранее областей (поиск цикла). Затем производится коррекция накопленных ошибок путем оптимизации графа с помощью нелинейной функции и учетом циклов, обеспечивающая глобальную согласованность [21].

Следующий алгоритм - ORB-SLAM2. В позапрошлом пункте рассматривались монокулярный и стерео режимы работы. Сейчас будет рассматриваться режим работы, использующий камеры глубины. Как и в остальных двух режимах использует ORB ключевые точки и дескрипторы, как 2D признаки, которые извлекаются из цветных изображений. ORB признаки обеспечивают устойчивость к изменениям масштаба, поворота и освещения. Глубинные данные, полученные от камеры, используются для добавления 3D информации к извлечённым признакам. Затем производится сопоставление ORB признаков между последовательными кадрами для оценки

относительного движения камеры. В алгоритме минимизируется 2D ошибка от точки к точке. Также используются расстояния до пикселей, для улучшения точности. Далее строится детализированная карта местности. Алгоритм использует метод мешок слов для распознавания циклов, используя цветовые признаки [8]. В случае, если детектируется ошибка, используется ранее сконструированная карта. Используется графовая оптимизация, после которой производится полная регулировка связки в разных потоках, для повышения производительности. Главными преимуществами системы является высокая точность и устойчивость в случае малого количества текстур, где монокулярные системы работают ненадежно. Также на выходе предоставляет детализированную карту. Главные недостатки: камеры глубины могут работать некорректно с отражающимися или прозрачными поверхностями, высокие затраты на вычисления [22, 23].

Теперь рассмотрим гибридные методы. Они минимизируют как геометрическую, так и фотометрическую ошибки. В сравнении с непрямыми методами, прямые выигрывают по производительности, так как не требуют извлечение признаков, которое требует большое количество вычислений. Прямые методы являются более робастными в условиях недостатка текстур. Но они не устойчивы к движущимся объектам. Гибридные методы реализуют преимущества обоих методов. Функция потерь представляет собой сумма геометрической и фотометрической ошибок.

Для начала рассмотрим CPA-SLAM [25], разработанный в 2016 году. Геометрическая ошибка представлена в двух видах: от точки к плоскости и от плоскости к плоскости. В этом методе, за исключением прямого выравнивания изображения, строится глобальная плоскостная модель, которая используется для оценки положения робота. Ключевым преимуществом работы является то, что дополнительные выравнивание делает возможным использование изображения доступной для ключевых фреймов для проведения точного слежения за позицией робота, а глобальная модель помогает избегать дрейфа.

KDP-SLAM – алгоритм, который объединяет в себе итеративный метод спроецированной плоскости (как геометрическую составляющую) и одометрию, использующую пирамидальную плотность (как фотометрическую составляющую) [28].

Следующий алгоритм – ElasticFusion, разработанные в 2015 году. Ключевой особенностью является то, что он использует детекцию циклов, как локальных, так и глобальных. Предоставляет карту, как набор элементов поверхностей. Использует метод, предложенный в статье 2013 года [24] для преобразования и инициализации элементов поверхностей. Делит элементы на активные и неактивные. На основе этих данных строится карта местности. Не использует графов [26].

Алгоритмы на основе камер глубины представляют собой мощный инструмент для решения задачи локализации и построения карты местности. Они создают детализированные трехмерные модели окружающей среды в реальном времени [27].

## 2 ПОСТАНОВКА ЗАДАЧИ

Задача построения карты и локализации на языке математики формулируется следующим образом:

Данные:

$x_{1:T} = \{x_1, x_2, \dots, x_T\}$  – последовательность случайных величин, описывающая траекторию движения робота за время от 1 до  $T$ .

$z_{1:T} = \{z_1, z_2, \dots, z_T\}$  – последовательность случайных величин, описывающая показания датчиков робота за время от 1 до  $T$ .

$u_{1:T} = \{u_1, u_2, \dots, u_T\}$  – последовательность случайных величин, описывающая входное воздействие за время от 1 до  $T$ .

$x_0$  – начальное положение робота.

$M$  – карта местности.

Решение задачи SLAM сводится к оценке апостериорного распределения, из-за наличия шума в измерениях:

$$p(x_{1:T}, M \mid z_{1:T}, u_{1:T}, x_0)$$

Чаще всего  $x_i, u_i \in SE(2)$  или  $SE(3)$ . Где  $SE(2)$ ,  $SE(3)$  – специальные евклидовы группы, которые задают трансформацию в 2D и 3D соответственно.



### 3 ИССЛЕДОВАНИЕ КОНКРЕТНЫХ РЕАЛИЗАЦИЙ

В данном пункте будут подробно описаны наиболее распространенные алгоритмы:

- Gmapping
- HectorSLAM
- Cartographer
- RTAB-Map

#### 3.1 GMapping

GMapping - это один из самых известных и широко используемых алгоритмов для решения задачи построения карты и локализации в робототехнике. Он был разработан Джорджио Гризетти, Сириллом Штахниссом и Вольфрамом Бургардом. GMapping основан на использовании вероятностных методов, в частности, Rao-Blackwellized Particle Filter (RBPF), и предназначен для построения карты окружающей среды и одновременной локализации робота в этой среде [45]. А также в основе GMapping лежит алгоритм FastSLAM, который был разработан для решения проблем традиционных методов SLAM, сталкивающихся с высокой размерностью задачи. Этот алгоритм используют ориентиры для решения задачи. GMapping стал популярным выбором для решения задачи картографирования поскольку не требует огромных затрат на вычисления и при этом получает довольно точные карты.

GMapping сочетает в себе несколько алгоритмов и концепций. Вот ключевые компоненты и алгоритмы, которые он реализует: фильтра частиц и алгоритм FastSLAM. Фильтры частиц, также известные как локализация Монте-Карло, представляют собой вероятностный подход, используемый для оценки положения робота. В GMapping фильтры частиц используются для представления возможных местоположений и ориентаций робота в окружающей среде. Каждая частица представляет собой возможную гипотезу о состоянии робота. Важной особенностью Gmapping является использование

фильтров частиц Рао-Блэкуэлла. Фильтры частиц Рао-Блэкуэлла (RBPF) — это техника, используемая для повышения эффективности фильтров частиц, в следующей статье подробно описаны преимущества этой техники [29]. RBPF позволяет разделить оценку позиции робота и карты. Это приводит к более эффективным вычислениям.

GMapping использует сеточные карты занятости для представления окружающей среды. Сеточная карта занятости делит окружающую среду на сетку ячеек, каждая из которых может быть занята, свободна или неизвестна. Это сеточное представление хорошо подходит для навигации роботов и особенно полезно для сред с препятствиями и свободными пространствами.

Алгоритм GMapping является точным и устойчивым к шумам, часто возникающих в реальном мире, благодаря фильтрам частиц. Но в больших помещениях и масштабах требует значительных вычислений [30, 45].

### **3.2 Hector SLAM**

Hector SLAM – алгоритм сейчас широко используется в приложениях робототехники. Этот алгоритм выигрывает в точности и надежности у многих алгоритмов. В отличие от многих SLAM-алгоритмов, Hector SLAM не зависит от одометрии или инерциальных измерений, что делает его весьма подходящим для платформ, не оснащённых этими датчиками. Hector SLAM использует несколько ключевых концепций и алгоритмов для достижения эффективного и точного картографирования и локализации: совмещение сканов, сеточную карту занятости, мультиразрешающиеся сетки, метод Гаусса – Ньютона для оптимизации. Теперь напишем подробнее об каждом из аспектов.

Совмещение сканов выравнивает последовательные сканы, сделанные лидаром, с текущей картой, чтобы оценить положение робота. Это достигается путём нахождения трансформации, которая включает в себя смещение и вращение, которая лучше всего совмещает новый скан с картой. Hector SLAM

использует итеративный алгоритм ближайших точек, который минимизирует разницу между сканированными точками и картой. [35]

Для улучшения производительности и точности Hector SLAM использует мультиразрешающиеся сетки. Этот подход поддерживает несколько карт с разными разрешениями. Карты с высоким разрешением предоставляют детальную информацию для точного совмещения сканов, в то время как карты с низким разрешением позволяют более быструю глобальную локализацию и обрабатываются для больших территорий.

Для сканирующего совмещения Hector SLAM применяет метод оптимизации Гаусса – Ньютона. Этот метод итеративно корректирует оценку позиции робота для минимизации ошибки между сканом лидара и картой. Процесс оптимизации данным методом обеспечивает максимально точное совмещение скана с картой, что ведёт к точной локализации и к точному картографированию.

Hector SLAM является мощным и эффективным алгоритмом SLAM, хорошо подходящим для приложений в реальном времени в мобильной робототехнике и навигации БПЛА. Он обеспечивает высокую точность без необходимости в одометрии. Несмотря на некоторые ограничения, особенно в сильно динамичных или 3D средах, Hector SLAM остаётся универсальным и широко используемым инструментом в области автономных систем [31, 32, 46, 46].

### **3.3 Cartographer**

Cartographer — это передовая библиотека SLAM с открытым исходным кодом, разработанная в Google. Она предназначена для решения задачи 2D и 3D построения карты и одновременной локализации. Этот алгоритм используется в различных робототехнических приложениях, включая мобильных роботов и автономные транспортные средства. Cartographer известен своей высокой точностью и надёжной работой в сложных условиях,

благодаря использованию сложных алгоритмов и методов слияния данных от датчиков.

Cartographer включает себя несколько ключевых методов, алгоритмов и этапов для решения задачи: графовая оптимизация, слияние данных с датчиков, построение подкарт, обнаружение циклов. В основе Cartographer лежит графовая оптимизация позиций. Граф позиций представляет собой граф, где узлы обозначают позиции робота в разные моменты времени, а рёбра представляют пространственные ограничения между этими позициями. Процесс оптимизации включает корректировку траектории робота для минимизации ошибки между наблюдаемыми данными сенсоров и уже оценёнными данными. Этот метод повышает глобальную согласованность карты и точность локализации. Cartographer использует слияние данных датчиков для объединения данных от нескольких датчиков, таких как лидар и IMU (инерциальные измерительные устройства). Слияние данных сенсоров улучшает надёжность и точность оценки позы, используя преимущества различных сенсоров. Например, лидар предоставляет точные измерения расстояний, а IMU обеспечивает данные о положении робота с высокой частотой [35].

Cartographer делит окружающую среду на меньшие регионы, называемые подкартами. Каждая подкарта представляет собой локальную карту, охватывающую ограниченную область и создаваемую с использованием последовательных сканов лидара. Подкарты позволяют эффективно управлять картой и обновлять её, так как изменения в окружающей среде могут быть локализованы в конкретных регионах без влияния на всю карту. Детекция циклов – это важный этап в алгоритме Cartographer. Он определяет, когда робот снова попадает в ранее посещенную область, и корректирует накопленную ошибку в траектории робота. С помощью этого Cartographer может выравнивать карту, обеспечивая глобальную согласованность и снижая

ошибки, накапливаемые со временем. Подробная схема работы приведена на Рисунок 2.

Описываемый алгоритм обеспечивает точную карту и локализацию в сложных условиях. Часто используются в реальном времени, обрабатывает данные с датчиков и обновляет карту занятости. Основными недостатками Cartographer является то, что он требует больших затрат на вычисления, не эффективен в условиях динамической среды, использует IMU, зависит от калибровки датчиков [33, 34, 47].

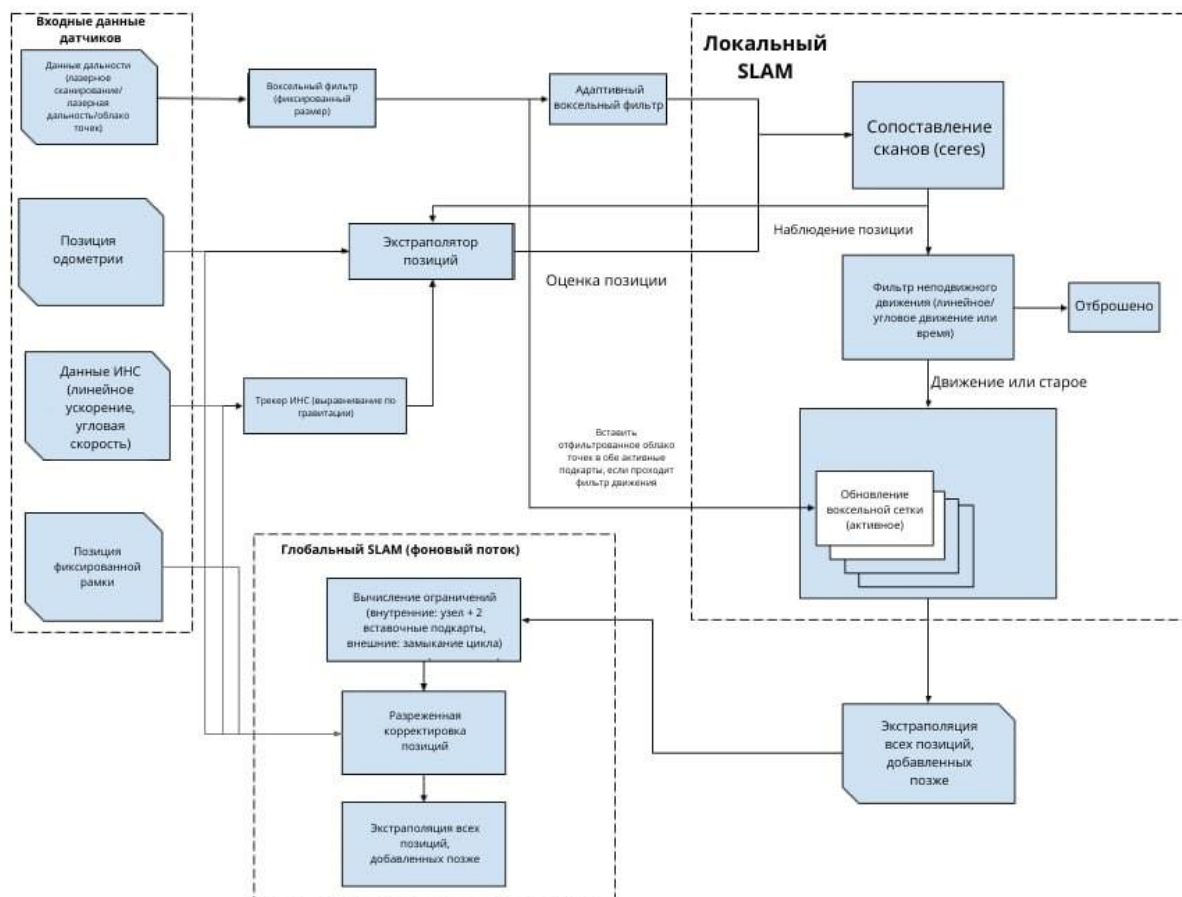


Рисунок 2 — Схема работы Cartographer

### 3.4 ORB-SLAM2

ORB-SLAM2 – это система SLAM (одновременная локализация и картографирование) с открытым исходным кодом, основанная на методе извлечения признаков, которая отличается высокой производительностью в реальном времени и надёжностью. Разработанная Раулем Мур-Арталь, Хуаном Д. Тардосом и Х. М. М. Монтьелем. Является расширенной версией ORB-SLAM, добавляя поддержку для моно, стерео и RGB-D камер. Она широко известна своей точностью, эффективностью и способностью работать в различных и сложных условиях. В основе ORB-SLAM2 лежат признаки ORB (Oriented FAST and Rotated BRIEF):

- FAST (Features from Accelerated Segment Test): метод обнаружения углов, который быстро идентифицирует ключевые точки на изображении.

- BRIEF (Binary Robust Independent Elementary Features): дескриптор, который кодирует внешний вид ключевых точек с помощью бинарных строк, что позволяет быстро их сопоставлять.

Признаки ORB объединяют эти методы с информацией об ориентации, чтобы создать дескрипторы, устойчивые к поворотам и изменениям точки зрения и освещённости. Для моно системы инициализация является критическим этапом. ORB-SLAM2 использует метод, основанный на гомографии [36], или метод, основанный на фундаментальной матрице [37], в зависимости от структуры сцены (плоская или общая). С помощью этих методов устанавливается начальную карту и оценку позы. Позиция робота уточняется с помощью регулировки связки, которая уменьшает ошибку расхождения. Отслеживание отвечает за оценку позиции камеры в реальном времени.

ORB-SLAM2 использует комбинацию методов для обеспечения надёжного отслеживания: модель движения, которая предсказывает позицию текущего кадра на основе предыдущих позиций и данных камер, предоставляя

начальную оценку. Также в алгоритме реализовано переопределение, когда отслеживание теряется, ORB-SLAM2 использует механизм переопределения, который сопоставляет признаки текущего кадра с ранее наблюдаемыми ключевыми кадрами, чтобы восстановить позицию камеры. Местоположение робота уточняется с помощью пакетной настройки, минимизируя ошибку повторного проецирования наблюдаемых признаков.

Локальное картографирование включает построение и уточнение локальной карты. Основные компоненты включают:

- Вставка ключевых фреймов: Значимые фреймы добавляются в качестве в карту, когда исследуется новая область или когда камера быстро движется.

- Локальная пакетная настройка: оптимизирует позиции ключевых фреймов и 3D-позиции точек карты в локальном окне для минимизации ошибок повторного проецирования.

- Отбраковка: удаляет избыточные ключевые фреймы и точки карты, чтобы поддерживать структуру карты.

ORB-SLAM2 использует подход BoW для обнаружения циклов, сравнивая дескрипторы ORB с визуальным словарём [8]. Когда цикл обнаруживается, строится граф позиций, где узлы представляют позиции ключевых фреймов, а рёбра представляют пространственные ограничения. Граф оптимизируется для минимизации ошибок, корректируя накопившийся дрейф и выравнивая карту.

ORB-SLAM2 обеспечивает точную локализацию и картографирование. Также система может работать в реальном времени, ORB-SLAM2 обрабатывает данные датчиков и обновляет карту с минимальной задержкой. Надёжное обнаружение циклов и последующая корректировка траектории и карты обеспечивают глобальную согласованность карты. Одним из главных плюсов является то, что алгоритм поддерживает моно, стерео и RGB-D камеры, что делает её адаптивной к различным установкам. Является

доступна под лицензией GPL, что позволяет настраивать и интегрировать систему в различные проекты. Однако алгоритм требует высоких вычислительных затрат. Плохо работает в условиях, неплотных признаков или в местностях, где часто повторяются шаблоны. Также ухудшается производительность в условиях динамической среды [38, 39, 40, 48].



## 4 ОПИСАНИЕ МОДЕЛИРОВАНИЯ

### 4.1 Используемое программное обеспечение

ROS – это известная платформа, предоставляющая широкий спектр для упрощения процесса написания программ, которые управляют роботами. ROS предоставляет набор библиотек и соглашений, которые позволяют описывать поведение роботов. Позволяет изолировать программы для отдельных частей робота, к примеру датчиков, контроллеров, алгоритмов построения карты, описывает взаимодействия между изолированными системами, с помощью узлов, топиков, действий и сервисов, а также системы подписчиков издателей. Широко используется, как в академических целях, так и в промышленности.

Gazebo – это мощный инструмент для симуляции роботов, предоставляющий 3D визуализацию моделирования. Он обеспечивает реалистичные сред и точную физику для тестирования и разработки алгоритмов робототехники. Интегрирована в библиотеку ROS. Демонстрирует поведение взаимодействия робота с внешним миром. Часто используется в академических и промышленных целях.

Python, C++ – два основных языка программирования, используемых в ROS. Python часто используется для написания приложений под ROS из-за своей доступности, читаемости и огромного количества библиотек, что делает его идеальным для написания скриптов. Однако, C++ обеспечивает превосходную производительность, точное управление ресурсами системы и возможности многопоточности, что делает его подходящим для вычислительно интенсивных задач, обработки данных в реальном времени и взаимодействия с аппаратным обеспечением на низком уровне в робототехнических приложениях на основе ROS. Оба языка играют важную роль в разработке ROS, с Python часто используется для быстрого прототипирования и разработки, а C++ применяется для компонентов и модулей, требующих высокой производительности. Их совместное использование позволяет разработчикам использовать преимущества каждого

языка для создания эффективных, гибких и надежных робототехнических систем.

Turtlebot3 – это набор стандартных моделей роботов, для исследования, обучения и тестирования различных гипотез. Созданный компанией ROBOTIS в сотрудничестве. Он построен на основе ROS. Использование Turtlebot3 упрощает взаимодействие между системами SLAM и роботом, позволяет сконцентрироваться только на конкретно изучаемой задаче, предоставляя уже готовую модель робота и модели датчиков. TurtleBot3 выпускается в различных моделях, таких как Burger, Waffle и Waffle Pi, каждая из которых предлагает разные возможности и датчики для выполнения задач от простой навигации до более сложных автономных операций. Его открытый исходный код оставляет широкое окно для настройки робота.

GMapping – реализация одноименного алгоритма SLAM для ROS [45].

Hector slam – реализация одноименного алгоритма SLAM для ROS [46].

ORB-SLAM2 – реализация одноименного алгоритма SLAM [48].

Cartographer – реализация одноименного алгоритма SLAM для ROS [47].

## 4.2 Описание выбранных объектов для моделирования

В качестве робота модели робота выбран waffle\_pi, так как он оснащен и лидаром, и камерой. В Таблица 1 приведены характеристики модели. В Таблица 2 приведены показатели лидара. В Таблица 3 приведены показатели камеры.

Элемент	Характеристика элемента
Максимальная линейная скорость	$0.26 \frac{\text{м}}{\text{с}}$
Максимальная угловая скорость	$1.82 \frac{\text{рад}}{\text{с}}$
Максимальная нагрузка	30 кг
Размер	281 мм × 306 мм × 141 мм
Масса	1.8 кг
Ожидаемое время работы	2 ч
Ожидаемое время зарядки	2 ч 30 мин
Лидар	360 Laser Distance Sensor LDS – 01
Камера	Raspberry Pi Camera Module v2.1
IMU	Gyroscope 3 Axis Accelerometer 3 Axis

Таблица 1 — Показатели модели робота

Элемент	Характеристика элемента
Диапазон расстояний	120 мм ~ 3500 мм
Точность расстояний(120 мм ~ 499 мм)	$\pm 15 \text{ мм}$
Точность расстояний(500 мм ~ 3500 мм)	$\pm 5.0\%$
Частота сканирования	$300 \pm 10 \frac{\text{об}}{\text{мин}}$
Угловой диапазон	$360^\circ$
Угловое разрешение	$1^\circ$

Таблица 2 — Показатели лидара

Элемент	Характеристики элемента
Разрешение	1920 × 1080
Соотношения сторон	16:9
Частота кадров	0.1 – 30 фпс

Таблица 3 — Показатели камеры

### 4.3 Описание процесса моделирования

Ход эксперимента:

- Создание карты помещения
- Создание реальной карты занятости помещения
- Проведение одного запуска программы для каждого из алгоритмов
- Сохранение карты и траектории движения робота
- Подсчет метрик

Карта помещения представлена на Рисунок 4. Стены могут означать, как реальные стены, так и места для стеллажей. Реальная карта занятости окружения представлена на Рисунок 3.

Конфигурация системы указана в Таблица 4.

CPU	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
Операционная система	Ubuntu 20.04.6 LTS
Оперативная память	8 Гб
Тип операционной системы	64-bit

Таблица 4 — конфигурация системы



Рисунок 3 — Реальная карта занятости

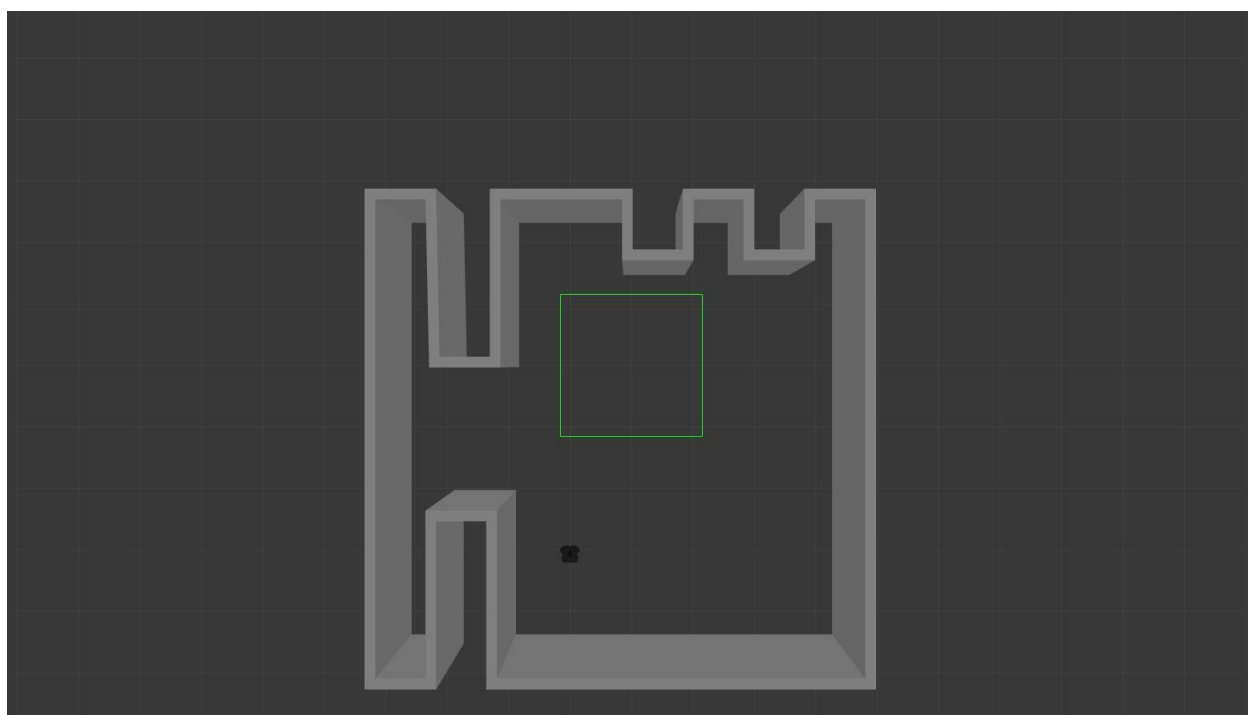


Рисунок 4 — Карта окружения

Далее продемонстрируем карты, построенные с помощью алгоритмов на Рисунок 5 и на Рисунок 6.

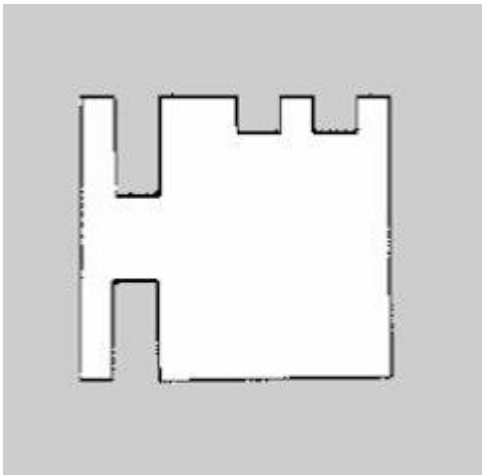


Рисунок 5 — Карта на основе  
GMapping

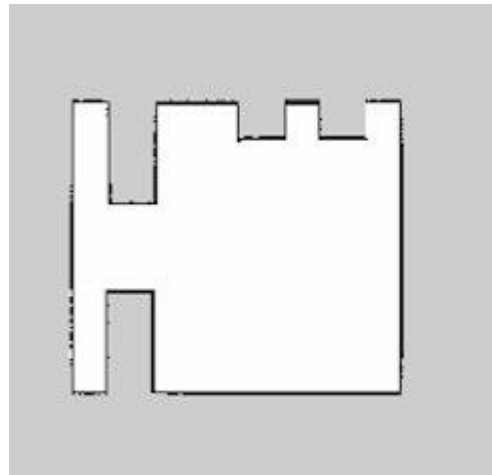


Рисунок 6 — Карта на основе  
HectorSlam

## 5 РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

Для оценки качества карты были выбраны такие метрики, как коэффициент Жаккара и индекс структурного сходства. А для оценки построения траектории была выбрана абсолютная ошибка траектории.

Алгоритм	Коэффициент Жаккара	Индекс структурного сходства
GMapping	0.9	0.93
Hector SLAM	0.87	0.89
Cartographer	0.86	0.89
ORB-SLAM2	0.8	0.85

Таблица 5 - Метрики для построения карты

Алгоритм	Абсолютная ошибка траектории
GMapping	1.30 м
Hector SLAM	1.54 м
Cartographer	1.56 м
ORB-SLAM2	2.23 м

Таблица 6 - метрики для оценки траектории

## ЗАКЛЮЧЕНИЕ

В данной лабораторной работе был проведен аналитическое сравнение существующих алгоритмов для решения задачи одновременного построения карты и локализации.

Также была сконструирована система для проведения эксперимента и произведен эксперимент с помощью симуляционной среды Gazebo.

Из анализа полученных данных, выявлено, что наиболее точным алгоритмом является GMapping. Hector SLAM и Cartographer тоже показали хорошие результаты. ORB-SLAM2 относительно других SLAM показал наихудшие результаты, хотя на часто на практике разница между этими алгоритмами не существенная и при этом ORB-SLAM2 обходится без дорогостоящих лидаров, что выделяет его из других исследуемых алгоритмов.

Результаты данного исследования подчеркивают важность тщательного выбора алгоритма для обеспечения оптимальной производительности системы.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Taketomi T., Uchiyama H., Ikeda S. Visual SLAM algorithms: A survey from 2010 to 2016 //IPSJ transactions on computer vision and applications. – 2017. – Т. 9. – С. 1-11.
2. R. Chatila and J. Laumond, "Position referencing and consistent world modeling for mobile robots," *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, St. Louis, MO, USA, 1985, pp. 138-145, doi: 10.1109/ROBOT.1985.1087373.
3. Martin A. Fischler and Robert C. Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6 (June 1981), 381–395. <https://doi.org/10.1145/358669.358692>
4. Bukhori, Imam & Ismail, Zool. (2017). Detection of kidnapped robot problem in Monte Carlo localization based on the natural displacement of the robot. *International Journal of Advanced Robotic Systems*. 14. 172988141771746. 10.1177/1729881417717469.
5. A. J. Davison, I. D. Reid, N. D. Molton and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052-1067, June 2007, doi: 10.1109/TPAMI.2007.1049.
6. Engels C., Stewénus H., Nistér D. Bundle adjustment rules //Photogrammetric computer vision. – 2006. – Т. 2. – №. 32.
7. Nistér D. An efficient solution to the five-point relative pose problem //IEEE transactions on pattern analysis and machine intelligence. – 2004. – Т. 26. – №. 6. – С. 756-770.
8. Jiachen Z. et al. Bag-of-words based loop-closure detection in visual SLAM //Advanced Optical Imaging Technologies. – SPIE, 2018. – Т. 10816. – С. 104-111.

9. Engel J., Sturm J., Cremers D. Semi-dense visual odometry for a monocular camera //Proceedings of the IEEE international conference on computer vision. – 2013. – C. 1449-1456.
10. Engel J., Schöps T., Cremers D. LSD-SLAM: Large-scale direct monocular SLAM //European conference on computer vision. – Cham : Springer International Publishing, 2014. – C. 834-849.
11. Baker S., Matthews I. Lucas-kanade 20 years on: A unifying framework //International journal of computer vision. – 2004. – T. 56. – C. 221-255.
12. Forster C., Pizzoli M., Scaramuzza D. SVO: Fast semi-direct monocular visual odometry //2014 IEEE international conference on robotics and automation (ICRA). – IEEE, 2014. – C. 15-22.
13. Wang R., Schworer M., Cremers D. Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras //Proceedings of the IEEE international conference on computer vision. – 2017. – C. 3903-3911.
14. Zhang Z. Microsoft kinect sensor and its effect //IEEE multimedia. – 2012. – T. 19. – №. 2. – C. 4-10.
15. Newcombe R. A. et al. Kinectfusion: Real-time dense surface mapping and tracking //2011 10th IEEE international symposium on mixed and augmented reality. – Ieee, 2011. – C. 127-136.
16. Chetverikov D. et al. The trimmed iterative closest point algorithm //2002 International Conference on Pattern Recognition. – IEEE, 2002. – T. 3. – C. 545-548.
17. Zhang S., Zheng L., Tao W. Survey and evaluation of RGB-D SLAM //IEEE Access. – 2021. – T. 9. – C. 21367-21387.
18. Concha A., Civera J. RGBDTAM: A cost-effective and accurate RGB-D tracking and mapping system //2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). – IEEE, 2017. – C. 6756-6763.

19. Fontan A., Civera J., Triebel R. Information-driven direct rgb-d odometry // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2020. – C. 4929-4937.
20. Civera J., Lee S. H. Rgb-d odometry and slam // RGB-D Image Analysis and Processing. – 2019. – C. 117-144.
21. Endres F. et al. An evaluation of the RGB-D SLAM system // 2012 IEEE international conference on robotics and automation. – IEEE, 2012. – C. 1691-1696.
22. Mur-Artal R., Montiel J. M. M., Tardos J. D. ORB-SLAM: a versatile and accurate monocular SLAM system // IEEE transactions on robotics. – 2015. – T. 31. – №. 5. – C. 1147-1163.
23. Mur-Artal R., Tardós J. D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras // IEEE transactions on robotics. – 2017. – T. 33. – №. 5. – C. 1255-1262.
24. Keller M. et al. Real-time 3d reconstruction in dynamic scenes using point-based fusion // 2013 International Conference on 3D Vision-3DV 2013. – IEEE, 2013. – C. 1-8.
25. Ma L. et al. CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM // 2016 IEEE International Conference on Robotics and Automation (ICRA). – IEEE, 2016. – C. 1285-1291.
26. Whelan T. et al. ElasticFusion: Dense SLAM without a pose graph // Robotics: science and systems. – 2015. – T. 11. – C. 3.
27. Zhang S., Zheng L., Tao W. Survey and evaluation of RGB-D SLAM // IEEE Access. – 2021. – T. 9. – C. 21367-21387.
28. Hsiao M. et al. Keyframe-based dense planar SLAM // 2017 IEEE International Conference on Robotics and Automation (ICRA). – Ieee, 2017. – C. 5110-5117.

29. Grisetti G., Stachniss C., Burgard W. Improved techniques for grid mapping with rao-blackwellized particle filters //IEEE transactions on Robotics. – 2007. – T. 23. – №. 1. – C. 34-46.
30. Grisetti G., Stachniss C., Burgard W. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling //Proceedings of the 2005 IEEE international conference on robotics and automation. – IEEE, 2005. – C. 2432-2437.
31. Nagla S. 2d hector slam of indoor mobile robot using 2d lidar //2020 international conference on power, energy, control and transmission systems (ICPECTS). – IEEE, 2020. – C. 1-4.
32. Harik E. H. C., Korsath A. Combining hector slam and artificial potential field for autonomous navigation inside a greenhouse //Robotics. – 2018. – T. 7. – №. 2. – C. 22.
33. Dwijotomo A. et al. Cartographer slam method for optimization with an adaptive multi-distance scan scheduler //Applied Sciences. – 2020. – T. 10. – №. 1. – C. 347.
34. Nüchter A. et al. Improving Google's Cartographer 3D mapping by continuous-time slam //The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. – 2017. – T. 42. – C. 543-549.
35. Biber P., Straßer W. The normal distributions transform: A new approach to laser scan matching //Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453). – IEEE, 2003. – T. 3. – C. 2743-2748.
36. Benhimane S., Malis E. Homography-based 2d visual tracking and servoing //The International Journal of Robotics Research. – 2007. – T. 26. – №. 7. – C. 661-676.

37. Yang M., Liu Y., You Z. Estimating the fundamental matrix based on least absolute deviation //Neurocomputing. – 2011. – T. 74. – №. 17. – C. 3638-3645.
38. Mur-Artal R., Tardós J. D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras //IEEE transactions on robotics. – 2017. – T. 33. – №. 5. – C. 1255-1262.
39. Mur-Artal R., Montiel J. M. M., Tardos J. D. ORB-SLAM: a versatile and accurate monocular SLAM system //IEEE transactions on robotics. – 2015. – T. 31. – №. 5. – C. 1147-1163.
40. Mur-Artal R., Tardós J. D. Orb-slam: Tracking and mapping recognizable //Proceedings of the Workshop on Multi View Geometry in Robotics (MVGRO)-RSS. – 2014.
41. Biber P., Straßer W. The normal distributions transform: A new approach to laser scan matching //Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453). – IEEE, 2003. – T. 3. – C. 2743-2748.
42. Kaess M., Ranganathan A., Dellaert F. iSAM: Incremental smoothing and mapping //IEEE Transactions on Robotics. – 2008. – T. 24. – №. 6. – C. 1365-1378.
43. Hess W. et al. Real-time loop closure in 2D LIDAR SLAM //2016 IEEE international conference on robotics and automation (ICRA). – IEEE, 2016. – C. 1271-1278.
44. Zou Q. et al. A comparative analysis of LiDAR SLAM-based indoor navigation for autonomous vehicles //IEEE Transactions on Intelligent Transportation Systems. – 2021. – T. 23. – №. 7. – C. 6907-6921.
45. Открытый код GMapping:  
[https://github.com/ros-perception/slam\\_gmapping](https://github.com/ros-perception/slam_gmapping)
46. Открытый код Hector SLAM:  
[https://github.com/tu-darmstadt-ros-pkg/hector\\_slam](https://github.com/tu-darmstadt-ros-pkg/hector_slam)

47. Открытый код Cartographer:

<https://github.com/cartographer-project/cartographer>

48. Открытый код ORB-SLAM2:

[https://github.com/raulmur/ORB\\_SLAM2](https://github.com/raulmur/ORB_SLAM2)

49. Библиотека для нелинейной оптимизации:

<https://github.com/ceres-solver/ceres-solver>

50. Библиотека для графовой оптимизации:

<https://github.com/RainerKuemmerle/g2o>