

Федеральное государственное бюджетное учреждение
«Национальный исследовательский центр „Курчатовский институт“»

На правах рукописи

Московский Антон Дмитриевич

**МЕТОДЫ И АЛГОРИТМЫ РАСПОЗНАВАНИЯ СЦЕН ДЛЯ
ЗАДАЧ ГЛОБАЛЬНОЙ ЛОКАЛИЗАЦИИ МОБИЛЬНЫХ
РОБОТОВ**

Специальность 2.3.5 –

Математическое и программное обеспечение вычислительных машин,
комплексов и компьютерных сетей

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель
д.т.н., доцент
Карпов Валерий Эдуардович

Москва – 2026

Оглавление

Оглавление	2
Введение	5
Глава 1. Современные методы семантической локализации	13
1.1. Методы визуальной локализации и картирования	13
1.2. Распознавание сцен в задаче локализации	15
1.2.1. Методы на основе поиска геометрических особенностей	15
1.2.2. Развитие методов визуального сходства в применении к задачам локализации	17
1.2.3. Наборы данных для тестирования методов распознавания сцен	18
1.2.4. Используемые метрики оценки качества решения задачи распознавания сцен	21
1.3. Определение положения в задаче семантической локализации	22
1.3.1. Методы определения положения в задаче локализации по визуальным ориентирам	22
1.3.2. Методы определения положения по трехмерным данным	24
1.4. Направление исследований и выводы	25
Глава 2. Теоретические основы разработанных методов и моделей	28
2.1. Метод распознавания сцен на основе поиска изоморфного подграфа	28
2.1.1. Постановка задачи распознавания сцены	28
2.1.2. Определение расстояний до объектов и их погрешностей	29
2.1.3. Общий подход с использованием изоморфизма подграфов	32
2.1.4. Прямое сравнение расстояний на рёбрах	37
2.1.5. Кластеризация отношений	40
2.1.6. Мягкое расширение метода	44
2.1.7. Учёт визуального сходства	49
2.2. Недоопределённая модель локализации	51
2.2.1. Постановка задачи локализации по ориентирам	51
2.2.2. Описание N-моделей	52

2.2.3. Общие принципы формирования Н-модели для задачи локализации ..	55
2.2.4. Ограничения, задаваемые одним ориентиром	57
2.2.5. Глобальная Н-локализация	59
2.2.6. Непрерывная Н-локализация	64
2.2.7. Проблема исчерпывающего оценивания	66
2.2.8. Использование дополнительных сенсоров в Н-локализации	68
2.2.9. Функции интерпретации для задачи локализации по пеленгу	69
2.3. Архитектура глобальной локализации по визуальным ориентирам	71
2.3.1. Общий подход	72
2.3.2. Использование полученных ограничений для методов локализации по облакам точек	73
2.3.3. Использование фильтра частиц для уточнения положения в ограничениях	74
2.3.4. Технология глобальной локализации по визуальным ориентирам	78
2.3.5. Требования к робототехническому комплексу	79
2.4. Метрики определения качества решения задачи распознавания сцен	80
2.5. Выводы к главе	83
Глава 3. Реализация разработанных методов и моделей	85
3.1. Метод распознавания сцен на основе поиска изоморфного подграфа	85
3.1.1. Модификация метода распознавания сцен на основе поиска максимальной клики	85
3.1.2. Деление семантической карты на подкарты	89
3.1.3. Программная реализация методов распознавания сцен	91
3.1.4. Исследование в среде имитационного моделирования	92
3.2. Метод Н-локализации	95
3.2.1. Программная реализация	95
3.2.2. Исследование в среде имитационного моделирования	96
3.3. Технология глобальной локализации по визуальным ориентирам	104

3.3.1. Система распознавания объектов.....	104
3.3.2. Реализация технологии глобальной локализации	112
3.3.3. Тестирование технологии глобальной локализации в среде Gazebo ..	114
3.4. Создание набора данных для тестирования	116
3.5. Выводы к главе.....	123
Глава 4. Экспериментальные исследования.....	125
4.1. Исследование метода распознавания сцен на реальном наборе данных	125
4.2. Исследование Н-локализации.....	136
4.2.1. Тестирование в среде Gazebo.....	136
4.2.2. Применение метода на реальных роботах.....	140
4.3. Исследование комплексной технологии глобальной локализации	146
4.3.1. Глобальная локализация на полученном наборе данных KITTI-360 ..	146
4.3.2. Применение комплексной технологии на реальном роботе.....	154
4.4. Выводы к главе.....	159
Заключение	161
Список используемых сокращений и условных обозначений	164
Список терминов	165
Список литературы	166
Приложение 1. Акты о внедрении результатов диссертационной работы	175
Приложение 2. Свидетельства о государственной регистрации РИД.....	180

Введение

Актуальность темы. Задача локализации (определения положения в пространстве) является краеугольной для большинства практических задач в области мобильной робототехники, ведь для автономного движения (навигации) роботу обычно требуется знать свои координаты в каждый момент времени. Развитие методов одновременного картирования и локализации (SLAM, simultaneous localization and mapping), позволяющих строить карту неизвестного пространства и проводить на ней локализацию, значительно расширило применение мобильных роботов. Подобные карты обычно представляют собой набор геометрических примитивов или облака точек, получающихся посредством «сканирования» пространства дальномерами. Помимо задачи SLAM отдельно выделяют задачу глобальной локализации (global localization) – т.е. определение положения уже на построенной карте. Определение положения по таким картам обычно происходит путём поиска лучшего «наложения» текущих данных с сенсоров робота на имеющуюся карту. Если во время задачи SLAM поиск такого наложения происходит в локальной области карты, то в случае глобальной локализации область задаётся размерами карты, что требует многочисленных наложений. Это же создаёт сложности, связанные с симметрией и однородностью пространства – в местах, где пространственные сенсоры одинаково хорошо накладываются на несколько точек карты.

Одним из способов решения описанных проблем является добавление размеченных на карте объектов (называемых семантической картой) определенных классов. Семантика позволяет понизить «размерность» пространства поиска, а также может добавить «особенности» к однообразной геометрии. С другой стороны, семантические объекты характеризуются тем, что зачастую представлены в среде во многих экземплярах, что затрудняет процедуру сопоставления видимых объектов с картой. Традиционно данная проблема решается с помощью методов распознавания связанных групп объектов - *методов распознавания сцен*. Их задача – выделение геометрических особенностей во взаимоположении объектов на наблюдаемой сцене и поиск этих же особенностей на семантической карте. Типичными проблемами при решении этих задач являются ошибки в системах распознавания ввиду неточности детектора или перекрытия другими объектами, а также ошибки в определении их положения в пространстве. Также следует учитывать изменчивость среды со временем. Хотя семантический уровень считается в целом более разнообразным, чем геометрический, всё же встречаются объекты, расположенные в некотором

свойственном для них порядке и практически не отличающиеся внешне друг от друга. Все перечисленные проблемы приводят к тому, что системы распознавания сцен должны иметь возможность предоставить несколько гипотез, а также дополнительные средства отсеивания ошибочных вариантов.

Стоит отметить, что решение задачи распознавания сцены является первым этапом в более общей задаче визуальной локализации, и лишь определяет соответствие между видимыми ориентирами и ориентирами на карте. Далее требуется по этим данным определить положение робота. Но работа с визуальными ориентирами обычно сопровождается большими погрешностями, особенно в части определения расстояний до них, а иногда во входных данных содержатся противоречивости, что приводит к весьма низкой точности определения положения. В тоже время, более точные методы локализации по данным дальномеров в постановке локализации по видимым ориентирам практически не используются, хотя локально могут обеспечить более высокую точность. Следовательно, требуется разработать специальные методы, позволяющие в таких условиях определить некую локальную область на карте, а затем произвести уточнение местоположения на ней геометрическими методами на основе облаков точек.

Таким образом, научная проблема создания эффективных способов сопоставления наблюдаемых объектов с семантической картой и дальнейшего определения положения робота по этим данным является **актуальной**. Речь идет о создании моделей и методов, позволяющих решать задачу глобальной локализации по визуальным данным. Актуальность определяется также и практическим аспектом работы. В ходе исследования была разработана комплексная технология глобальной локализации по визуальным ориентирам, позволяющая повысить автономность робототехнических средств за счёт решения задачи определения стартового положения робота на больших территориях и в сложных условиях, связанных с симметричностью и повторяемостью пространства. Разработкой методов распознавания сцен, в применении к локализации мобильных роботов, а также средствами верификации их результатов занимается ряд научных коллективов, что также говорит об актуальности темы.

Сегодня исследования в этой области семантической локализации охватывают много различных поднаправлений. Этим направлением в целом занимается ряд отечественных и зарубежных учёных, например: Ющенко А.С., Кирильченко А.А., Романов А.М., Платонов А.К., Яковлев К.С., Юдин Д.А., Колубин С.А., Фомин И.С., Burgard W., How J.P., Lusk P.C., Ankenbauer J. Liu Y.,

Matsuzaki S. и др. Много работ посвящено именно задачам семантического SLAM – построению карты с размеченными объектами, которые в дальнейшем используются в других робототехнических применениях.

Анализ работ как вышеупомянутых отечественных, так и ряда зарубежных исследователей позволяет сделать вывод, что создание устойчивых к перечисленным проблемам систем распознавания сцены и уточнения своего положения в применении к задачам глобальной локализации является актуальным.

Опираясь на вышесказанное, были поставлены цель и задачи.

Целью диссертационного исследования является повышение качества процедуры глобальной локализации мобильных робототехнических систем на основе применения методов распознавания сцен с учетом визуального сходства объектов и использования недоопределённых вычислений.

Для достижения заданной цели были определены следующие **задачи**:

1. Провести анализ существующих подходов в области глобальной локализации мобильных роботов, в частности по визуальным ориентирам и дальномерным данным, с целью выявления ограничений, влияющих на качество решения.

2. Разработать метод распознавания сцен, позволяющий найти соответствие между видимым набором объектов и семантической картой местности, которые способны работать в условиях высокой погрешности определения расстояния и т.н. выбросов (отсутствия нужных и наличия лишних объектов), а также определять сразу несколько гипотез и снабжать их численной степенью уверенности.

3. Разработать метод определения положения робота по данным визуальных ориентиров в условиях больших погрешностей, противоречивости данных и множественных гипотез, результат которого может быть использован классическими геометрическими методами определения положения для дальнейшего уточнения.

4. Объединить разработанные методы и модели в комплексную технологию глобальной локализации, дополнив её модификациями классических методов локализации с использованием облаков точек, которые способны работать в ограничениях.

5. Подготовить на основе открытых наборов данных систему тестирования, включающую в себя как данные, так и метрики оценивания решений задачи распознавания сцен и задач локализации по видимым ориентирам.

6. Разработать комплекс программных и технических решений по реализации созданных моделей и методов для решения прикладных задач в области локализации мобильных роботов, провести оценку работоспособности и эффективности предложенных решений.

Объектом исследования является процедура глобальной локализации мобильных роботов.

Предметом исследования являются методы, алгоритмы глобальной локализации на основе распознавания сцен.

Научная новизна работы заключается в следующем:

1. Предложен новый метод распознавания сцен на основе алгоритмов поиска изоморфного подграфа, позволяющий по видеоданным с бортовых сенсоров сопоставлять видимые объекты с семантической картой, учитывая визуальное сходство объектов методами на основе машинного обучения (пункты 4 и 7 паспорта специальности 2.3.5).

2. Предложен новый метод N-локализации на основе недоопределённых моделей, позволяющий получить решение задачи локализации по ориентирам в интервальных терминах (ограничениях), что делает возможным использование классических методов локализации по дальномерным данным (пункт 7 паспорта специальности 2.3.5.).

3. Предложена архитектура комплекса глобальной локализации, включающей в себя программные средства распознавания сцен, N-локализации и классических методов локализации, а также порядок их взаимодействия (пункт 3 паспорта специальности 2.3.5).

4. Разработаны метрики оценки качества и требования к данным для тестирования программных реализаций алгоритмов распознавания сцен (пункт 10 паспорта специальности 2.3.5).

Теоретическая значимость работы заключается в создании новых методов распознавания сцен, способных учитывать как геометрические особенности наблюдаемой сцены, так и визуальное сходство объектов. Применение подхода недоопределённых вычислений может использоваться для отсеивания гипотез и определения положения робота по визуальным ориентирам в условиях больших погрешностей.

Практическая значимость полученных результатов заключается в повышении эффективности решения задачи глобальной локализации роботов, оперирующих на больших/симметричных/повторяемых территориях, что расширяет их автономность и функциональность.

Методы исследования. В диссертационной работе применяются методы теории графов, недоопределённых вычислений, теории вероятности, оптимизации, машинного обучения, имитационного моделирования, технического зрения.

Положения, выносимые на защиту:

1. Новый метод распознавания сцен на основе поиска изоморфного подграфа с учетом визуального сходства и его реализация, повышающие точность получаемого решения в диапазоне 13-49% в сравнении с современными аналогами.

2. Метод и алгоритмы недоопределённой локализации, позволяющие проводить как глобальную, так и пошаговую (непрерывную) локализацию в ограничениях (интервалах), что позволяет уточнять положение классическими методами локализации, тем самым повышая до порядка общую точность и\или скорость локализации.

3. Комплексная технология глобальной локализации, реализующая предложенную архитектуру и включающая разработанные алгоритмы распознавания сцен и недоопределённой локализации, а также модификации классических алгоритмов локализации с использованием облаков точек, позволяющая повысить точность (от 5 до 48% по метрике попадания положения робота в доверительный интервал) и скорость процедуры (до 4х раз в сравнении с геометрическими методами) глобальной локализации по визуальным ориентирам и дальномерным данным.

4. Алгоритм формирования набора данных на основе KITTI-360, а также метрики, позволяющие проводить оценку качества алгоритмов распознавания сцен.

Достоверность результатов подтверждается численными экспериментами, проведенными на открытом наборе данных, а также в средах симуляции и на реальных роботах. Все реализованные алгоритмы подробно описаны, и их реализация частично выложена в открытый доступ, что позволяет проверить полученные результаты.

Апробация результатов проходила на ряде конференций и семинаров, основными из которых являются: V Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта» (БТС-ИИ-2019), 22-24 мая 2019 г., Санкт-Петербург; 4-я практическая конференция «ROS Meetup» (удалённый формат), 5 декабря 2020 г., Москва; XIX Национальная конференция по искусственному интеллекту с международным участием (КИИ-2021), 11-16 октября 2021 г., Таганрог; Семинар «Теория

управления организационными системами» (ТУОС), 3 февраля 2022 г., ИПУ РАН, Москва (удалённый формат); 34-я международная научно-техническая конференция «Экстремальная робототехника» (ЭР-2023), 23-24 ноября 2023 г., Санкт-Петербург; Семинар «Проблемы управления автономными робототехническими комплексами», 15 февраля 2024 г., ИПУ РАН, Москва; 7-я практическая конференция «ROS Meetup», 27-28 апреля 2024 г., Москва; XII Международная научно-практическая конференция «Интегрированные модели и мягкие вычисления в искусственном интеллекте» (ИММВ-2024), 14-17 мая 2024 г., Коломна; 35-я международная научно-техническая конференция «Экстремальная робототехника» (ЭР-2024), 29-30 октября 2024 г., Санкт-Петербург; Объединенный семинар "Интеллектуальные системы" и "Системное программирование", 13 февраля 2025 г., ИСИ СО РАН, Новосибирск; Донецкий международный научный круглый стол «Искусственный интеллект: теоретические аспекты, практическое применение», 28 мая 2025 г., ФГБНУ ИПИИ, Донецк (удалённый формат).

Внедрение результатов. Результаты исследований были использованы при:

1. Проведении НИР «Разработка экспериментального образца аппаратно-программного комплекса сбора и обработки данных для контроля характеристик и калибровки целевой аппаратуры КА ДЗЗ в части создания макета подвижной платформы для проведения подспутниковых наземных радиометрических измерений» (шифр «Мониторинг-СГ-1.3.1.1.1-МПП») (акт о внедрении, полезная модель).

2. В учебном процессе кафедры Информатики и вычислительных сетей Физтех-школы природоподобных, плазменных и ядерных технологий им. И.В. Курчатова (КНТ МФТИ) (акт о внедрении).

3. При проведении НИР «Исполнительные механизмы и системы электродвижения» по теме «Роботизированная многофункциональная коляска для людей с ограниченными возможностями» (2019-2022гг.) и НИР «Разработка робототехнических комплексов, в том числе энергетически автономных» (тематический план НИЦ «Курчатовский институт», тематика 1п.1.2, приказ №86 от 20.01.2023 г. «О проведении фундаментальной научно-исследовательской работы по комплексной теме "Прикладные междисциплинарные исследования в области создания природоподобных технологий"») (акт о внедрении).

Личный вклад. Все результаты, приведенные в данной работе, включая разработку алгоритмов, их программную реализацию и проведение экспериментов, получены автором лично. В работе [5] автору принадлежит обзор методов локализации, методов навигации без карты, а также участие в

обсуждении и формировании тенденций и открытых проблем области. Остальные работы выполнены без соавторов.

Соответствие паспорту специальности. Диссертация выполнена в соответствии с паспортом научной специальности 2.3.5 «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» по следующим пунктам:

– п.3. Модели, методы, архитектуры, алгоритмы, языки и программные инструменты организации взаимодействия программ и программных систем (создание архитектуры комплексной технологии локализации, включающей в себя программные средства методов распознавания сцен, Н-локализации и классических подходов).

– п.4. Интеллектуальные системы машинного обучения, управления базами данных и знаний, инструментальные средства разработки цифровых продуктов (создание гибридных методов распознавания сцен с использованием технологий машинного обучения, позволяющих учитывать визуальное сходство объектов).

– п.7. Модели, методы, архитектуры, алгоритмы, форматы, протоколы и программные средства человеко-машинных интерфейсов, компьютерной графики, визуализации, обработки изображений и видеоданных, систем виртуальной реальности, многомодального взаимодействия в социкиберфизических системах (создание систем распознавания сцен и локализации по ориентирам, опирающихся на обработку визуальных данных).

– п.10. Оценка качества, стандартизация и сопровождение программных систем (создание набора данных и разработка метрик оценки качества решений задачи распознавания сцен).

Публикации. Основные результаты изложены в работах [Московский, 2023; Московский, 2024a; Московский, 2024b; Moscowsky, 2021; Moscowsky, 2022a; Moscowsky и др., 2025; Московский, 2019a; Московский, 2025a; Московский, 2025b; Московский, 2025c], две в журналах, рекомендуемых ВАК по специальности 2.3.5, две в журналах, индексируемом в *Web of Science*, три в журналах, индексируемых в *Scopus*. Получен патент на полезную модель, в которой частично использовались разработанные алгоритмы и методы к локализации [Карпов и др., 2018]. По результатам создания программного обеспечения, реализующего разработанные методы, было получено два свидетельства о регистрации РИД в виде программ для ЭВМ [Московский, 2019b; Московский, 2020].

Структура и объем работы. Диссертация состоит из введения, четырех глав, заключения, списков сокращений и терминов, библиографического списка

из 113 наименований и приложения. Работа изложена на 182 страницах и включает в себя 77 рисунков и 39 таблиц.

Глава 1. Современные методы семантической локализации

1.1. Методы визуальной локализации и картирования

Совершенствование систем технического зрения, произошедшее в последнее десятилетие, способствовало развитию систем визуально-семантической локализации мобильных роботов, которые опираются на распознавание отдельных объектов, выделение типа подстилающей поверхности, определении типов помещения и т.п. Использование семантики позволяет решить ряд проблем, возникающих в классических подходах к задаче картирования и локализации (SLAM) на основе трехмерных данных без семантической разметки [Sahili и др., 2023], построенных с помощью лидаров и камер глубины. Эти проблемы в целом можно охарактеризовать как недостаток информации во входных данных (обычно облаках точек), не позволяющих эффективно проводить сравнение данных кроме как локально (расположенных близко друг к другу). Однако проводить подобное сравнение для несвязанных локально данных требуется для ряда подзадач, среди них: глобальная локализация (global localization, момент первичной локализации без начальной догадки) и замыкание (loop-closing, определение положения, в котором робот уже был).

К глобальной локализации также относятся ситуации, когда робот «потерялся» - попал в ситуацию, когда данные с сенсоров не соответствуют рассчитанному положению. Такие ситуации могут происходить ввиду ошибок алгоритмов локализации или в результате изменения положения робота третьими лицами (kidnapping). Обзорная работа [Yin и др., 2024], посвященная вопросам глобальной локализации в системах по облакам точек без использования семантики, выделяет проблему чувствительности к ракурсу у облаков точек, а также проблему времени вычисления для больших карт. Сюда же можно добавить проблемы периодичности и симметричности окружения, характеризующиеся тем, что городская среда в целом содержит ряд типовых построек и помещений, слабо различимых трехмерными сенсорами. Использование же семантики позволит перейти от сравнения облаков точек к сравнению наполнения окружения расположенными в нём объектами.

Представление объекта в виде материальной точки в пространстве позволяет значительно сократить исследуемую среду в сравнении с облаками точек, описывающих геометрию. Помимо положения, объект характеризуется своим описанием и семантикой, и таким образом происходит переход от карты пространства в виде облака точек к семантической карте, а задача локализации

требует соотношения наблюдаемых роботом объектов (сцены) с объектам семантической карты. Таким образом, ставится задача **распознавания сцен**, исследуемая в данной работе. Распознавание сцен (scene recognition) также иногда используется для определения типа помещения, а иногда и для семантической сегментации изображения. В мировом научном сообществе нет общего названия для данной задачи, в англоязычной научной литературе используются такие названия как: data association (сопоставление данных), scene matching (сопоставление сцен), registration problem (задача регистрации), object matching (сопоставление объектов), иногда задача называется по методам, её реализующим, например, graph matching (сопоставление графов), что будет видно из дальнейшего обзора.

Задача построения семантической карты остаётся за рамками данной работы, однако эта область также получает широкое развитие [Davison, 2018; Xia и др., 2020] в современной повестке, т.к. семантические карты используются не только для задач локализации. Так, например, на основе семантических карт создаются системы человеко-машинного взаимодействия, в т.ч. и на основе языка, приближенного к естественному, например, при управлении инвалидными колясками [Rovbo, Sorokoumov, 2022] или при управлении роботом-помощником [Zemskova и др., 2024].

Задача распознавания сцен является первым шагом в процедуре локализации, её решение позволяет сформировать задачу локализации по визуальным ориентирам и определить положение робота в пространстве. Общий подход к подобной двухшаговой локализации приведён на Рис. 1.1.

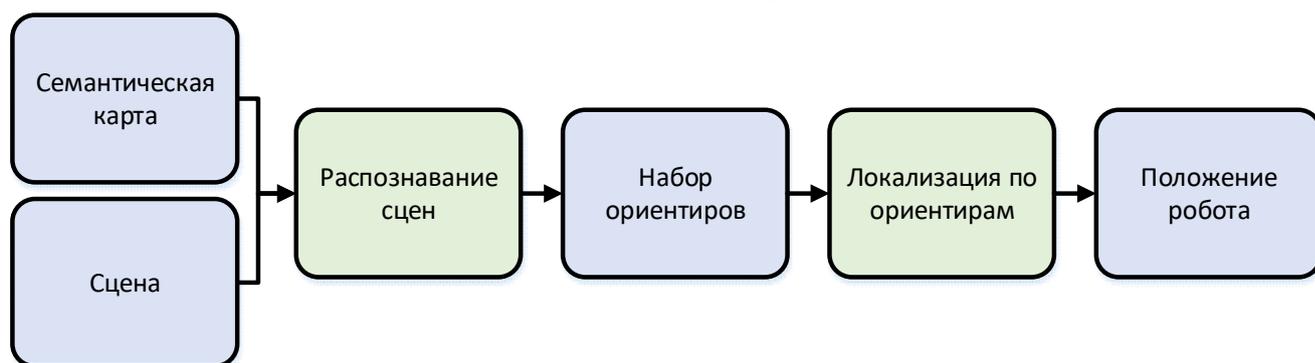


Рис. 1.1. Подход к локализации по семантическим данным

Далее в главе будут рассмотрены как методы распознавания сцен, так и методы локализации по ориентирам.

1.2. Распознавание сцен в задаче локализации

Задача распознавания сцен есть поиск соответствий между набором наблюдаемых объектов (сценой) и семантической картой (формальная постановка задачи приведена во второй главе данной работы). В обоих случаях (на сцене и на карте) объекты имеют свои положения в пространстве, заданные в несвязанных между собой системах координат, и некоторую семантическую информацию, например, описывающую тип объекта (класс). Помимо вопросов неоднозначности сопоставления, есть проблемы, связанные с системами детектирования объектов и определения их положения. Объект может быть не распознан или помечен как объект другого класса, также могут быть распознаны объекты, не присутствующие на карте. Методы распознавания сцен должны уметь работать с такими «выбросами» (outliers). Вопросы определения положения объектов в пространстве также связаны с системами распознавания и сенсорами, доступными роботу. Например, от того, насколько система способна отделить объект от окружения, зависит расчёт расстояния до него.

1.2.1. Методы на основе поиска геометрических особенностей

Поскольку в окружающей среде часто присутствует множество объектов одного и того же типа, то задача поиска сопоставления между сценой и картой исключительно на уровне объектов проблематична, особенно принимая во внимание тот факт, что в среде могут быть визуально идентичные объекты. Поэтому традиционный подход к данной задаче основывается на выделении геометрических особенностей наблюдаемого набора объектов и поиск их на карте. Классический метод *RANSAC* [Fishler, Bolles, 1980] применяется в решении задачи распознавания сцен в работах [Cramariuc и др., 2021; Dubé и др., 2020; Tinchev, Nobili, Fallon, 2018]. Однако использование данного метода затруднено при увеличении числа упомянутых выбросов в сценах [Raguram, Frahm, Pollefeys, 2008; Yang, Shi, Carlone, 2021].

Методы на основе анализа на графах стараются обойти эти проблемы. В таком подходе наблюдаемая карта и искомая сцена представляются в виде графов, вершинами которых выступают объекты, а ребрами – отношения между объектами, обычно построенные на основе их взаимоположения. Метод сравнения графов на основе дескрипторов случайного блуждания представлен в подходе *X-view* [Gawel и др., 2018] и в работе [Liu и др., 2019a]. Такой подход позволяет кодировать окружение объекта, случайно перемещаясь по рёбрам

графа, определенным на основании порога для расстояний между объектами. Такие дескрипторы сравнимы между собой и позволяют провести сопоставление графов на уровне объектов. Расширение случайных дескрипторов информацией о размере, ориентации было предложено в работе [Wu и др., 2023]. Другой способ кодирования ближайшего окружения вершины графа на основе его соседей с помощью семантических гистограмм также используется в работе [Guo и др., 2021] и в методе *GOReloc* [Wang, Jiang, Chen, 2024]. Сравнение графов двух сцен с помощью нейросетевых моделей представлено в работе [Kong и др., 2020], однако она не расширяется на обработку полноценной семантической карты. Задача в похожей постановке приведена в работе [Qin и др., 2021], использующей Венгерский алгоритм [Munkres, 1957] для поиска соответствия, и в подходе *GOSMatch* [Zhu и др., 2020] где используются гистограммные дескрипторы расстояний до ближайшего окружения вершины графа. Венгерский алгоритм также используется в подходе *BoxGraph* [Pramatarov и др., 2022], где максимизируется сходство элементов двух графов по их размерам. В отличие от других подходов, данный не занимается выделением отдельных объектов, а работает с кластерами одного семантического типа, позволяя добавлять такие протяженные объекты, как дороги и растительность. Однако метод не использует на этапе сопоставления графов расстояния между элементами: делается это только для расчёта финального веса решения, чтобы определить его допустимость. В методе *SemanticTopoLoop* [Сао и др., 2024] используется спектральный метод [Leordeanu, Hebert, 2005] для определения сопоставления между графами, максимизирующий введенную функцию сходства, основанную на соответствиях классов объекта и расстояниях между ними. Похожая техника максимизации функционала сходства на основе поиска собственных значений матрицы связи графов применена в работах, посвященных подходу *CLIPPER* [Lusk, Fathian, How, 2021; Lusk, How, 2024]. *CLIPPER* решает задачу регистрации для облаков точек, однако в работе [Peterson и др., 2024] этот же подход был применен и к объектам. Ряд работ разных коллективов [Ankenbauer и др., 2023; Matsuzaki и др., 2024a] посвящён использованию методов поиска наибольшей клики (полносвязного подграфа) на графе специального вида, где в качестве вершин выступают классовые соответствия между объектами исходных множеств, а в качестве рёбер – геометрические соответствия этих объектов. Такой подход позволяет быть очень устойчивым к разного рода выбросам, т.к. не требует полного соответствия между множествами. Комплексный подход на основе максимизации функционала сходства используется и в работе [Shaheer и др., 2023] для иерархических графов, описывающих помещение на разных

уровнях (этаж-комната-стена-дверной проём). Подобные многоуровневые графы, дополненные семантическими объектами, также используются в подходе *Hydra* [Hughes, Chang, Carlone, 2022], но сравнение происходит на основе дескрипторов, описывающих отдельные вершины графа по его окружению. Все перечисленные работы используют семантическую информацию в виде класса объекта, иногда дополняя её знаниями о размерах или ориентации в пространстве.

Основной проблемой перечисленных методов является то, что объекты одного класса в большинстве случаев никак не различимы между собой. Это приводит к тому, что методы, ища уникальные композиции объектов, неизбежно могут найти группы объектов близкие к друг другу (по сочетанию классов). Добавление возможности учитывать визуальное сходство/различие отдельных объектов позволило бы снизить влияние этой проблемы на качество дальнейшей локализации. Развитие соответственных подходов будет приведено в следующем подразделе.

1.2.2. Развитие методов визуального сходства в применении к задачам локализации

Развитие ИИ в области создания базисных (foundational) моделей, таких как LLM (large language models, большие языковые модели) и VLM (visual-language models, визуально-языковые модели) дало определённое развитие изложенным методам распознавания сцен, т.к. позволяет эффективно кодировать конкретный объект в виде вектора чисел и определять визуальное сходство с другими объектами. Это стало возможным благодаря созданию архитектуры CLIP (Contrastive Language-Image Pretraining) [Radford и др., 2021], которая позволяет получать робастные вектора благодаря одновременному обучению на изображениях и их текстовых описаниях. В работе [Mirjalili, Krawez, Burgard, 2023] используется как CLIP, так и большая языковая модель GPT-3 для формирования системы, которая позволяет извлечь описание окружающего помещения без начального знания о нём и производить локализацию «по типам» помещений. В 2024 году было представлено несколько работ по использованию CLIP в задаче распознавания сцен. Первой была работа [Matsuzaki и др., 2024b], в которой использовались только векторные представления объектов, позволяющие определить визуальное сходство объектов. Идея использовать подход, основанный как на геометрическом сопоставлении, так и на визуальном сходстве, является одним из ключевых в данной диссертации и впервые был изложен автором диссертации в [Московский, 2024b], где CLIP-признаки используются

для улучшения ранее созданного автором метода распознавания сцен на основе поиска изоморфного графа, описанного в [Московский, 2023], и метода на основе модификации поиска наибольшей клики, изложенного в [Московский, 2024a]. Также идея использования визуального сходства в подходах, опирающихся на геометрические особенности, была, примерно в это же время, предложена двумя другими коллективами (осенью 2024 года выложены как препринты на arxiv.org). Коллектив S. Matsuzaki и др. в продолжение своих работ [Matsuzaki и др., 2024a; Matsuzaki и др., 2024b] разработал смесь метода на основе поиска наибольшей клики и использовании CLIP-признаков [Matsuzaki, Tanaka, Shintani, 2024], а коллектив под руководством J. P. How также предложил подход [Peterson и др., 2024] на основе максимизации функции сходства двух графов, где добавились признаки, рассчитанные при помощи CLIP в продолжение своих работ [Lusk, Fathian, How, 2021; Lusk, How, 2024]. Факт одновременного подхода к этой теме несколькими независимыми коллективами говорит о её важности и необходимости проведения исследований в этой области.

1.2.3. Наборы данных для тестирования методов распознавания сцен

Важным аспектом многих современных исследований является наличие общепринятого набора данных (dataset, benchmark), на которых тестируются методы. Однако, несмотря на определённый интерес к теме семантической локализации с использованием средств распознавания сцен со стороны коллективов, занимающихся ею на протяжении нескольких лет, на сегодняшний день отсутствует общепринятый набор данных, который позволяет проводить оценку точности разработанных методов. Авторам указанных в обзоре работ приходится либо собирать свои наборы данных, либо адаптировать существующие из смежных областей, например, визуального SLAM.

Такой набор данных должен удовлетворять следующим критериям:

1. Иметь семантическую карту местности, содержащую идентификаторы, положение объектов в глобальной системе координат и их семантическую информацию.
2. Иметь набор сцен, где каждая сцена есть набор объектов с идентификаторами, положениями, заданными в локальной системе координат робота, и семантической информацией. Для каждой сцены должна быть предоставлена информация о связи локальных координат с глобальными (для проверки точности разрабатываемых методов локализации).

3. При этом крайне важно, чтобы идентификаторы семантической карты и идентификаторы объектов сцен совпадали для одних и тех же объектов (для проверки точности задачи распознавания сцен в отдельности).
4. Также для тестирования средств, позволяющих определять визуальное сходство, желательно иметь реальные или хотя бы фотореалистичные данные.

Единственный набор, удовлетворяющий перечисленным требованиям, найденный в открытом доступе, это ScanNet [Dai и др., 2017] (и его расширение ScanNet++ [Yeshwanth и др., 2023]). Однако данный набор разбит на сцены, которые охватывают небольшие помещения среднего размера – со стандартную комнату. Это не позволяет исследовать вопросы применимости методов распознавания сцен в сценариях работы на больших (в т.ч. открытых) пространствах. Исследованные наборы данных, потенциально подходящие для тестирования задач распознавания сцен, и в т.ч. использовавшиеся в приведённых работах по этой тематике, представлены в Табл. 1.1.

Табл. 1.1. Наборы визуальных данных для задачи локализации

Набор данных	Тип данных ¹	Семантическая карта	Разметка объектов сцены	3D-положение объектов (или сенсор глубины)	Связь объектов карты и сцены	Работы по распознаванию сцен
SYNTHIA [Ros и др., 2016]	SO	✗	✓	✓	-	[Gawel и др., 2018; Guo и др., 2021; Qin и др., 2021]
SceneNN [Hua и др., 2016]	RI	✓	✓	✓	✗	[Liu и др., 2019a]
KITTI [Geiger, Lenz, Urtasun, 2012]	RO	✗	✓	✓	-	[Ankenbauer и др., 2023; Guo и др., 2021; Kong и др., 2020; Pramatarov и др., 2022; Zhu и др., 2020]

¹ S – синтетические, R – реальные, I – в помещении, O – вне помещения

TUM [Sturm и др., 2012]	RI	✗	✗	✗	-	[Matsuzaki и др., 2024b; Matsuzaki, Tanaka, Shintani, 2024; Qin и др., 2021; Wang, Jiang, Chen, 2024; Wu и др., 2023]
7-scenes [Glocker и др., 2013]	RI	✗	✗	✗	-	[Qin и др., 2021; Wu и др., 2023]
uHumans [Rosinol и др., 2021]	SIO	✗	✓	✓	-	[Hughes, Chang, Carlone, 2022]
Katwijk [Hewitt и др., 2018]	RO	✓	✗	✗	-	[Ankenbauer и др., 2023]
KITTI-360 [Liao, Xie, Geiger, 2023]	RO	✓	✓	✓	✗	-
ScanNet [Dai и др., 2017]	RI	✓	✓	✓	✓	[Matsuzaki, Tanaka, Shintani, 2024]
ScanNet++ [Yeshwanth и др., 2023]	RI	✓	✓	✓	✓	-
Khimera-Multi [Tian и др., 2023]	RO	✗	✗	✗	-	[Peterson и др., 2024]
ICL-Data [Saeedi и др., 2019]	RI	✗	✗	✗	-	[Wang, Jiang, Chen, 2024]
ITLP-Campus [Melekhin и др., 2024]	RIO	✗	✓	✓	-	-
Sematic KITTI [Behley и др., 2019]	RO	✗	✓	✓	-	[Ankenbauer и др., 2023; Pramatarov и др., 2022]

Помимо указанных наборов данных, авторы работ по теме распознавания сцен также использовали симуляторы, например, AirSim [Shah и др., 2017] и CARLA [Dosovitskiy и др., 2017], ручную разметку из GoogleMaps, средство Revit от Autodesk, а также свои собственные наборы данных. Т.к. во всех приведенных наборах данных отсутствуют какие-то из необходимых для тестирования задачи распознавания сцены пункты, то авторам перечисленных работ требовалось самостоятельно дополнять эти наборы ручной или автоматической разметкой с использованием средств SLAM. При таком подходе всё ещё отсутствует связь между объектами сцены и объектами карты, что не позволяет оценить точность решения задачи распознавания сцен отдельно. Отчасти поэтому большинству исследователей приходилось рассматривать задачу распознавания сцен вкуче с целевой задачей, будь то глобальная локализация или замыкание. С одной стороны, это даёт возможность оценить весь подход, с другой, не позволяет сравнить между собой различные методы распознавания сцен.

Таким образом, создание набора данных, отвечающего всем сформулированным пунктам, является важной задачей в рамках исследований методов распознавания сцен и будет рассмотрена далее.

1.2.4. Используемые метрики оценки качества решения задачи распознавания сцен

Для тестирования качества методов распознавания сцен требуется определить соответствующие метрики. Однако в большинстве рассмотренных работ [Ankenbauer и др., 2023; Gawel и др., 2018; Guo и др., 2021; Liu и др., 2019b; Matsuzaki и др., 2024a; Matsuzaki и др., 2024b; Peterson и др., 2024] этой проблеме в отдельности не уделяется внимание. Задача сопоставления данных рассматривается вкуче с механизмами определения положения робота, которое уже оценивается другими метриками. С одной стороны, такой подход обоснован тем, что без определения положения результат сопоставления объектов не имеет особого практического смысла. С другой же стороны, с ростом количества подобных методов и их вариаций, важно проводить сравнения по единым метрикам.

В работе [Wu и др., 2023] используется простая метрика, показывающая абсолютное число корректно сопоставленных объектов, причем для разных наборов тестовых данных истинное число разнится, что несколько затрудняет интерпретацию полученных результатов. В методе *GOReloc* [Wang, Jiang, Chen,

2024] для оценки качества сопоставления объектов уже используется нормированная метрика *Accuracy*, измеряемая в процентах. Авторы явно не указывают способ её расчета, однако, вероятнее всего, это процент правильно соотнесённых объектов к общему числу объектов в сцене. Авторы метода *CLIP-Clique* [Matsuzaki, Tanaka, Shintani, 2024] используют распространенные в задачах классификации метрики *Precision* и *Recall*, но тоже без подробного указания, как они определяются для задачи сопоставления объектов. Также стоит отметить, что методы распознавания сцен обычно выдают несколько результатов, поэтому полученные метрики обычно применяют к ответу с самым высоким коэффициентом уверенности.

Таким образом, можно сделать вывод, что разработка и описание единых метрик является также важной задачей, в том числе в плане создания метрик, оценивающих число выдаваемых ответов и зависимость точности от коэффициентов уверенности, рассчитываемых методами.

1.3. Определение положения в задаче семантической локализации

1.3.1. Методы определения положения в задаче локализации по визуальным ориентирам

Задача распознавания сцен не решается сама по себе, являясь первым шагом в рамках глобальной локализации робота. После определения соответствий между объектами на сцене и объектами на семантической карте требуется найти положение робота. Для определения преобразования между локальными координатами сцены и глобальными координатами карты по соответствиям объектов используются также различные методы. Формальная постановка задачи поиска положения по видимым ориентирам приведена во второй главе данной диссертации.

Методы триангуляции широко используются во многих сферах, включая геодезию, морскую навигацию, радиопеленгацию. Обычно для локализации на плоскости требуется не менее трёх ориентиров. Эта задача решается геометрическими подходами на основе пересечения окружностей или при помощи тригонометрии [Cohen, Koss, 1993; Font, Batlle, 2006; Pierlot, Droogenbroeck Van, 2014]. Однако в робототехнике для учета погрешностей измерения расстояний или углов направлений на ориентиры чаще прибегают к вероятностным подходам, которые также не зависят от числа наблюдаемых ориентиров и используют историю предыдущих измерений. Классические

вероятностные подходы, на основе фильтра Калмана, гистограммного фильтра (ГФ) и фильтра частиц (ФЧ), приведенные в монографии [Thrun, Wolfram, Fox, 2005], способны решать задачу определения положения по ориентирам. Однако фильтру Калмана и ФЧ обычно требуется выполнить несколько итераций, чтобы сойтись к какому-то устойчивому решению. Также эти методы могут попасть в т.н. локальный минимум, в котором удовлетворяются заданные критерии.

Различные варианты оптимизации также используются для задачи определения положения робота (камеры) по визуальным ориентирам. Например, в упомянутом подходе *X-view* [Gawel и др., 2018] используется нелинейный оптимизатор Гаусса-Ньютона [Fletcher, 1987] для оценки апостериорного максимума положения робота. В работе [Matsuzaki и др., 2024a] используется оптимизация положения классическим алгоритмом Левенберга-Марквардта [Levenberg, 1944; Marquardt, 1963]. Подобный подход также применяется и в работах [Matsuzaki и др., 2024b; Wang, Jiang, Chen, 2024]. Одна из вариаций такого алгоритма [Arun, Huang, Blostein, 1987] на основе сингулярного разложения (SVD – Singular Value Decomposition) широко применяется и в других работах [Ankenbauer и др., 2023; Peterson и др., 2024; Wu и др., 2023].

Ряд работ [Guo и др., 2021; Liu и др., 2019a] использует алгоритмы семейства ICP [Rusinkiewicz, Levoy, 2001] (Iterative Closest Point) для определения положения робота. Эти методы активно используются для работы с трехмерными облаками точек, однако могут быть применены и к задаче локализации по видимым объектам. В работе [Liu и др., 2019a] был использован интересный подход, где проводилась регистрация методом ICP на уровне облаков точек отдельных объектов, а далее определялось положение камеры.

Направление, использующее облака точек в дополнение к уровню семантики, выглядит перспективным, поскольку позволяет достичь дополнительной точности. Однако данное направление не рассматривалось ни в одной из упомянутых работ по данной тематике, за исключением [Liu и др., 2019a]. Связано это может быть с тем, что остаётся открытым вопрос перехода от объектов обратно к облакам точек. Требуется каким-то образом ограничить зону применения методов по облакам точек. Сделать это на основе рассчитанного положения проблематично, поскольку встаёт вопрос определения размеров подобной области. В то же время существует подход интервального анализа, который решает эти вопросы.

Монография [Jaulin, Didrit, Walter, 2001] – одна из первых работ по интервальному анализу и применению его в робототехнических задачах, в том числе в задаче локализации на основе дальномеров на двухмерных картах

препятствий. Работа [Jiang, Ehambram, Wagner, 2022] использует этот же подход в нотации локализации по визуальным ориентирам. В симуляции происходит сравнение разработанного подхода и локализации вероятностными методами: уточняется, что второй класс подходов обычно даёт меньший разброс неопределённости, однако интервальный подход показывает лучшие результаты в случае не гауссовой модели шумов в измерениях. В нашей стране интервальный анализ представлен развитием подхода недоопределённых вычислений (моделей), предложенных А.С. Нариньяни [Нариньяни, 1986], который развивается далее его последователями [Загорулько, Загорулько, 2020]. Недоопределённые модели (Н-модели) представляют собой обобщённую вычислительную процедуру, позволяющую работать с данными практически любого характера, включая и интервалы. Считается, что Н-модели являются обобщением направления программирования в ограничениях (Constraint Programming) и развивались независимо от него [Handbook of Constraint Programming, 2006]. Применение данного аппарата к задачам в робототехнике, включая локализацию по ориентирам, также было предпринято в [Карпов, 2009]. Таким образом, методы интервального анализа выглядят перспективными для работы совместно с результатами задачи распознавания сцен, поскольку позволяют получать интервальные оценки на положения, которые в дальнейшем удобно использовать методам локализации по трехмерным данным. Это позволит соединить подходы к локализации по ориентирам и по облакам точек.

Также стоит отметить, что в большинстве рассмотренных работ перечисленные методы применяются к нескольким лучшим результатам распознавания сцен, после чего происходила дополнительная верификация на основе ошибок в положении объектов сцены и карты, переведённых в одну координатную систему. Развитие методов локализации, способных определять качество различных гипотез, получаемых от систем распознавания сцен, и способных работать с набором таких гипотез также является актуальной задачей.

1.3.2. Методы определения положения по трехмерным данным

Облака точек - данные, которые получают распространёнными в робототехнике сенсорами, такими как лидары и камеры глубины. В задаче локализации требуется выполнять регистрацию облаков точек – определять преобразование координат для одного из облаков, так, чтобы оно совпало с другим. При этом ввиду дискретности облаков точек критерий совпадения также является неоднозначным. Один из подходов – создание дескрипторов. Дескриптор

– это структура, часто связанная с точкой облака и её ближайшим окружением. Среди методов выделения дескрипторов можно отметить такие как FPFH (Fast Point Feature Histograms) [Rusu, Blodow, Beetz, 2009], обучаемые подходы NeRF (Neural Radiance Fields) [Mildenhall и др., 2022], Minkloc3d [Komorowski, 2021] и др. [Rosen и др., 2021]. Дескрипторы позволяют либо понизить размерность исследуемого пространства, либо увеличить «уникальность» отдельных точек. Построенные дескрипторы обычно сравнимы между собой, что позволяет применять методы сопоставления дескрипторов друг другу, такие как базирующиеся на упомянутом RANSAC [Fishler, Bolles, 1980], на основе алгоритмов на графах [Yang, Shi, Carlone, 2021], оптимизационные алгоритмы [Lusk, How, 2024], обучаемые подходы [Choy, Dong, Koltun, 2020]. Методы ICP [Rusinkiewicz, Levoy, 2001] и его дальнейшие расширения, например [Vizzo и др., 2023], а также NDT (Normal Distribution Transform) [Saarinen и др., 2013] используются для последующего уточнения положения. Также используются такие вероятностные подходы как ГФ или ФЧ [Barfoot, 2017], примененные к облакам точек [Chen и др., 2021].

Одной из основных проблем всех перечисленных методов является работа с большими объемами данных, которые потенциально содержат множественные, близкие друг к другу решения. Наложение ограничений на положение робота в виде интервалов может значительно уменьшить время выполнения, а также повысить точность. Для некоторых из подходов интервальные ограничения могут быть напрямую использованы, например, в качестве ограничений для оптимизационных алгоритмов или при инициализации вероятностных методов. Также по данным ограничениям всегда можно «обрезать» облако точек карты, проводя локализацию только в интересующей области.

1.4. Направление исследований и выводы

Задача распознавания сцен определённо вызывает интерес в современной робототехнике, а в последнее время для её решения активно используются новые наработки в области ИИ и технического зрения, позволяющие решать проблему идентичности объектов. На Рис. 1.2 приведена динамика развития методов распознавания сцен, где показана тенденция использования новых подходов. По оси ординат на Рис. 1.2 отмечены цветом подходы, в рамках которых сделаны соответствующие работы. Если работа рассматривает несколько подходов, то принадлежность к вторичному подходу указана цветом заливки (кроме белого) кружка, отмечающим работу. Цветом окантовки (кроме черного) отмечено

пересечение коллективов авторов, где красным выделены работы автора диссертации. Стрелками указано использование работ в качестве сравнения со своим подходом (baselines).

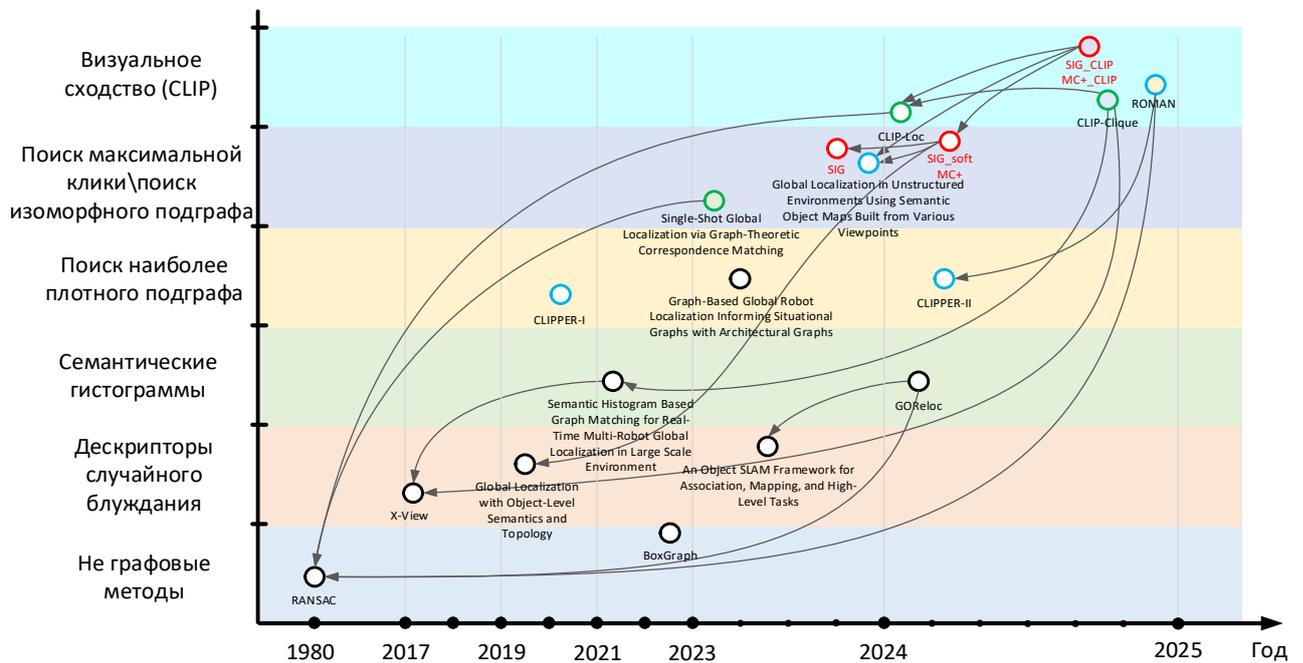


Рис. 1.2. Некоторые из работ, посвященных методам распознавания сцен, помещенные на временную линию

На Рис. 1.2 видна тенденция использования методов из теории графов вкуче с применением средств визуального сходства детектируемых объектов при помощи CLIP, данная диссертация также относится к данному направлению. Сложности сравнения работ между собой (факт сравнения отмечен стрелками на Рис. 1.2) можно объяснить отсутствием как специализированных наборов данных, так и выложенных в открытый доступ программных реализаций предлагаемых алгоритмов.

Таким образом, по результатам приведенного анализа области для достижения поставленной цели требуется выполнить следующие задачи:

1. Разработать метод распознавания сцен, позволяющий найти соответствие между видимым набором объектов и семантической картой местности, которые способны работать в условиях высокой погрешности определения расстояния и т.н. выбросов (отсутствия нужных и наличия лишних объектов), а также определять сразу несколько гипотез и снабжать их численной степенью уверенности.

2. Разработать метод определения положения робота по данным визуальных ориентиров в условиях больших погрешностей, противоречивости данных и множественных гипотез, результат которого может быть использован классическими геометрическими методами определения положения для дальнейшего уточнения.
3. Объединить разработанные методы и модели в комплексную технологию глобальной локализации, дополнив её модификациями классических методов локализации с использованием облаков точек, которые способны работать в ограничениях.
4. Подготовить на основе открытых наборов данных систему тестирования, включающую в себя как данные, так и метрики оценивания решений задачи распознавания сцен и задач локализации по видимым ориентирам.
5. Разработать комплекс программных и технических решений по реализации созданных моделей и методов для решения прикладных задач в области локализации мобильных роботов, провести оценку работоспособности и эффективности предложенных решений.

Глава 2. Теоретические основы разработанных методов и моделей

Глава посвящена теоретическому аспекту разрабатываемых методов и моделей. Будут рассмотрены формальные постановки решаемых задач и предложены:

- метод распознавания сцен на основе поиска изоморфного подграфа (раздел 2.1);
- метод N-локализации по ориентирам;
- архитектура комплексной технологии глобальной локализации по визуальным и дальномерным данным;
- метрики оценки качества задачи распознавания сцен.

2.1. Метод распознавания сцен на основе поиска изоморфного подграфа

В разделе предлагается новый метод распознавания сцен на основе алгоритмов поиска изоморфного подграфа, позволяющий учитываться также и визуальное сходство наблюдаемых объектов сцены и объектов карты.

2.1.1. Постановка задачи распознавания сцены

Пусть мир (пространство, в котором оперирует робот) описывается множеством объектов o , для которых задан класс c , декартовы координаты в пространстве p и визуальный образ I (представленный каким-то способом внешний вид объекта, например его двумерным изображением):

$$W = \{o_i^W = \langle c_i^W, p_i^W, I_i^W \rangle\}, c \in \mathbb{N}, p \in \mathbb{R}^3. \quad (2.1)$$

Это множество, строго говоря, неизвестно в принципе, однако система распознавания объектов робота способна получить образ множества (фрагмент окружающего пространства) W – аналогично устроенное множество объектов S (сцена):

$$f_r: W \rightarrow S, S = \{o_i^S = \langle c_i^S, p_i^S, I_i^S \rangle\}, \quad (2.2)$$

где f_r – функция, реализующая процедуру распознавания. Данная функция сопоставляет часть объектов из W объектам из S , однако может изменять все атрибуты этих объектов, равно как и добавлять новые (так называемые выбросы). Аналогичная по смыслу функция f_m отражает процедуру семантического картирования (автоматического или ручного):

$$f_m: W \rightarrow M, M = \{o_i^M = \langle c_i^M, p_i^M, l_i^M \rangle\}, \quad (2.3)$$

где M – множество объектов семантической карты. В множествах S и M положения объектов p задаются в локальных системах координат для каждого множества, и связь между ними неизвестна. Тогда **задача распознавания сцены** это поиск такого соответствия f из S в M , которое сохраняет прообразы объектов в множестве W :

$$f: S \rightarrow M: \forall o_i^W \in W, f_r(o_i^W) \neq \emptyset, f_m(o_i^W) \neq \emptyset, f(f_r(o_i^W)) = f_m(o_i^W). \quad (2.4)$$

Иллюстрация данной постановки приведена на Рис. 2.1.

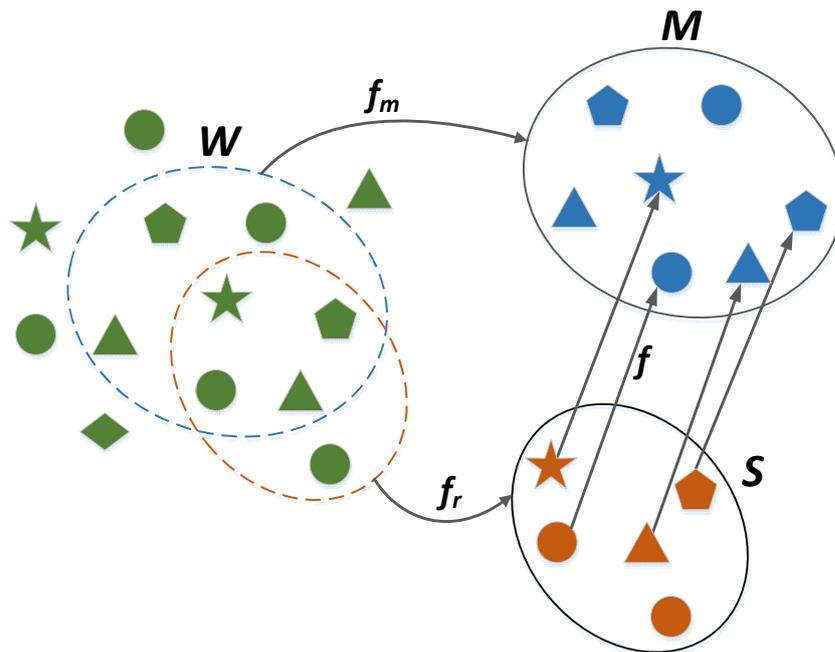


Рис. 2.1. Иллюстрация постановки задачи распознавания сцен

Проблема данной постановки заключается в том, что в реальном мире ни множество W , ни функции f_r и f_m в явном виде не заданы. Однако при тестировании в средах симуляции можно получить все требуемые соотношения. Это же возможно и при использовании специально размеченных наборов данных, где известно соотношение между объектами множеств S и M , т.е. функция $f_r^{-1}f_m$. Разработав методы распознавания сцен в подобных условиях, можно рассчитывать на сохранение их возможностей и в реальном применении.

2.1.2. Определение расстояний до объектов и их погрешностей

В методах распознавания сцен ключевым фактором является определение так называемых отношений между объектами, которые, в большинстве случаев,

формируются на основе расчёта евклидова расстояния между ними. Однако на семантической карте объект обычно характеризуется положением своего центра (реже – с указанием размеров). На наблюдаемой сцене расстояние можно определить с помощью сенсоров и зачастую это расстояние до видимой поверхности объекта, которая редко совпадает с физическим центром. Развитие систем трёхмерной детекции [Chen и др., 2024] по данным лидара отчасти решает такие проблемы, но не для всех конфигураций сенсоров такие методы доступны. Более классический подход – это определение объектов по двумерным изображениям и дальнейшее определение расстояния до них по карте глубины или облаку точек. При таком подходе расстояния между объектами на карте и сцене будут отличаться друг от друга. В связи с этим требуется определить способ сравнения двух расстояний между объектами в разных множествах (карты M и сцене S) с учётом описанной специфики.

Для того чтобы оценить погрешность определения расстояния, полученного таким способом, предлагается следующий подход. Положение каждого объекта в локальных координатах робота задаётся двумя величинами – углом на объект α , для определения которого достаточно лишь изображения и параметров камеры, и расстоянием до объекта r , получаемым при помощи сенсора глубины. Для определения расстояния с помощью сенсора глубины, формирующего облако точек, требуется совершить перевод этого облака в пространство изображения, а далее выделить те точки, что относятся к объекту. Для определения точек используется обрамляющий прямоугольник (брус) объекта или сегментационная маска, позволяющая отсеять лишние точки. Для определения расстояния обычно используется усреднение полученных точек, чаще всего в виде медианы, т.к. помимо усреднения требуется еще отбросить точки других объектов, случайно попавших в маску [Lafuente-Arroyo и др., 2024; Masoumian и др., 2021]. Но такой подход может плохо работать для не сплошных и частично прозрачных объектов, когда в область маски попадает слишком много лишних точек окружения. Также в зависимости от формы объекта и ракурса, с которого он наблюдается, медиана может давать разное расстояние. Из-за этого имеет смысл определять кратчайшее расстояние до объекта (его ближайшей к роботу точки) и получать расстояние до центра объекта, добавив к нему поправку - половину примерных размеров данного объекта. Т.к. в измерениях расстояния присутствуют шумы за счет неточного выделения маски (попадают пиксели соседних объектов и фона), то предлагается брать не минимальное расстояние, а отсекал выбросы, рассчитывая первый квартиль всех расстояний до объекта.

Получив расстояние до ближайшей точки объекта описанным способом, предлагается перевести его в расстояние до центра объекта, основываясь на знании о физических размерах объектов тех или иных типов. На заранее размеченных данных возможно рассчитать среднюю ошибку dr между расстоянием от робота до центра объекта и расстоянием до ближайшей точки, а также среднеквадратичное отклонение этой величины σr . Данные величины должны быть рассчитаны для каждого класса объектов. Имея данные значения для расстояния и угла на объект, можно оценить погрешность измерения расстояния между двумя объектами, которое в свою очередь рассчитывается как:

$$D(o_1, o_2) = \sqrt{(r_1 + dr_1)^2 + (r_2 + dr_2)^2 - 2(r_1 + dr_1)(r_2 + dr_2)\cos(\alpha_1 - \alpha_2)} \quad (2.5)$$

Для оценки погрешности данного значения требуется определить параметры эллипса рассеивания для каждого объекта – σr и $\sigma \alpha$, т.е. значения дисперсий данных величин. Согласно [Вентцель, 1999], дисперсии рассчитываются следующим образом из значений среднеквадратических отклонений:

$$\sigma r = dr^2, \quad (2.6)$$

$$\sigma \alpha = r^2 \left(\left(\frac{1}{\cos d\alpha} \right)^4 d\alpha + \frac{\sin d\alpha}{\cos^3 d\alpha} d\alpha^2 \right). \quad (2.7)$$

Далее, следуя принципу композиции нормальных законов, можно рассчитать результирующий эллипс рассеивания для расстояния между исходными объектами и взять радиус этого эллипса по направлению рассчитываемого расстояния. Подробные выкладки представлены в [Вентцель, 1999] (параграф 12.8), искомая величина является функцией от указанных, рассчитанных ранее значений:

$$\sigma D(o_1, o_2) = F(r_1 + dr_1, r_2 + dr_2, \sigma r_1, \sigma r_2, \sigma \alpha_1, \sigma \alpha_2, D). \quad (2.8)$$

Полученное значение предлагается использовать как приближение среднеквадратичного отклонения расстояния между объектами (Рис. 2.2), рассчитанного по входным данным (2.8).

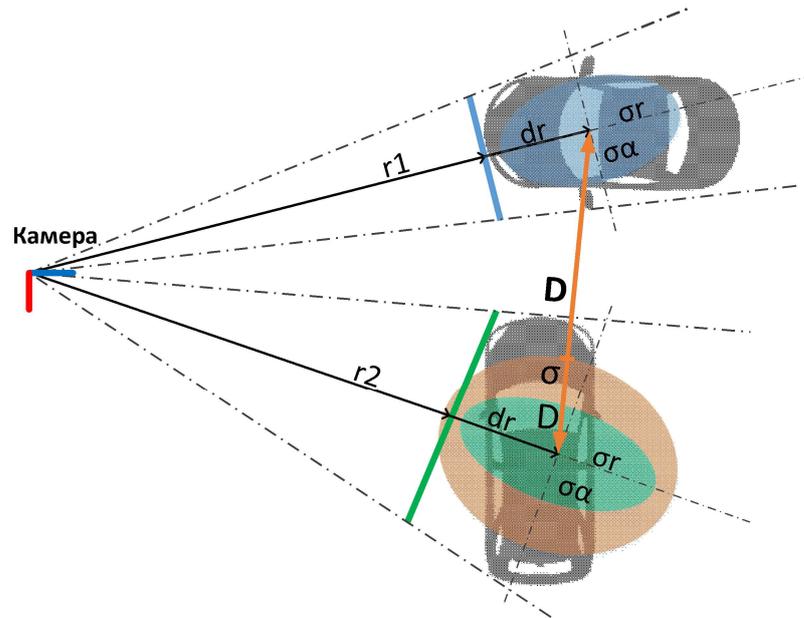


Рис. 2.2. Оценка погрешности измерения расстояния между двумя объектами. Синим и зелёным эллипсом отмечены распределения положений объектов, оранжевым эллипсом отмечен результат композиции двух распределений

Рассчитав расстояние между центрами объектов по сенсорам робота и оценив его погрешность, а также имея расстояние между объектами с карты, предлагается использовать следующую меру сравнения полученных расстояний:

$$p(D_1, D_2, \sigma D) = e^{-\frac{1}{2} \left(\frac{D_1 - D_2}{\sigma D} \right)^2}. \quad (2.9)$$

Такая мера нормирована на отрезок $[0,1]$ и учитывает погрешность измерений расстояний. Полученная мера сравнения расстояний между соответствующими парами объектов на карте и сцене позволит учитывать погрешность измерений, которая может значительно отличаться для объектов разных классов. Предполагается, что интеграция этой меры в методы распознавания сцен (которые в основном опираются на попарные сравнения расстояний, как будет видно далее) повысит чувствительность методов для близких решений задачи распознавания сцен.

2.1.3. Общий подход с использованием изоморфизма подграфов

Методы на основе поиска изоморфного подграфа в разрезе задачи распознавания сцен интересны тем, что позволяют найти все вхождения одного графа в другой. Поэтому, если представить оба рассматриваемых множества M (2.3) и S (2.2) в виде графов G_M и G_S , которые отражают исходные множества, то

применение алгоритмов поиска изоморфного подграфа позволит решить поставленную задачу (2.4), ища граф сцены G_S как подграф графа карты G_M .

Естественным представлением множеств M и S в виде графов является представление объектов как вершин, а дуг (неориентированных) как отношений между вершинами, характеризующихся расстояниями. Однако критерий изоморфизма графов учитывает только связность соответствующих вершин, т.е. графы $G_1 = \langle V_{G_1}, E_{G_1} \rangle$ и $G_2 = \langle V_{G_2}, E_{G_2} \rangle$ изоморфны друг другу ($G_1 \simeq G_2$), если существует биекция f :

$$f: G_1 \rightarrow G_2: \forall (v_1, v_2) \in E_{G_1}, (f(v_1), f(v_2)) \in E_{G_2}. \quad (2.10)$$

Критерий (2.10) сохраняет попарное взаиморасположение объектов исходных множеств, в то время как для задачи распознавания сцен (2.4) требуется ещё и сохранение классов этих объектов. Это влечет введение дополнительных критериев к критерию изоморфизма (2.10). Таким критерием может выступать разметка графов – сопоставление вершинам и\или ребрам некоторых меток. Семейство алгоритмов поиска изоморфного подграфа VF [Carletti и др., 2019; Cordella и др., 1999; Jüttner, Madarasi, 2018] позволяет в дополнение к (2.10) учитывать и подобную разметку. Таким образом требуется её определить так, чтобы она позволяла решить исходную задачу (2.4).

Так как вершины графов G_S и G_M соответствуют объектам множеств S и M , то это соответствие можно использовать в качестве разметки:

$$G_S = \langle V_S, E_S \rangle, f_{V_S}: V_S \rightarrow S, v \in V_S, f_{V_S}(v) = o^S \in S, \quad (2.11)$$

где G_S – граф сцены, состоящий из множества вершин V_S и множества рёбер E_S , а f_{V_S} – функция разметки, определяемая биекцией множества вершин в S . Аналогичным образом происходит разметка для графа карты G_M .

В соответствие рёбрам данных графов ставятся рассчитанные расстояния между объектами (2.5) и погрешности их определения (2.8), если это граф сцены G_S :

$$f_{E_M}((v_i, v_j) \in E_M) = D(f_{V_M}(v_i), f_{V_M}(v_j)), \quad (2.12)$$

$$f_{E_S}((v_i, v_j) \in E_S) = \langle D(f_{V_S}(v_i), f_{V_S}(v_j)), \sigma D(f_{V_S}(v_i), f_{V_S}(v_j)) \rangle. \quad (2.13)$$

Процедура построения графа сцены представлена в виде псевдокода (Табл. 2.1) и схемы (Рис. 2.3).

Табл. 2.1. Псевдокод алгоритма построения графа сцены G_S

1:	Algorithm build_scene_graph (S) :
2:	-- S – сцена (множество объектов)
3:	init V, E

```

4:   init  $U$ 
5:   for  $o$  in  $S$  do
6:     init  $v$ 
7:      $v.attr = o$  -- разметка (2.11)
8:     add  $v$  to  $V$ 
9:     if  $o.c$  not in  $U$  then
10:      add  $o.c$  to  $U$ 
11:   for  $v1$  in  $V$  do
12:     for  $v2$  in  $V[v1+1:]$  do
13:        $e = (v1, v2)$ 
14:        $D, \sigma D = \text{calc\_distance\_and\_error}(v1, v2)$ 
15:        $e.attr = D, D\sigma$  -- разметка (2.13)
16:       add  $e$  to  $E$ 
17:    $G_S = (V, E)$ 
18:   return  $G_S, U$ 

```

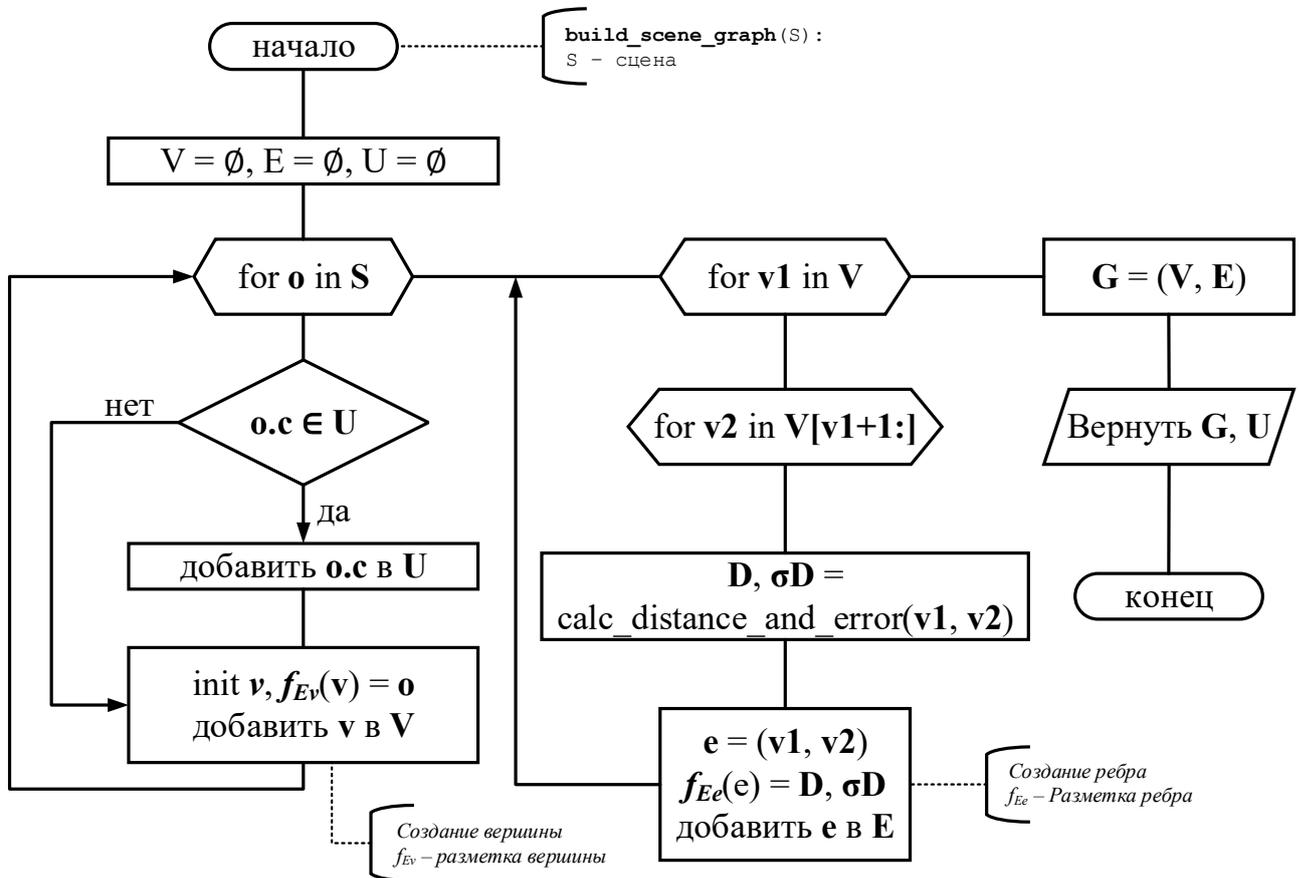


Рис. 2.3. Схема алгоритма построения графа сцены G_S

Алгоритм **build_scene_graph** (Табл. 2.1, Рис. 2.3) принимает на вход описание множество объектов сцены S , возвращает граф сцены G_S и набор уникальных классов U . Функция **calc_distance_and_error** определяет расстояние (2.5) и его среднеквадратичную ошибку (2.8) между двумя объектами на основе

их положений. Вычислительная сложность алгоритма **build_scene_graph** естественным образом оценивается как $O(N^2)$, где N число элементов сцены.

Граф сцены G_S , построенный алгоритмом **build_scene_graph** (Табл. 2.1), является полносвязным, в то время как граф карты G_M , подготовленный для алгоритмов поиска изоморфных подграфов, не обязан быть таковым. Целесообразно рассматривать только те отношения между объектами, что не превышают максимального расстояния в сцене, равно как только объекты тех классов, что присутствуют в сцене. В качестве ограничения расстояния для графа карты можно выбрать удвоенную максимальную рабочую дальность использующегося сенсора. Процедура построения графа карты представлена в виде псевдокода (Табл. 2.2) и схемы ().

Табл. 2.2. Псевдокод алгоритма построения графа карты G_M

1:	Algorithm build_map_graph ($M, U, Dsens$):
2:	-- M - семантическая карта (множество объектов)
3:	-- U - набор уникальных классов сцены
4:	-- $Dsens$ - дальность сенсора
5:	init V, E
6:	for o in M do
7:	if $o.c$ in U then
8:	init v
9:	$v.attr = o$ -- разметка (2.11)
10:	add v to V
11:	for $v1$ in V do
12:	for $v2$ in $V[v1+1:]$
13:	$r = \text{calc_distance}(v1, v2)$
14:	if $r < 2 * Dsens$ then
15:	$e = (v1, v2)$
16:	$e.attr = r$ -- разметка (2.12)
17:	add e to E
18:	$G_M = (V, E)$
19:	return G_M

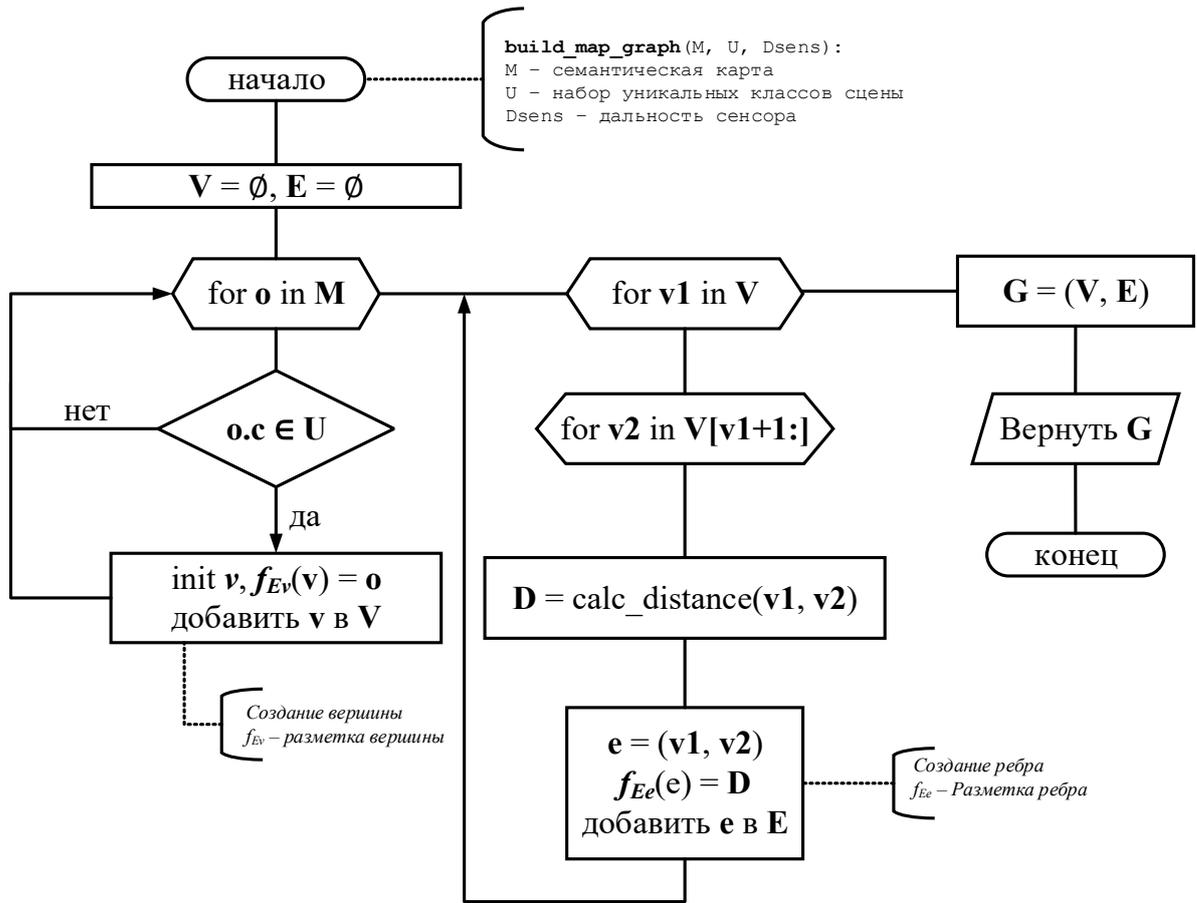


Рис. 2.4. Схема алгоритма построения графа карты G_M

Алгоритм **build_map_graph** (Табл. 2.2, Рис. 2.4) принимает на вход множество объектов карты M , набор уникальных классов объектов сцены U , полученных от алгоритма **build_scene_graph** (Табл. 2.1, Рис. 2.3), максимальную рабочую дальность сенсора $Dsens$. Функция **calc_distance** определяет расстояние между двумя объектами на основе их положения. Алгоритм возвращает граф карты G_M . Вычислительная сложность алгоритма **build_map_graph** оценивается как $O(N^2)$, где N число элементов карты.

Когда построены оба графа и определена их разметка, требуется определить критерии соответствия меток вершин и рёбер для алгоритмов семейства VF. В случае с вершинами такой критерий будет включать в себя соответствие классов с объектов данных вершин:

$$v_i \in V_S, v_j \in V_M: c^{f_{V_S}(v_i)} = c^{f_{V_M}(v_j)} \rightarrow v_i \approx v_j, \quad (2.14)$$

где знаком « \approx » указано, что вершины соответствует друг другу. Таким образом, алгоритм выдаст только такие биекции $f: G_S \rightarrow G_M$ (2.10), которые помимо изоморфизма сохраняют и класс объекта.

Критерии соответствия рёбер графов, а также вопросы численной оценки получившихся биекций (которых может быть более одной) требуют более

детального рассмотрения. Также важным вопросом стоит учёт упомянутых ранее выбросов в данных – отсутствия и наличия лишних объектов в карте и сцене, которые будут нарушать изоморфизм, приводя к потере решений. Далее будут представлены различные варианты решения данных вопросов.

2.1.4. Прямое сравнение расстояний на рёбрах

Критерий сравнения рёбер должен указывать алгоритмам поиска изоморфного подграфа близкие по расстоянию, которое ставится в соответствие ребру (2.12), (2.13). Ввиду наличия погрешностей измерений невозможно требовать точного равенства расстояния на карте и расстояния на сцене. Использование порога для разницы данных расстояний приведёт к проблемам в пограничных случаях (очень маленьких и очень больших расстояниях). Наиболее целесообразно использовать меру p (2.9), которая в зависимости от конкретной погрешности каждого расстояния даст общую меру для всех пар, вне зависимости от их абсолютных значений. Таким образом, критерий сравнения рёбер выглядит следующим образом:

$$e_i \in E_S, e_j \in E_M: p(f_{E_M}(e_i), f_{E_S}(e_j)^1, f_{E_S}(e_j)^2) \geq \tau \rightarrow e_i \simeq e_j, \quad (2.15)$$

где τ настраиваемый порог сходства в диапазоне $[0, 1]$, а $f_{E_S}(e_i)^k$ – k -ый элемент f_{E_S} (2.13). С применением данной меры, алгоритмы поиска изоморфного графа выдадут такие биекции $f: G_S \rightarrow G_M$ (2.10), которые будут учитывать и сходство расстояний между парами объектов, что, в свою очередь, является учетом геометрических особенностей взаиморасположения объектов. Таких биекций (гипотез) может быть не одна, а множество $\{f\}$: это зависит от входных данных и от величины параметра τ . Как неоднократно упоминалось, важно иметь численную оценку данных биекций, чтобы была возможность отобрать лучшие из них. Далее эта численная оценка будет называться коэффициентом уверенности и обозначаться как dc . Для этой задачи также предлагается воспользоваться уже рассчитанной мерой p (2.9) следующим образом:

$$dc(f: G_S \rightarrow G_M) = \frac{1}{|E_S|} \sum_{e_i \in E_S} [p(f_{E_M}(f(e_i)), f_{E_S}(e_j)^1, f_{E_S}(e_j)^2)], \quad (2.16)$$

где биекция f в применении к ребру G_S есть соответствующее ребро в G_M :

$$f(e = (v_i, v_j) \in E_S) = (f(v_i), f(v_j)) \in G_M. \quad (2.17)$$

Коэффициент уверенности dc (2.16) также нормирован на отрезок $[0, 1]$, что делает его удобным в дальнейшем использовании. Алгоритм распознавания сцен

на основе поиска изоморфного подграфа с прямым сравнением меток рёбер представлен в виде псевдокода (Табл. 2.3) и схемы (Рис. 2.5).

Табл. 2.3. Псевдокод алгоритма распознавания сцены на основе поиска алгоритмов изоморфного подграфа с прямым сравнением расстояний на рёбрах

```

1: Algorithm sig(S, M, Dsens,  $\tau$ , dc_min):
2:   -- S - сцена (множество объектов)
3:   -- M - семантическая карта (множество объектов)
4:   --  $\tau$  - порог сходства отношений
5:   -- Dsens - дальность сенсора
6:   -- dc_min - порог степени уверенности
7:   GS, U = build_scene_graph(S) -- Табл. 2.1
8:   GM = build_map_graph(M, U, Dsens, k) -- Табл. 2.2
9:   function cmpV(vM, vS)
10:    if vM.attr[0] == vS.attr[0] then
11:      return True
12:   function cmpE(eM, eS)
13:     $p = \mathbf{P}(eM.attr, eS.attr[0], eS.attr[1])$  -- (2.9)
14:    if  $p > \tau$  then
15:      return True
16:    return False
17:   F = VF(GM, GS, cmpV, cmpE)
18:   init R
19:   for f in F do
20:     dc = calc_dc (f) -- (2.16)
21:     if dc > dc_min then
22:       add (f, dc) to R
23:   sort R by dc
24:   return R

```

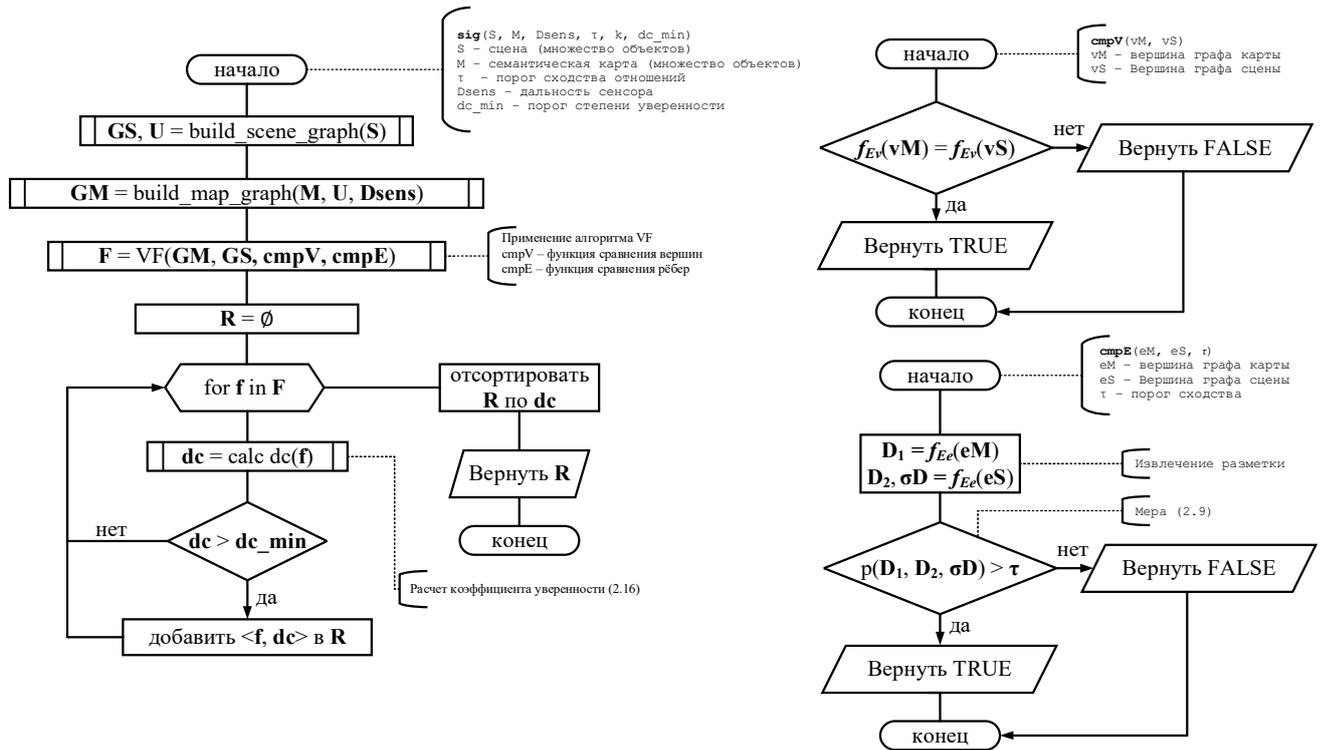


Рис. 2.5. Схема алгоритма распознавания сцены на основе поиска алгоритмов изоморфного подграфа с прямым сравнением расстояний на рёбрах

Алгоритм **sig** (Табл. 2.3, Рис. 2.5) принимает на вход множество объектов карты M и множество объектов сцены S , максимальную рабочую дальность сенсора $Dsens$, порог сходства отношений τ , порог степени уверенности dc_min . Функции **build_scene_graph** (Табл. 2.1) и **build_map_graph** (Табл. 2.2) формируют соответственные графы, функция **P** выполняет расчет согласно (2.9), функция **VF** выполняет поиск изоморфного подграфа с указанием критериев сходства вершин и рёбер, функция **calc_dc** рассчитывает коэффициент уверенности согласно (2.16). Алгоритм возвращает отсортированный массив соответствий $f:S \rightarrow M$ с указанием их коэффициентов уверенности. Поскольку алгоритм **sig** использует алгоритм **VF**, то его вычислительная сложность также в худшем случае экспоненциальная, поскольку задача поиска изоморфного подграфа относится к классу NP-полных. Однако эвристики в виде разметки графа, а также понижение числа ребер в графах на практике позволяют существенно сократить время работы.

Результаты работы алгоритма **sig**, в том числе и временные характеристики, приведены в четвёртой главе.

2.1.5. Кластеризация отношений

Альтернативным подходом к сравнению отношений может выступать не прямое сравнение (2.15), а присвоение отношениям некоторых меток классов, подобно тому, как это было сделано для вершин. Переход к сравнению в дискретном множестве может благотворно сказаться на вычислительной эффективности. Получить эти метки классов можно на основе кластеризации набора отношений. Предлагается рассчитать все попарные отношения в множестве объектов сцены S , разбив их по типам соединяемых отношениями объектов, например, «столб-столб», «столб-светофор» и т.п. Таким образом будет сформировано множество S' , которое состоит из кортежа идентификатора, описывающего типы объектов, и множества расстояний D (2.5) и погрешностей σD (2.8):

$$S' = \{\langle id(c_i, c_j), \{\langle D, \sigma D \rangle\} \rangle\}, \quad (2.18)$$

$$id(c_i, c_j) \in \mathbb{N}; id(c_i, c_j) = id(c_j, c_i); id(c_i, c_j) \neq id(c_i, c_m), c_j \neq c_m.$$

Для каждого элемента множества предлагается провести кластеризацию иерархическими методами [Жамбю, Миркин, 1988], которые не разбивают множество на заданное число кластеров, а выстраивают иерархию, пользуясь порогом схожести элементов, в роли которого может выступать мера (2.9). Поскольку расстояния сцены снабжены ещё и погрешностями (2.8), то порог для кластеризации выбирается следующий:

$$p(r_1, r_2, \sigma r_1 + \sigma r_2) \geq \tau_c. \quad (2.19)$$

Применив алгоритм иерархической кластеризации к элементам S' , будет получено множество S'' :

$$S'' = \{\langle id(c_i, c_j), \{\{\langle D, \sigma D \rangle\}_m\} \rangle\}, \quad (2.20)$$

где m указывает на номер кластера, значения id и m предлагается использовать в качестве разметки рёбер графа сцены:

$$\bar{f}_{E_S}((v_i, v_j) \in E_S) = \langle id(c_i, c_j), m \rangle. \quad (2.21)$$

Перед тем, как определить критерий сходства, требуется описать похожую разметку для графа карты G_M . Т.к. идеологически требуется, чтобы оба графа использовали общую разметку, то будет выполняться классификация рёбер графа карты на основании полученного множества S'' (2.20). Сравнение будет происходить с центроидами кластеров всех отношений карты метрикой (2.9). Однако в процессе сравнения может возникнуть ситуация, когда отношение карты удовлетворяет нескольким получившимся кластерам. Поэтому вместо взятия лучшего значения предлагается для рёбер карты организовать разметку таким

образом, что в ней будут указаны все подходящие кластеры S'' под порог классификации τ_m :

$$\bar{f}_{E_M}((v_i, v_j) \in E_M) = \langle id(c_i, c_j), \{m_k\}, p(r_{ij}^M, \overline{\{D\}_{m_k}}, \overline{\{\sigma D\}_{m_k}}) \rangle < \tau_m. \quad (2.22)$$

В качестве погрешности в формуле (2.22) берётся среднее всех погрешностей элементов кластера. Те рёбра, у которых множество номеров кластеров оказалось пустым, удаляются с графа карты. Тогда критерий сравнения рёбер принимает вид:

$$e_i \in E_S, e_j \in E_M: id^{e_i} = id^{e_j}, m^{e_i} \in \{m\}^{e_j} \rightarrow e_i \simeq e_j, \quad (2.23)$$

Что позволяет учесть и сходство типов отношений, и наличие номера кластера отношения сцены в множестве типов кластеров карты. Учитывая вышесказанное, алгоритм построения графа сцены принимает вид:

Табл. 2.4. Псевдокод алгоритма построения графа сцены с кластеризацией отношений

1:	Algorithm build_scene_graph_cluster (S, τ_c):
2:	-- S - сцена (множество объектов)
3:	-- τ_c - порог кластеризации
4:	init V, E
5:	init U
6:	init S'
7:	for o in S do
8:	init v
9:	$v.attr = o$ -- разметка (2.11)
10:	add v to V
11:	for $o2$ in $S[o+1:]$ do
12:	$r, \sigma = \text{calc_distance_and_error}(o, o2)$
13:	add (r, σ) to $S'[id(o.c, o2.c)]$
14:	init S''
15:	for id, s' in S' do
16:	$s'' = \text{h_cluster}(s', \tau_c)$
17:	add s'' to $S''[id]$
18:	for $v1$ in V do
19:	for $v2$ in $V[v1+1:]$ do
20:	$e = (v1, v2)$
21:	$id = \text{id}(v1.attr.c, v2.attr.c)$
22:	$e.attr = (id, \text{get_cluster_no}(S'', e))$ -- (2.21)
23:	add e to E
24:	$G_S = (V, E)$
25:	return G_S, S''

Алгоритм **build_scene_graph_cluster** (Табл. 2.4) принимает на вход множество объектов сцены S и порог кластеризации τ_c (2.19). Функция **calc_distance_and_error** определяет расстояние (2.5) и его среднеквадратичную

ошибку (2.8) между двумя объектами на основе их положений. Функция **id** рассчитывает тип отношения согласно (2.18), функция **h_cluster** выполняет иерархическую кластеризацию по порогу, функция **get_cluster_no** возвращает номер кластера для отношения. Возвращает алгоритм граф сцены G_S и множество S'' . Вычислительную сложность алгоритма **build_scene_graph_cluster** можно оценить, как $O(N^2)$, где N – количество объектов в сцене. Использование алгоритмов иерархической кластеризации не ухудшает вычислительную сложность, поскольку их собственная сложность $O(N^2 \log N)$.

Полученное множество S'' используется алгоритмом построения графа карты:

Табл. 2.5. Псевдокод алгоритма построения графа карты с кластеризацией отношений

1:	Algorithm build_map_graph_cluster (M, S'', τ_m):
2:	-- M – семантическая карта (множество объектов)
3:	-- S'' – множество кластеров сцены
4:	-- τ_c – порог кластеризации
5:	init V, E
6:	for o in M do
7:	if $o.c$ in S'' then
8:	init v
9:	$v.attr = o$ -- разметка (2.11)
10:	add v to V
11:	for $v1$ in V do
12:	for $v2$ in $V[v1+1:]$
13:	$r = \text{calc_distance}(v1, v2)$
14:	$id = \text{id}(v1.attr.c, v2.attr.c)$
15:	for c_id, s'' in S'' do
16:	if $id == c_id$ then
17:	for $cluster_no, meanD, mean\sigma D$ in s'' then
18:	$p = \mathbf{P}(r, meanD, mean\sigma D)$
19:	if $p < \tau_m$ then
20:	$e = (v1, v2)$
21:	if e in E then
22:	add $cluster_no$ to $e.attr$
23:	else
24:	add e to E
25:	$e.attr = (id, \{cluster_no\})$
26:	$G_M = (V, E)$
27:	return G_M

Алгоритм **build_map_graph_cluster** (Табл. 2.5) принимает на вход множество объектов карты M , множество кластеров сцены S'' , полученных от алгоритма **build_scene_graph_cluster** (Табл. 2.4), и порог классификации τ_m .

Функция **calc_distance** определяет расстояние между двумя объектами на основе их положения. Функция **id** рассчитывает тип отношения согласно (2.18), функция **P** соответствует (2.9). Алгоритм возвращает граф карты G_M . Вычислительная сложность оценивается как $O(N^2MK)$, где N – число элементов карты, M – число типов отношений, K – число кластеров в каждом типе.

Алгоритм распознавания сцены с данными модификациями тогда принимает вид:

Табл. 2.6. Псевдокод алгоритма распознавания сцены на основе поиска изоморфного подграфа с кластеризацией отношений

1:	Algorithm sig_cluster ($S, M, Dsens, \tau_c, \tau_m, dc_min$):
2:	-- S – сцена (множество объектов)
3:	-- M – семантическая карта (множество объектов)
4:	-- τ_c – порог кластеризации
5:	-- τ_m – порог классификации
6:	-- dc_min – порог степени уверенности
7:	$GS, S'' = \text{build_scene_graph_cluster}(S, \tau_c)$ -- Табл. 2.4
8:	$GM = \text{build_map_graph_clsuetr}(M, S'', \tau_m)$ -- Табл. 2.5
9:	function compare_vertices (vM, vS)
10:	if $vM.attr[0] == vS.attr[0]$ then
11:	return <i>True</i>
12:	function compare_edges (eM, eS)
13:	$idS, cluster = eS.attr$
14:	$idM, clusters = eM.attr$
15:	if $idS == idM$ then
16:	if $cluster \in clusters$ then
17:	return <i>True</i>
18:	return <i>False</i>
19:	return <i>False</i>
20:	$F = \mathbf{VF}(GM, GS, \text{compare_vertices}, \text{compare_edges})$
21:	init R
22:	for f in F do
23:	$dc = \text{calc_dc}(f)$ -- (2.16)
24:	if $dc > dc_min$ then
25:	add (f, dc) to R
26:	sort R by dc
27:	return R

Алгоритм **sig_cluster** (Табл. 2.6) принимает на вход описание множеств карты M и сцены S , пороги кластеризации и классификации, а также порог коэффициента уверенности. Функция **VF** реализует поиск изоморфного подграфа. Алгоритм возвращает отсортированный массив соответствий $f:S \rightarrow M$ с указанием их коэффициентов уверенности. Вычислительная сложность этого алгоритма аналогично **sig** (Табл. 2.3) оценивается в худшем случае как $O(e^N)$.

К сожалению, некоторые реализации алгоритмов семейства VF не поддерживают произвольные критерии сходства рёбер, а могут работать только с раскраской графов – вариантом разметки, где каждому ребру сопоставляется лишь один идентификатор. Для таких реализаций множество номеров кластеров в (2.22) заменяется на взятие номера лучшего кластера.

Подобная модификация метода потенциально позволит ускорить процедуру распознавания, т.к. единожды размечает граф, не требуя в дальнейшем многократных сравнений одних и тех же рёбер, как это делается в (2.15). Однако из-за использования центроидов вместо оригинальных расстояний может быть потеряна часть информации. Экспериментальное изучение данных вопросов приведено в четвёртой главе работы (раздел 4.1).

2.1.6. Мягкое расширение метода

Из-за ошибок в системах распознавания на сцене могут быть распознаны лишние объекты, которые не представлены на семантической карте. Также такая ситуация может произойти ввиду изменчивости окружающей среды. Если на графе сцены присутствуют такие лишние объекты, то метод поиска изоморфных подграфов не найдёт решения, т.к. не будет соответствия этому объекту на графе сцены. Чтобы решить данную проблему, было предложено мягкое расширение предложенного алгоритма. Основная идея состоит в добавлении на граф карты так называемых фиктивных вершин. Их цель – соответствовать лишним объектам на графе сцены, не нарушая общий изоморфизм. Добавляется фиксированное число фиктивных объектов для каждого класса, представленного в сцене. Число таких объектов может варьироваться и зависит от качества системы распознавания и от того, как много ожидается ложных срабатываний на кадрах. Чтобы сохранять изоморфизм, к фиктивным вершинам проводятся рёбра, также называемые фиктивными. Разметка (2.11) для фиктивных вершин — это также класс объекта и метка фиктивности (ложь\истина), а разметка (2.12) для фиктивных рёбер – только метка фиктивности. Алгоритм построения графа карты для мягкого расширения представлен в виде псевдокода (Табл. 2.7) и схемы ().

Табл. 2.7. Псевдокод алгоритма построения графа карты G_M для мягкого расширения

1:	Algorithm build_map_graph_soft ($M, U, Dsens, k$):
2:	-- M – семантическая карта (множество объектов)
3:	-- U – набор уникальных классов сцены
4:	-- $Dsens$ – дальность сенсора

```

5:      -- k - число фиктивных объектов
6:      init  $V, E$ 
7:      for  $o$  in  $M$  do
8:          if  $o.c$  in  $U$  then
9:              init  $v$ 
10:              $v.attr = o$  -- разметка (2.11)
11:              $v.fake = False$  -- метка фиктивности
12:             add  $v$  to  $V$ 
13:         for  $c$  in  $U$  do
14:             for  $i$  in  $0:k$  do
15:                 init  $v$ 
16:                  $v.attr = c$  -- измененная разметка
17:                  $v.fake = True$  -- метка фиктивности
18:                 add  $v$  to  $V$ 
19:             for  $v1$  in  $V$  do
20:                 for  $v2$  in  $V[v1+1:]$ 
21:                     if  $v1.fake$  OR  $v2.fake$  then
22:                          $e = (v1, v2)$ 
23:                          $e.fake = True$  -- метка фиктивности
24:                         add  $e$  to  $E$ 
25:                     else
26:                          $r = \text{calc\_distance}(v1, v2)$ 
27:                         if  $r < 2*Dsens$  then
28:                              $e = (v1, v2)$ 
29:                              $e.attr = (r, \sigma)$  -- разметка (2.12)
30:                              $e.fake = False$  -- метка фиктивности
31:                             add  $e$  to  $E$ 
32:              $G_M = (V, E)$ 
33:         return  $G_M$ 

```

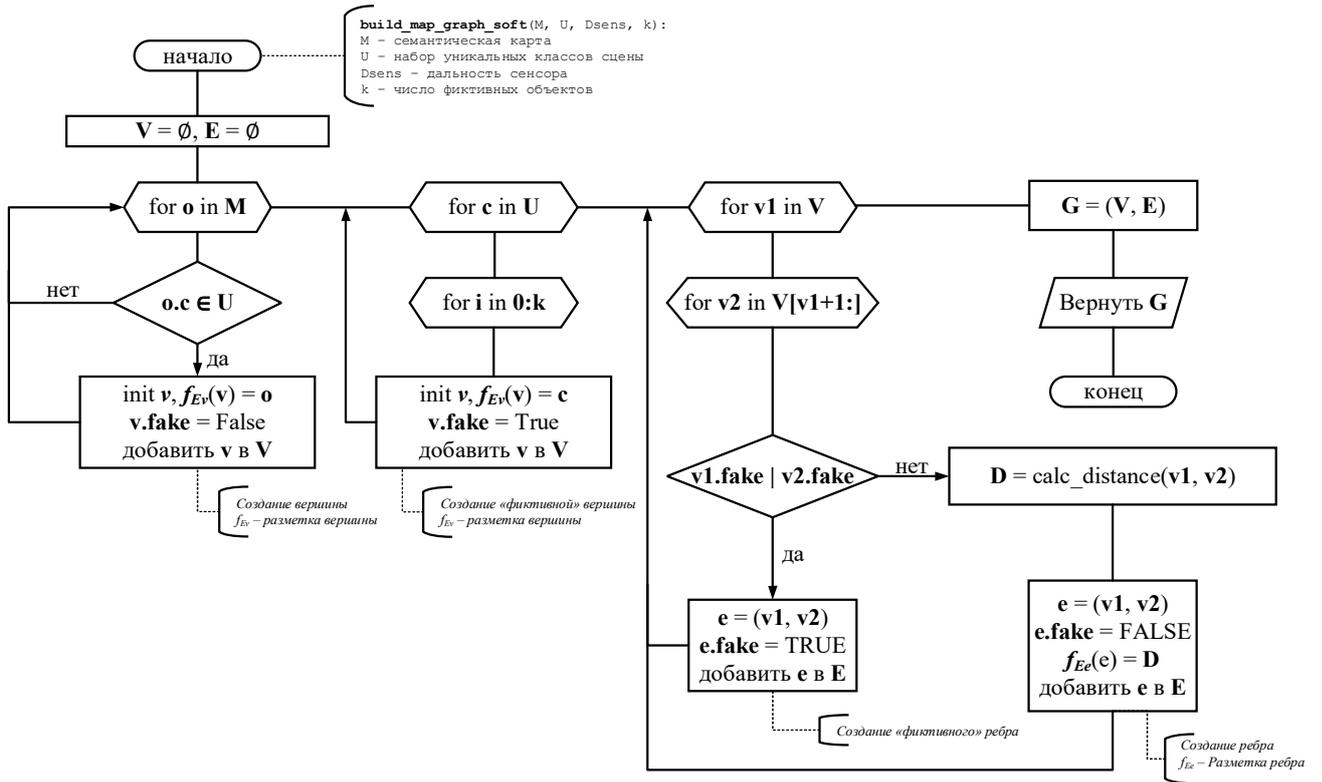


Рис. 2.6. Схема алгоритма построения графа карты GM для мягкого расширения

Вход и выход алгоритма **build_map_graph_soft** (Табл. 2.7, Рис. 2.6), а также внутренние функции соответствуют алгоритму **build_map_graph** (Табл. 2.2, Рис. 2.4), за тем исключением, что на вход добавляется число фиктивных объектов k . Вычислительная сложность алгоритма **build_map_graph_soft** оценивается как $O(N^2)$.

При такой модификации графа G_M в ответы $\{f\}$ будут попадать фиктивные вершины и рёбра, вклад которых требуется учесть при расчёте коэффициента уверенности. Предлагается в (2.16) выбирать значение функции p (2.9) равным нулю, если хоть один из её аргументов является ребром с фиктивной меткой. Такой подход позволит понизить общую уверенность в ответах с фиктивными элементами, но при этом сохранить решения, которые могут быть потенциально утрачены ввиду выбросов во входных данных. Алгоритм мягкого расширения метода поиска на основе изоморфного подграфа представлен в виде псевдокода (Табл. 2.8) и схемы ().

Табл. 2.8. Псевдокод мягкого расширения алгоритма распознавания сцен на основе поиска изоморфных подграфов

1:	Algorithm sig_soft ($S, M, Dsens, \tau, k, dc_min$):
2:	-- S - сцена (множество объектов)

```

3:      -- M - семантическая карта (множество объектов)
4:      --  $\tau$  - порог сходства отношений
5:      -- Dsens - дальность сенсора
6:      -- k - число фиктивных объектов
7:      -- dc_min - порог степени уверенности
8:      GS, U = build_scene_graph(S)
9:      GM = build_map_graph_soft(M, U, Dsens, k)
10:     function compare_vertices(vM, vS)
11:         if vM.attr[0] == vS.attr[0] then
12:             return True
13:     function compare_edges(eM, eS)
14:         if eM.fake then
15:             return True
16:         p = P(eM.attr, eS.attr[0], eS.attr[1]) -- (2.9)
17:         if p >  $\tau$  then
18:             return True
19:         return False
20:     F = VF(GM, GS, compare_vertices, compare_edges)
21:     init R
22:     for f in F do
23:         dc = calc_dc(f) -- (2.16)
24:         if dc > dc_min then
25:             remove fakes from f
26:             add (f, dc) to R
27:     sort R by dc
28:     return R

```

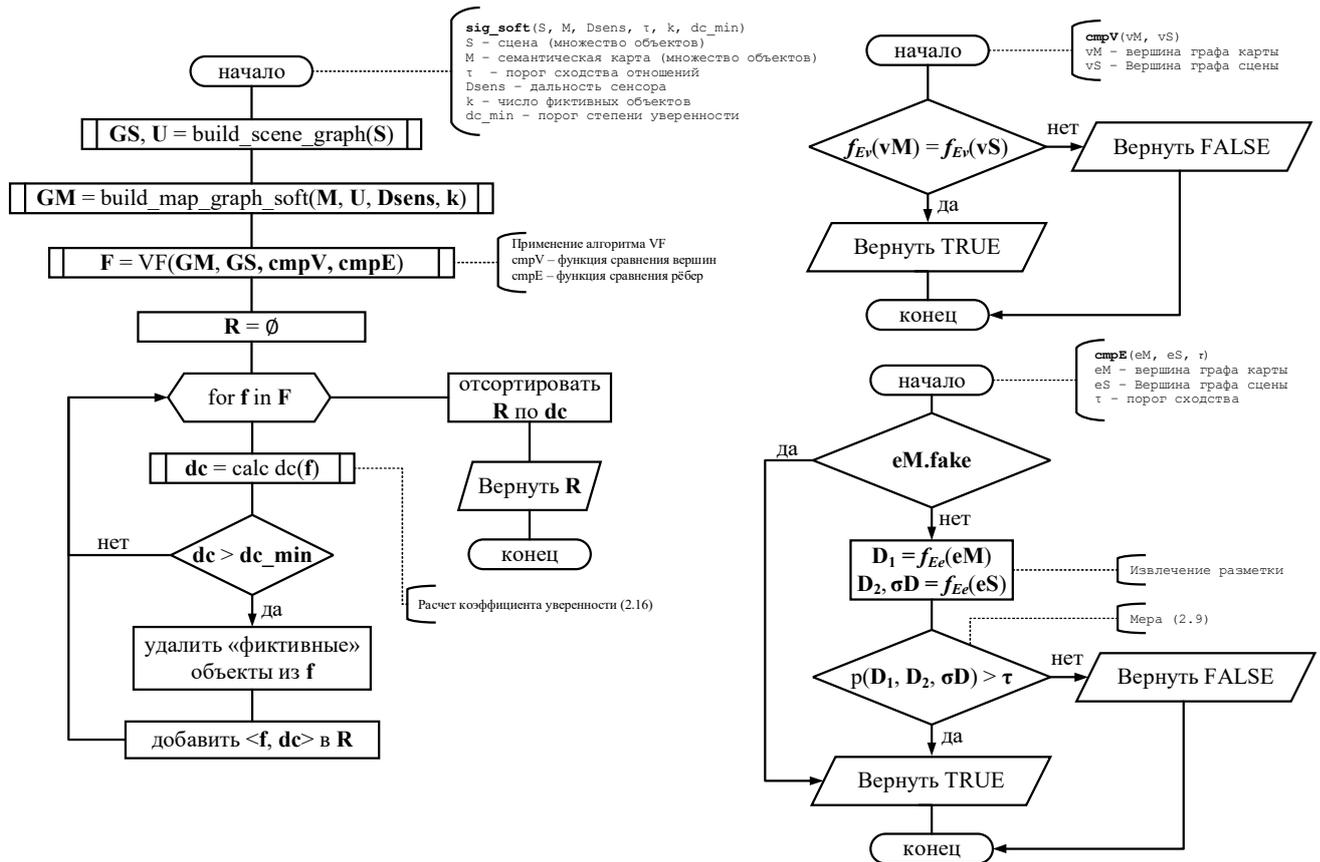


Рис. 2.7. Схема мягкого расширения алгоритма распознавания сцен на основе поиска изоморфных подграфов

Вход и выход алгоритма **sig_soft** (Табл. 2.8, Рис. 2.7) аналогичны алгоритму **sig** (Табл. 2.3), за исключением числа фиктивных вершин k . В 25 строчке алгоритма **sig_soft** происходит очистка из биекции f соответствий, связанных с фиктивными вершинами графа, таким образом, для каждого объекта сцены имеется сопоставления с картой. Аналогично предыдущим алгоритмам, использующих в себе алгоритмы VF, вычислительная сложность в худшем случае экспоненциально зависит от числа элементов в большем графе.

Предложенное расширение исходного алгоритма позволяет находить решения в условиях выбросов во входных данных, однако увеличивает время исполнения, что подтверждается проведенными экспериментами (подраздел 3.1.4 и раздел 4.1).

2.1.7. Учёт визуального сходства

Разработанный алгоритм распознавания сцен (Табл. 2.8) позволяет учитывать геометрические особенности взаиморасположения объектов на сцене и находить группы объектов с такими же особенностями, а также учитывать их семантические классы. Однако в реальном мире возможны ситуации, когда разные группы объектов сформируют похожие композиции, как, например, показано на Рис. 2.8.



Рис. 2.8. Две сцены, по которым будет построены графы одинакового вида, изображенного справа (ряд рёбер был опущен для наглядности)

Поскольку сцены на Рис. 2.8 с семантической точки зрения идентичные, то при решении задачи распознавания сцены может быть найден неверный ответ, что в дальнейшем приведет к ошибкам в локализации. Для разрешения подобных ситуаций предлагается использовать методы оценки визуального сходства. Для этого можно применить базисную визуально-языковую модель CLIP [Radford и др., 2021] (или её аналогам), которая по изображению строит вектор признаков (embedding, feature vector). Предлагается использовать только кодировщик, работающий с изображениями, кодировщик для текста в данном применении не используется. Визуальный образ объектов I (2.1) можно представить в виде единичного изображения (для сцены) или множества таковых, сделанных, например, с разных ракурсов. Таким образом, построить вектор признаков h для объекта o можно при помощи функции H , отражающей работу модели CLIP:

$$h(o) = \frac{\sum_{i \in I^o} \|H(i)\|}{|I^o|}. \quad (2.24)$$

В формуле (2.24) используется нормализация полученных векторов и дальнейшее усреднение по всем элементам визуального образа. Это возможно

сделать, т.к. обучение CLIP происходит с использованием косинусного расстояния u в составе функции потерь, которое линейно по своим аргументам:

$$u(h_1, h_2) = \frac{h_1 \cdot h_2}{\|h_1\| \|h_2\|}. \quad (2.25)$$

Эта мера также нормирована на отрезок $[0,1]$ и может быть использована для сравнения визуального сходства объектов сцены с объектами карты. Целесообразно сделать это на этапе расчёта коэффициента уверенности, когда уже получены гипотезы, отражающие геометрические особенности. Формула (2.16) примет вид:

$$dc(f) = \frac{1}{|E_S|} \sum_{(v_i, v_j) \in E_S} [u(H(o_i^S), H(o_i^M)) \cdot u(H(o_j^S), H(o_j^M)) \cdot p(D(o_i^S, o_j^S), D(o_i^M, o_j^M), \sigma D(o_i^S, o_j^S))] , o_i^S = f_{V_S}(v_i), o_i^M = f_{V_M}(f(v_i)). \quad (2.26)$$

Интегрировав полученную формулу в 23 строку алгоритма **sig_soft** (Табл. 2.7), будет получен алгоритм, способный выполнять задачу распознавания сцен с учётом как геометрических особенностей взаиморасположения объектов, так и визуального сходства. При этом он будет работать в условиях выбросов во входных данных и генерировать набор гипотез, снабжая их численной оценкой. При расчёте коэффициента уверенности (2.26) в случае алгоритма мягкого расширения, функции u (2.25) также, как ранее p (2.9), выбираются равными нулю, если хотя бы один аргумент является фиктивным объектом. Схематично предложенный метод изображен на Рис. 2.9.

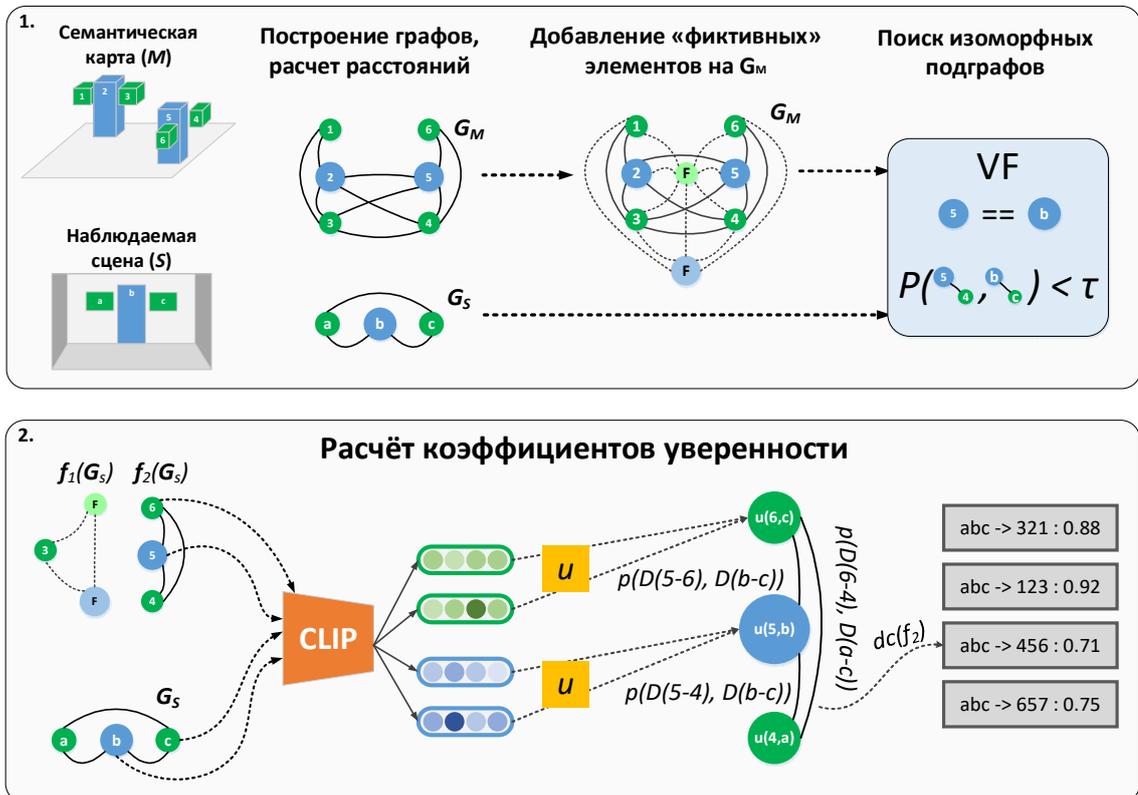


Рис. 2.9. Схема метода распознавания сцен на основе поиска изоморфного подграфа с учётом визуального сходства объектов моделью CLIP и мягким расширением

Полученное расширение увеличивает время выполнения алгоритма в сравнении с базовым подходом (**sig**, Табл. 2.3) за счёт того, что происходят дополнительные вычислительные операции, связанные с получением и сравнением векторов признаков (2.25). Однако расширение позволит увеличить точность работы благодаря учёту схожести объектов. Также стоит отметить, что требуется создание семантической карты с объектами, у которой, помимо положений и семантических меток, есть информация в виде CLIP-векторов. Численные результаты сравнения алгоритмов с и без учета визуального сходства приведены в разделе 4.1.

2.2. Недоопределённая модель локализации

В разделе формализуется аппарат Недоопределённых моделей [Нариньяни, 2007] в применении к задаче локализации по визуальным ориентирам. Используемый аппарат позволяет получать решение в виде ограничений на положение робота, представленных интервальными оценками, что предоставляет возможность использовать их как начальные данные для методов локализации по дальномерным показаниям.

Пояснение. Несмотря на общемировое развитие смежных с Недоопределёнными вычислениями направлений, включая программирование в ограничениях и интервальный анализ, в диссертации в дань традиции отечественной школе искусственного интеллекта будет использоваться терминология А.С. Нариньяни.

2.2.1. Постановка задачи локализации по ориентирам

Общая формальная постановка задачи определения положения по ориентирам состоит в том, что имеется набор ориентиров L , где каждый ориентир задан своим положением в пространстве l . В зависимости от задачи, положение может быть как двумерным, так и трехмерным, а также содержать ориентацию ориентира. Рассмотрим двумерный случай без ориентации, где:

$$L = \{l = \langle x_l, y_l \rangle\}. \quad (2.27)$$

Также имеется набор измерений R сенсорами робота, который, в свою очередь, зависит от условий задачи. Набор R обычно должен содержать расстояние до ориентира и/или углы на ориентир, но опять же, возможны случаи, когда сенсорика также способна определить ориентацию ориентира. Для двумерного случая без ориентации измерения будут задаваться расстоянием и углом:

$$R = \{ \langle r_l, a_l \rangle \}. \quad (2.28)$$

Измерения R (2.27) выполнены в локальных координатах робота, в то время как положение ориентиров L (2.28) в глобальных координатах карты. Измерения R естественно переводятся в положение ориентира в локальных координатах L_R :

$$L_R = \{ \langle r_l \cos(a_l), r_l \sin(a_l) \rangle \}. \quad (2.29)$$

Требуется найти такое преобразование T , заданное матрицей, для которого среднеквадратичная ошибка по расстояниям между L (2.27) и TL_R (2.29) минимальна:

$$\min_T \left(\overline{(L - TL_R)^2} \right). \quad (2.30)$$

Условие минимизации (2.30) также может варьироваться в зависимости от задачи, например, учитывать дополнительные измерения, принимать во внимание погрешности измерений и т.п.

Как упоминалось в разделе 1.3, такая задача обычно решается методами глобальной оптимизации и вероятностными подходами, однако в данной работе предлагается использовать подход, основанный на Недоопределённых моделях (Н-моделях, здесь и далее приставка «Н-» перед словом, будет означать «недоопределённая/ое/ые»). Этот подход обладает рядом свойств, важных для решаемой в исследовании задачи локализации.

2.2.2. Описание Н-моделей

Н-модели были предложены отечественным ученым А.С. Нариньяни [Нариньяни, 1986; Нариньяни, 2007], как универсальный метод программирования в ограничениях. Сформировано понятие Н-модели, задающей связи между элементами. Искомые (неизвестные величины) описываются Н-переменными, которые могут принимать различные типы Н-расширений [Telerman, Ushakov, 1996]. Вычислительная процедура основана на применении Н-операций, которые обновляют одни Н-переменные за счёт других. Формально обобщенная вычислительная Н-модель \mathcal{M} задаётся четвёркой:

$$\mathcal{M} = (V, W, C, R), \quad (2.31)$$

где V – множество объектов из заданной предметной области, R – множество ограничений на значения объектов из V , W – множество функций присваивания, C – множество функций проверки корректности. Объектам v из V (2.31) ставится в соответствие полное множество принимаемых значений (универсальное множество), начальное значение внутри этого универсума, функция присваивания W_v и функция проверки корректности C_v . Функции присваивания обновляют значение N -переменной на основе старого и нового значения, функция проверки корректности определяет правильность полученной переменной. Важным элементом является множество ограничений R (2.31), где каждое ограничение должно интерпретироваться функционально. На Рис. 2.10 изображено представление N -модели в виде ориентированного двудольного графа, где имеются функциональные (R_1, R_2) и объектные вершины (v_1, v_2).

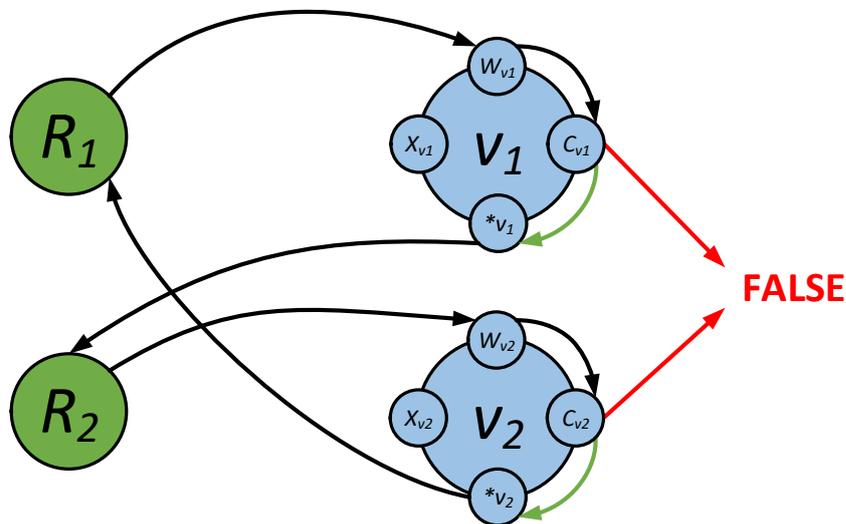


Рис. 2.10. Схематичное изображение N -модели \mathcal{M} (2.31) в виде двудольного графа

На Рис. 2.10 стрелка, направленная от объектной вершины в функциональную, показывает, что эта переменная идёт на вход данной функции. Стрелка, направленная от функциональной вершины к объектной, показывает, что эта функция обновляет значения N -переменной. При обновлении значений N -переменной последовательно применяются функции присваивания W и проверки корректности C . Если функция проверки корректности C выдаёт ошибку, то процесс вычисления заканчивается неудачей. В противном случае, и, если значение N -переменной изменилось, то активируются все ограничения R , входом для которых является эта N -переменная. Когда все функции R были активированы, процедура завершается успешно. Алгоритм процесса вычисления N -модели, используемый в данной диссертации, представлен ниже в виде псевдокода:

Табл. 2.9. Алгоритм вычислительной процедуры Н-модели

1.	Algorithm <code>sdm_process</code> (<i>V</i> , <i>R</i> , <i>W</i> , <i>C</i>):
2.	-- V - Н-переменные
3.	-- R - функции интерпретации
4.	-- W - функция присваивания
5.	-- C - функция проверки корректности
6.	<code>active_R = R</code>
7.	while <code>active_R ≠ ∅</code> do
8.	<code>f = active_R[0]</code>
9.	remove <code>f</code> from <code>active_R</code>
10.	<code>v_out = output var of f from V</code>
11.	<code>v_in = input vars of f from V</code>
12.	<code>v_new = f(v_in)</code>
13.	<code>w = W(v_out)</code>
14.	<code>v_upd = w(v_out, v_new)</code>
15.	if <code>v_upd ≠ v_out</code> then
16.	<code>c = C(n_out)</code>
17.	if <code>c(v_upd)</code> then
18.	<code>v_out = v_upd</code>
19.	for <code>v</code> in <code>v_in</code> do
20.	for <code>f</code> in <code>R</code> do
21.	if <code>f</code> not in <code>active_R</code> then
22.	if <code>v</code> input for <code>f</code> then
23.	add <code>f</code> in <code>active_R</code>
24.	else
25.	return <code>False, V</code>
26.	return <code>True, V</code>

Алгоритм `sdm_process` (Табл. 2.9) принимает на вход параметры Н-модели (2.31), возвращает статус вычислений и обновлённые значения Н-переменных. Задача нахождения области значений функций от многих переменных в ограничениях традиционно оценивается как NP-сложная [Шарый, 2010] и имеет экспоненциальную зависимость от объема входных данных. Однако вид и характер ограничений, а также размер стартовых значений могут приводить к ускорению решения конкретных практических задач вплоть до полиномиального времени. Также стоит отметить, что на практике может использоваться другой критерий остановки (строка 3 алгоритма `sdm_process`), например достижение заданной точности переменных или просто ограничения числа итераций.

Таким образом, Н-модели представляют собой универсальную вычислительную процедуру, оперирующую связями между сущностями. Набор связей фактически произволен и задаётся в зависимости от решаемой задачи, наличия данных и т.д. Использование функций проверки корректности даёт возможность выявить противоречия во входных данных, позволяя отсеять неправильные гипотезы, если таковые имеются. Также важно отметить, что

задачи локализации мобильных роботов решаются в условиях наличия погрешностей, которые как раз учитываются Н-моделями в виде задания областей неопределённости. Все указанные преимущества данного аппарата делают его привлекательным для применения в локализации мобильных роботов по визуальным ориентирам, формируемым в процессе решения задачи распознавания сцен.

2.2.3. Общие принципы формирования Н-модели для задачи локализации

Одним из плюсов построения Н-моделей является то, что в них нет явного указания, что является входными данными, а что выходными. Указав связи, можно решать как прямую, так и обратную задачу. В задаче локализации робота по визуальным ориентирам есть связи между положениями ориентиров и положением робота, задаваемые показаниями сенсоров. Таким образом, если известны положения ориентиров, то задача локализации является прямой, а если известно положение робота, то задача картирования является обратной. Однако рассмотрим только прямую задачу, и зафиксируем положение ориентиров, тогда для двумерного случая положение робота будет задаваться тремя Н-переменными: $*X$, $*Y$, $*\theta$ – здесь и далее, «*» слева от имени переменной будет означать, что она является недоопределённой. Введённые Н-переменные, соответственно, включают в себя истинное положение робота – X , Y , θ . Эти Н-переменные образуют множество V (2.31). В качестве Н-расширений предлагается использовать мульти-интервалы – упорядоченные, непересекающиеся отрезки из множества действительных чисел:

$$*a = \left[[a_{low}^1, a_{high}^1], \dots [a_{low}^n, a_{high}^n] \right], a_{low}^i \setminus a_{high}^i \in \mathbb{R}, a_{low}^i \leq a_{high}^i, a_{high}^i < a_{low}^{i+1}. \quad (2.32)$$

Представление в виде мульти-интервала наиболее удобное из доступных Н-расширений для использования в задачах локализации. Координатные Н-переменные $*X$ и $*Y$ могут быть заданы на всей оси действительных чисел, однако на практике наибольшее значение, принимаемое ими, сводится к размеру карты или рабочей области робота. Угловая Н-переменная $*\theta$ принимает наибольшее значение $[-\pi, \pi]$, при этом для угловых Н-переменных важно соблюдать особенности, связанные с π -переходом. Например, задать половину окружности можно как от $-\pi/2$ до $\pi/2$, так и наоборот. Однако, если в первом случае Н-

переменная примет значение $[-\pi/2, \pi/2]$, то во втором, чтобы удовлетворять условиям (2.32), значение будет задаваться как $[-\pi, -\pi/2], [\pi/2, \pi]$.

Также важно отметить, как будут задаваться функции, интерпретирующие ограничения R (2.31) – т.н. функции интерпретации. Если есть функция в классическом представлении, задающая связь одной переменной от других, например $X = f(Y, \theta)$, то функция интерпретации $*f$ в применении к Н-переменной $*X$ будет образом f на множестве декартова произведения $*Y$ и $*\theta$:

$$*X = *f(*Y, *\theta) = \{f(y, \varphi) | y, \varphi \in *Y \times *\theta\}. \quad (2.33)$$

Для монотонных функций f можно предложить следующее упрощение, сводящееся к поиску точной верхней и нижней граней на отрезках из декартова произведения мульти-интервалов входных Н-переменных:

$$*X = *f(*Y, *\theta) = \{[inf_r(f), sup_r(f)]_{r \in *Y \times *\theta}\}. \quad (2.34)$$

Однако для мульти-интервалов можно воспользоваться интервальной арифметикой, сформировав поведение различных функций на интервалах и мульти-интервалах. После получения обновленного значения Н-переменной с помощью функции (2.33), требуется обновить текущее значение этой Н-переменной, для этого используются функции присваивания W (2.31):

$$W(*A, *B) = *A \cap *B. \quad (2.35)$$

Такая функция не будет расширять начальное значение Н-переменной. В случае, когда пересечения нет, результатом становится пустое множество, показывающее, что входные данные противоречат друг другу. Функции проверки корректности C (2.31) выполняют эту роль:

$$C(*A) = (*A \neq \emptyset). \quad (2.36)$$

Помимо этого, как уже упоминалось, задача локализации связана с рядом погрешностей. Например, в измерениях ориентиров (2.28) величина с погрешностью также может быть задана в виде Н-переменной. Предлагается использовать правило «N сигм» для перевода классической среднеквадратичной ошибки в Н-переменную:

$$\sigma a \rightarrow *\Delta a = [-N\sigma a, N\sigma a]. \quad (2.37)$$

Введя основные принципы формирования Н-модели для задачи локализации по ориентирам, требуется описать функции интерпретации, формируемые ограничениями R (2.31). Данные ограничения задаются из описания задачи и различные подходы их формирования приведены в следующих подразделах. Далее для краткости метод локализации на основе Н-моделей будет именоваться Н-локализацией.

2.2.4. Ограничения, задаваемые одним ориентиром

После введения N -переменных и их стартовых значений, требуется ввести ограничения R (2.31), которые должны быть представимы функционально для подстановки в (2.33). Соответственно, базовым элементом, дающим информацию в исследуемой постановке, является ориентир с заданным положением (2.27) и его измерениями (2.28) (Рис. 2.11).

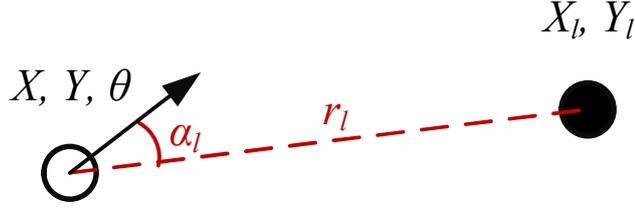


Рис. 2.11. Связь положения ориентира (X_l, Y_l), измерений ориентира (r_l, α_l) и положения робота (X, Y, θ)

Функционально зависимость, изображенную на Рис. 2.11, можно описать следующим образом:

$$(Y_l - Y)^2 + (X_l - X)^2 = (r_l)^2, \quad (2.38)$$

$$\theta = \arctan\left(\frac{Y_l - Y}{X_l - X}\right) - \alpha_l. \quad (2.39)$$

Однако, если измерения (2.28) определяются с погрешностями (2.37), тогда искомые переменные X, Y и θ заменяются на N -переменные и добавляются дополнительные N -переменные, характеризующие погрешности:

$$(Y_l - {}^*Y)^2 + (X_l - {}^*X)^2 = (r_l + {}^*\Delta r_l)^2, \quad (2.40)$$

$${}^*\theta = \arctan\left(\frac{Y_l - {}^*Y}{X_l - {}^*X}\right) - (\alpha_l + {}^*\Delta\alpha_l). \quad (2.41)$$

Далее из уравнений (2.40) и (2.41) требуется выразить искомые N -переменные ${}^*X, {}^*Y$:

$${}^*X = \pm\sqrt{(r_l + {}^*\Delta r_l)^2 - ({}^*Y - Y_l)^2} + X_l, \quad (2.42)$$

$${}^*Y = \pm\sqrt{(r_l + {}^*\Delta r_l)^2 - ({}^*X - X_l)^2} + Y_l, \quad (2.43)$$

$${}^*X = \frac{({}^*Y - Y_l)}{\tan({}^*\theta + {}^*\Delta\alpha_l + \alpha_l)} + X_l, \quad (2.44)$$

$${}^*Y = ({}^*X - X_l) \tan({}^*\theta + \alpha_l + {}^*\Delta\alpha_l) + Y_l. \quad (2.45)$$

Н-переменная θ уже выражена в (2.41). Таким образом, функциями интерпретации будут выступать (2.41)-(2.45). После этого вычислительная Н-модель принимает вид, изображенный на Рис. 2.12.

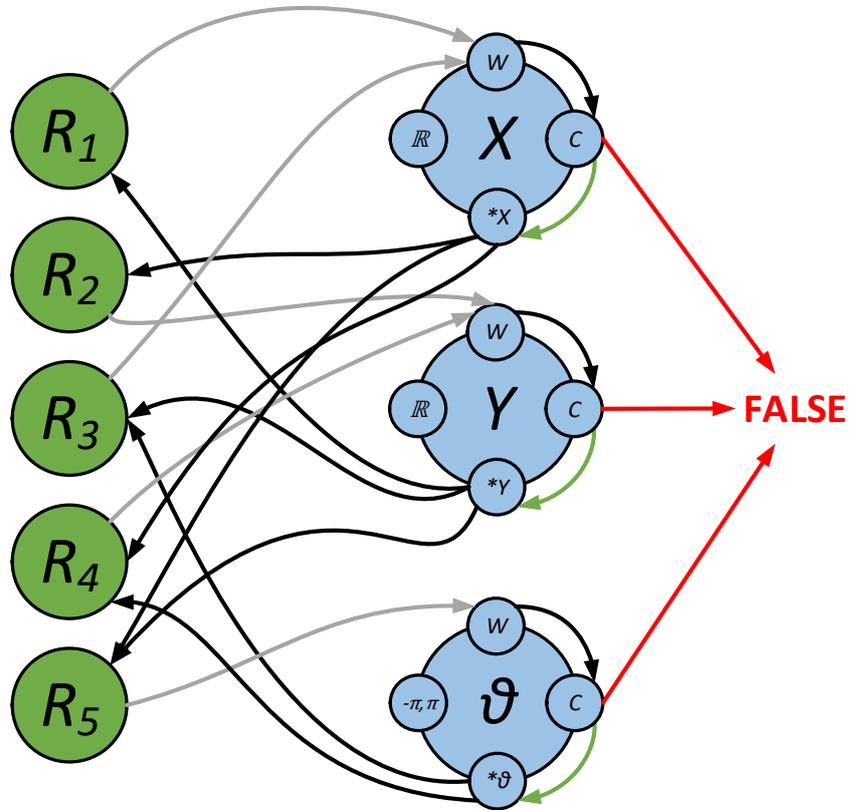


Рис. 2.12. Схематичное изображение Н-модели в виде графа для локализации по одному ориентиру. Функции интерпретации слева соответствуют: R_1 – (2.42), R_2 – (2.43), R_3 – (2.44), R_4 – (2.45), R_5 – (2.41)

На Рис. 2.12 изображена Н-модель для одного ориентиру, но локализация по одному ориентиру обычно даёт слишком большую область неопределённости. Требуется вводить либо дополнительные измерения, например, ориентацию в пространстве наблюдаемого ориентиру, или увеличивать их количество. При увеличении числа ориентиру левая часть на Рис. 2.12 кратно увеличивается.

В случае, когда доступны данные об ориентации распознаваемых объектов γ_l , и она также задаётся на карте θ_l , тогда (2.27) и (2.28) принимают вид:

$$L = \{l = \langle x_l, y_l, \theta_l \rangle\}, \quad (2.46)$$

$$R = \{\langle r_l, a_l, \gamma_l \rangle\}. \quad (2.47)$$

Связь между нововведёнными величинами изображена на Рис. 2.13.

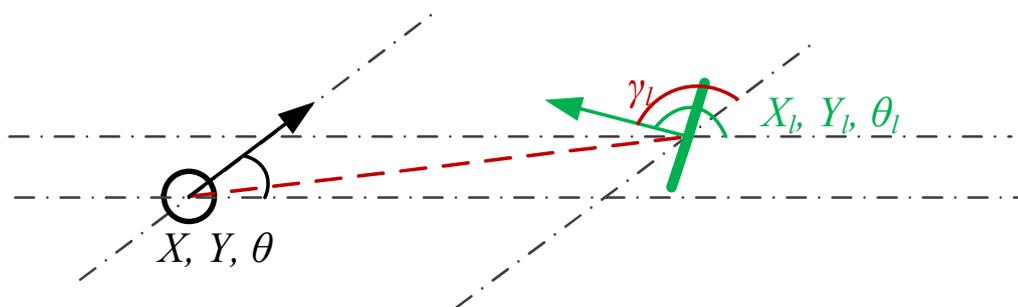


Рис. 2.13. Связь данных об ориентации с положением робота

Из связи на Рис. 2.13 получается следующая функция интерпретации:

$${}^*\theta = \theta_l - (\gamma_l + {}^*\Delta\gamma_l). \quad (2.48)$$

Получившаяся функция также может быть добавлена к вычислительной модели на Рис. 2.12.

Перечисленные функции интерпретации из ограничений R (2.31), (2.38), (2.39) позволят выполнить локализацию методом Н-моделей по визуальным ориентирам, до которых можно замерить расстояние и угол и (опционально) их ориентацию. Это требуется для определения положения робота, который выполнил до этого задачу распознавания сцен (раздел 2.1) и получил набор гипотез f (2.4) о сопоставлении наблюдаемых объектов с объектами на карте.

2.2.5. Глобальная Н-локализация

Задача глобальной локализации по данным от системы распознавания сцен, которая выдает набор гипотез, заключается в отсеивании ложных гипотез и определении ограничений в виде мульти-интервалов (2.32) для положительных гипотез. Ложные гипотезы, которые содержат невыполнимые ограничения по расстояниям и углам, автоматически проверяются на непротиворечивость в ходе вычисления Н-модели. Однако, помимо указанных ограничений, задаваемых ориентиром (Рис. 2.12), существуют также другие особенности. Например, ограничение на угол обзора сенсоров, а также связанные с симметрией наблюдаемой сцены. Для таких особенностей можно ввести дополнительные функции проверки корректности, которые будут применены к результату декартова произведения мульти-интервалов (2.32), получившихся в ходе процедуры вычисления Н-модели (**sdm_process**, Табл. 2.9):

$${}^*a \times {}^*b = \{ \langle [a_{low}^1, a_{high}^1], [b_{low}^1, b_{high}^1] \rangle, \dots, \langle [a_{low}^N, a_{high}^N], [b_{low}^M, b_{high}^M] \rangle \}. \quad (2.49)$$

При этом, в отличие от (2.36), дополнительные функции проверки корректности должны быть от нескольких Н-переменных.

Для учёта горизонтального угла обзора камеры α_h , рассчитанный из (2.41) угол на ориентир α_l должен лежать в пределах $[-\alpha_h, \alpha_h]$. На основе этого можно получить следующую дополнительную функцию проверки корректности:

$$C'_{hfov}(*X, *Y, *\theta) = \begin{cases} True, & \left(\arctan\left(\frac{Y_l - *Y}{X_l - *X}\right) - *\theta \right) \cap [-a_h, a_h] \neq \emptyset, \\ False, & otherwise. \end{cases} \quad (2.50)$$

Упомянутая симметрия наблюдаемой сцены трактуется таким образом, что для вышеуказанных методов распознавания сцен невозможно определить, с какой из противоположных сторон наблюдается сцена (Рис. 2.14).



Рис. 2.14. Пример симметричной сцены для методов распознавания сцен, которым соответствует один и тот же граф

Эта ситуация возможна потому, что методы распознавания сцен рассматривают отношения между объектами исключительно попарно, и учитывают только расстояния. Однако она может быть разрешена на этапе применения дополнительных функций проверки корректности. Рассмотрим пример с двумя ориентирами, которые задают две симметричные области (Рис. 2.15).

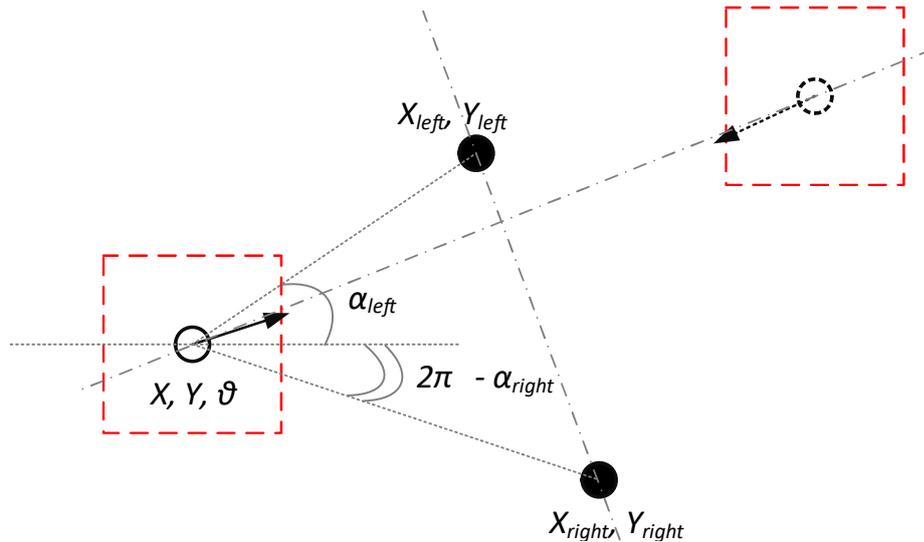


Рис. 2.15. Пример симметричной сцены, состоящей из двух объектов

На Рис. 2.15 две области (отмеченные красным пунктиром) формируются в ходе вычисления Н-модели по двум указанным ориентирам. Однако определить, какая из них верная, можно, сравнив значения углов на ориентир α_{left} и α_{right} в глобальных координатах; если верное положение на рисунке слева, то их разница должна быть больше нуля. Выразив эти углы через Н-переменные положения робота и положения ориентиров, получим:

$$\neg \left(\arctan \left(\frac{*Y - Y_{left}}{*X - X_{left}} \right) - \arctan \left(\frac{*Y - Y_{right}}{*X - X_{right}} \right) < 0 \right) \quad (2.51)$$

Требование положительности в (2.51) было заменено на неотрицательность, это связано с особенностью интервальной арифметики, т.к. для мульти-интервала (2.32) быть больше нуля означает, что все его значения должны быть более нуля, в то время как мульти-интервал может содержать и положительные, и отрицательные значения.

Алгоритм глобальной Н-локализации представлен в виде псевдокода (Табл. 2.10) и схемы.

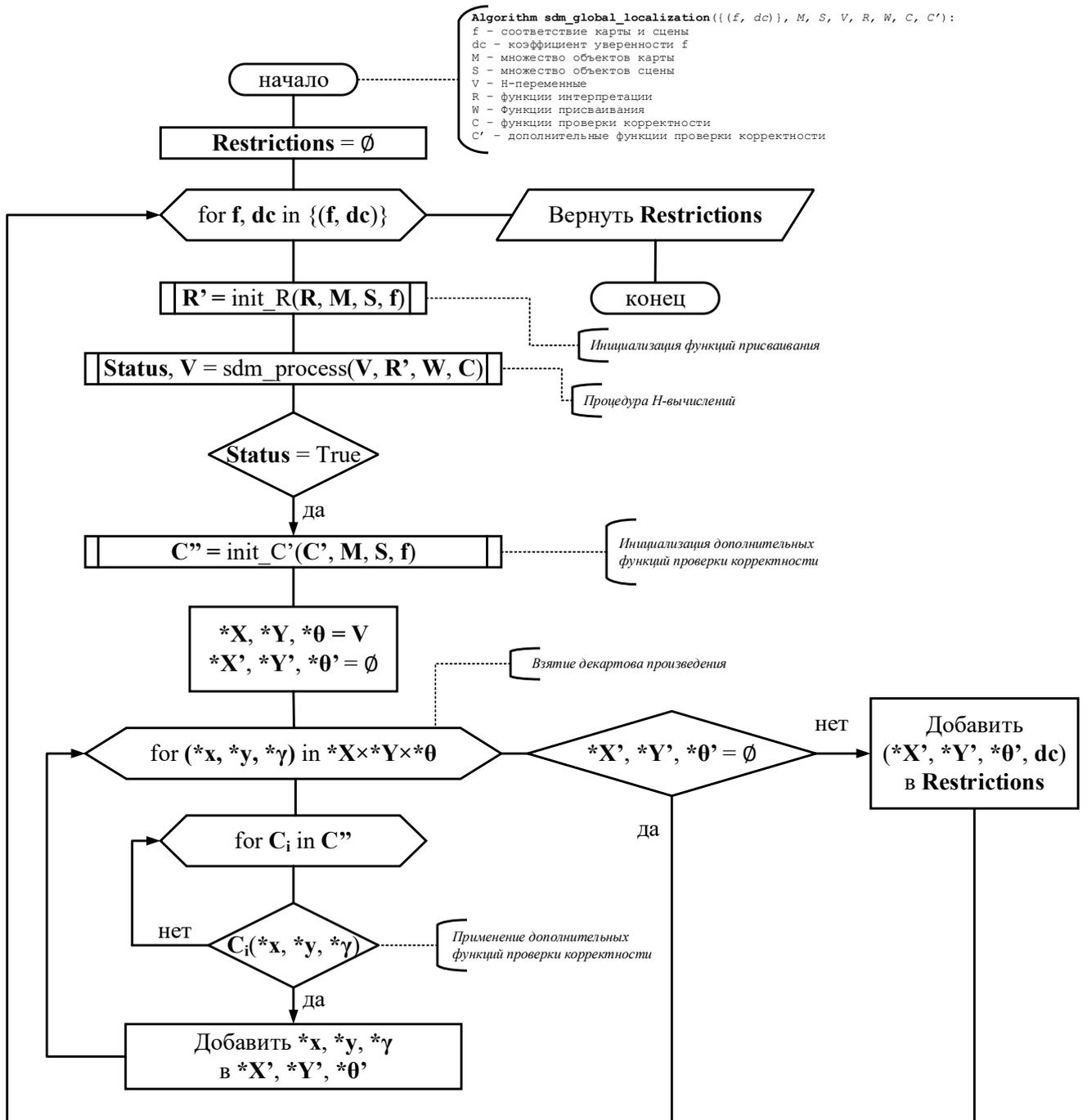
Табл. 2.10. Псевдокод алгоритма глобальной Н-локализации

1.	Algorithm sdm_global_localization ({(f, dc)}, M, S, V, R, W, C,
2.	C') :
3.	-- f - соответствие карты и сцены
4.	-- dc - коэффициент уверенности f
5.	-- M - множество объектов карты
6.	-- S - множество объектов сцены
7.	-- V - Н-переменные
8.	-- R - функции интерпретации
9.	-- W - функции присваивания
10.	-- C - функции проверки корректности

```

11.  -- C' - дополнительные функции проверки корректности
12.  init Restrictions
13.  for f, dc in {(f, dc)}
14.      R' = init_R_params(R, M, S, f)
15.      Status, V = sdm_process(V, R', W, C)
16.      init *X', *Y', *θ'
17.      if Status
18.          C'' = init_C'_params(C', M, S, f)
19.          *X, *Y, *θ = V
20.          for (*x, *y, *γ) in *X*Y*θ
21.              for Ci in C''
22.                  if Ci>(*x, *y, *γ)
23.                      add *x to *X'
24.                      add *y to *Y'
25.                      add *γ to *θ'
26.          add ((*X', *Y', *θ'), dc) to Restrictions
27.  return Restrictions

```

Рис. 2.16. Схема алгоритма глобальной N -локализации

Алгоритм `sdm_global_localization` (Табл. 2.10, Рис. 2.16) принимает на вход результат распознавания сцен, описание карты M и сцены S , элементы N -модели (2.31) и дополнительные функции проверки корректности C' . Функция `init_R_params` инициализирует параметры функций интерпретации, например (2.41)-(2.44), в зависимости от текущего соответствия f , то же самое делает функция `init_C'_params` для дополнительных функций проверки корректности (2.50) и (2.51), функция `sdm_process` реализует алгоритм вычисления N -модели

(Табл. 2.9). Алгоритм возвращает новый набор N -переменных, снабженный соответствующим коэффициентом уверенности (2.26), (3.3) от алгоритмов распознавания сцен (Табл. 2.8, Табл. 3.1). Несмотря на то, что используемый алгоритм **sdm_process** имеет экспоненциальную сложность относительно числа переменных, в данном случае оно ограничено тремя (X, Y, θ). Также, как упоминалось, на практике, обычно ограничивается число итераций в алгоритме **sdm_process**, что в итоге приводит к зависимости времени вычисления от числа функций интерпретации, которое линейно зависит от числа наблюдаемых ориентиров. Таким образом вычислительную сложность алгоритма **sdm_global_localization** можно оценить $O(N)$, где N – число наблюдаемых ориентиров.

Полученные в результате алгоритма **sdm_global_localization** (Табл. 2.10) ограничения могут в дальнейшем использоваться для определения положения робота. В зависимости от числа ориентиров и величины погрешностей размер пространства, описанный данными ограничениями, может варьироваться.

2.2.6. Непрерывная N -локализация

Несмотря на то, что основной целью работы является задача глобальной локализации роботов, возможность разработанного метода проводить непрерывную локализацию также представляет интерес и будет рассмотрена.

Непрерывная локализация отличается от глобальной в первую очередь тем, что локализация происходит в ограниченной области, вокруг положения, рассчитанного на предыдущем шаге. В N -локализации можно оперировать подобным образом N -переменными, описывающими положение робота. Между шагами, на которых выполняется непрерывная локализация, обычно происходит смещение робота. Таким образом, требуется обновить значения N -переменных на основе этого смещения. В двумерном случае при достаточной дискретизации по времени dt смещение робота с дифференциальным приводом может задаваться двумя параметрами – линейной скоростью, смещением v и угловой скоростью w , которые, как и другие измеряемые величины, задаются с погрешностями, выраженными N -переменными (2.37) - ${}^* \Delta v, {}^* \Delta w$. Тогда обновлённое значение N -переменных будет рассчитываться как:

$$\begin{aligned} {}^* \theta_{t+dt} &= {}^* \theta_t + (w + {}^* \Delta w) dt, \\ {}^* X_{t+dt} &= {}^* X_t + (v + {}^* \Delta v) \cos({}^* \theta_{t+dt}) dt, \\ {}^* Y_{t+dt} &= {}^* Y_t + (v + {}^* \Delta v) \sin({}^* \theta_{t+dt}) dt. \end{aligned} \quad (2.52)$$

После того, как расширение Н-переменных получено, можно воспользоваться алгоритмом глобальной локализации (**sdm_global_localization**, Табл. 2.10), подав ему на вход полученные ограничения как стартовые значения. Подача стартовых значений позволит сократить время вычислений, а также исключить возможности попадания в удаленные области карты. В случае, если полученные значения не удовлетворяют новым ограничениям (произошла ошибка или робот был перемещён независимым от него способом), то они сбрасываются до стартовых и заново происходит процедура глобальной локализации.

Табл. 2.11. Псевдокод алгоритма непрерывной Н-локализации

1.	Algorithm sdm_continuous_localization ($\{(f, dc)\}, M, S, V, R, W, C, C', Odom$):
2.	-- f - соответствие карты и сцены
3.	-- dc - коэффициент уверенности f
4.	-- M - множество объектов карты
5.	-- S - множество объектов сцены
6.	-- V - Н-переменные
7.	-- R - функции интерпретации
8.	-- W - функции присваивания
9.	-- C - функции проверки корректности
10.	-- C' - дополнительные функции проверки корректности
11.	-- Odom - смещение робота
12.	$V_{exp} = \text{sdm_odom}(V, Odom)$
13.	$Restrictions = \text{sdm_global_localization}(\{(f, dc)\}, M, S,$
14.	$V_{exp}, R, W, C, C')$
15.	if $Restrictions = \emptyset$
16.	reinit V
17.	$Restrictions = \text{sdm_global_localization}(\{(f, dc)\}, M,$
18.	$S, V, R, W, C, C')$
19.	return $Restrictions$

Алгоритм **sdm_continuous_localization** (Табл. 2.11) принимает на вход результат методов распознавания сцен (Табл. 2.8, Табл. 3.1), карту M и наблюдаемую сцену S , параметры Н-модели, где V это набор Н-переменных с предыдущего шага (или стартовых), дополнительные функции проверки корректности C' и смещение $Odom$. Функция **sdm_odom** расширяет значения Н-переменных на основе данных о смещении робота согласно (2.52) или аналогичным способом. Алгоритм возвращает новые ограничения на положение робота. Вычислительная сложность алгоритма **sdm_continuous_localization** оценивается как линейная от числа наблюдаемых ориентиров аналогично алгоритму **sdm_global_localization** (Табл. 2.10).

2.2.7. Проблема исчерпывающего оценивания

Во время применения механизма N -вычислений может возникнуть ситуация, когда для некоторых входных данных обновление N -переменных заканчивается на ранней стадии и не может быть улучшено (Рис. 2.17).

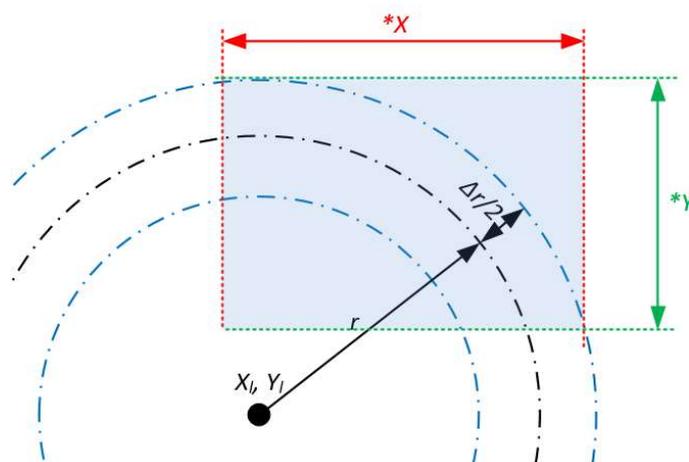


Рис. 2.17. Пример ситуации исчерпывающего оценивания: по значению $*X$ можно получить с помощью (2.43) $*Y$, однако дальнейшее применение функций интерпретации не улучшит ситуацию. В данном случае в область неопределённости $*X \times *Y$ попадает значительное число лишних значений

Такая ситуация в интервальном анализе именуется исчерпывающим оцениванием [Баженов и др., 2024]. Способ её устранения заключается в применении алгоритмов, последовательно разбивающих набор ограничений и применяющих к получившимся разбиениям вычислительную процедуру. Основная идея данного семейства алгоритмов, носящих название ESIS (Exhaustion of Solutions to Interval System), SIVIA (Set Inversion via Interval Analysis), состоит в итеративном разбиении интервала, превосходящего порог. В случае N -переменных, такая переменная, выраженная интервалом, делится на две независимые N -переменные, а дальше к ней применяется процедура вычисления (**sdm_process**, Табл. 2.9) за тем исключением, что ей передаётся не весь набор функций интерпретации, а только те, для которых разбитая N -переменная является входом. На Рис. 2.18 указан результат уточнения после первого шага.

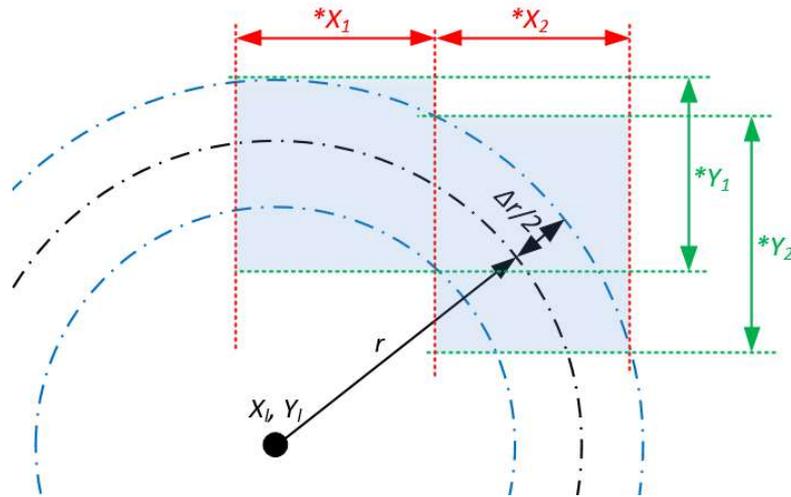


Рис. 2.18. Разрешение ситуации исчерпывающего оценивания N -переменных путем разбиения N -переменных

Продолжая данную операцию, можно снизить количество лишних значений в результирующих интервалах. Если после разбиения и применения функций интерпретации результирующие значения по-прежнему превышают порог, то процедура повторяется. Для данной процедуры требуется, чтобы входными значениями были именно интервалы, а не мульти-интервалы (2.32), поэтому сначала происходит взятие декартова произведения (2.49), и к нему уже применяется описанная процедура (Табл. 2.12).

Табл. 2.12. Псевдокод алгоритма исчерпывающего оценивания N -переменных

1.	Algorithm <code>sdm_esis(V, R, W, C, xy_lim):</code>
2.	-- V - N -переменные
3.	-- R - функции интерпретации
4.	-- W - функции присваивания
5.	-- C - функции проверки корректности
6.	-- xy_lim - порог разбиения
7.	init <code>rectified</code>
8.	function <code>process_leaf(*x, *y, *γ)</code>
9.	>Status, V' = <code>sdm_process(*x, *y, *γ, R, W, C)</code>
10.	if <code>Status</code>
11.	(*x, *y, *γ) = V'
12.	if <code> *x > xy_lim:</code>
13.	*x1, *x2 = <code>divide(*x)</code>
14.	process_leaf (*x1, *y, *γ)
15.	process_leaf (*x2, *y, *γ)
16.	elif <code> *y > xy_lim:</code>
17.	*y1, *y2 = <code>divide(*y)</code>
18.	process_leaf (*x, *y1, *γ)
19.	process_leaf (*x, *y2, *γ)
	else

```

20.         add (*x, *y, *γ) to rectified
21.     *X, *Y, *θ = V
22.     for (*x, *y, *γ) in *X*Y*θ
23.         process_leaf(*x, *y, *γ)
24.     return rectified
25.

```

Алгоритм **sdm_esis** (Табл. 2.12) принимает на вход N-модель (2.31), порог разбиения xu_lim , итеративно разбивает N-переменные на две равные части функцией **divide** и применяет к ним процедуру вычисления N-модели **sdm_process** (Табл. 2.9). Вычислительная сложность данного алгоритма вырастает по-прежнему линейна от числа ориентиров, однако в худшем случае квадратична от числа разбиений интервалов, которое обратно пропорционально параметру xu_lim , т.е. $O(N^2)$, где $N = \max(|*X|, |*Y|)/xu_lim$.

Безусловно, такая процедура значительно увеличивает время вычисления, однако поднимает точность определения финальной зоны (Рис. 2.19).

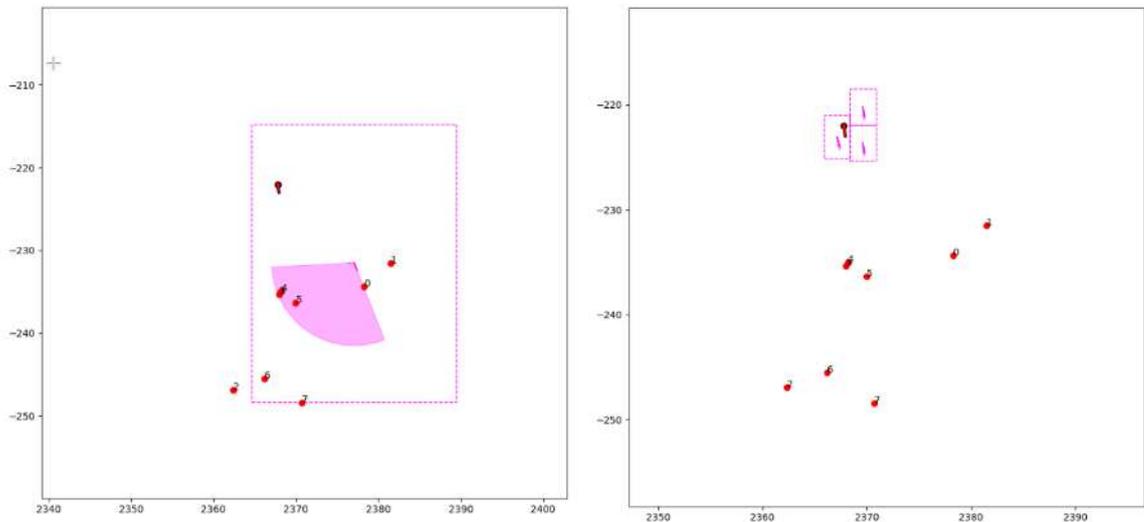


Рис. 2.19. Пример результата алгоритма уточнения N-переменных. Красная точка со стрелкой с черной окантовкой – положение робота, пронумерованные красные точки – ориентиры, лиловая область – значения N-переменных до алгоритма уточнения (слева) и после (справа)

Эксперименты по сравнению точности и времени расчета значений с использованием приведенного алгоритма и без него приведены в подразделе 4.3.1.

2.2.8. Использование дополнительных сенсоров в N-локализации

В практической робототехнике могут быть дополнительные источники данных о положении робота, полученные, например, от средств глобальной

навигации или иных сенсоров. Механизм Н-вычислений позволяет также учесть разнообразные данные путем формирования ограничений на их основе. Таким образом, если есть положение глобальной навигации, выраженное в виде точки в пространстве (X_G, Y_G, ϑ_G) , и значений погрешности, например, заданной ковариационной матрицей P_G , то по ним можно сформировать следующий набор ограничений:

$${}^*X_G = X_G + {}^*\Delta(P_G[0,0] \cos(\theta_{P_G})), \quad (2.53)$$

$${}^*Y_G = Y_G + {}^*\Delta(P_G[1,1] \sin(\theta_{P_G})), \quad (2.54)$$

$${}^*\theta_G = \theta_G + {}^*\Delta(P_G[2,2]), \quad (2.55)$$

где θ_{P_G} – угол поворота эллипса, построенного по матрице ковариации P_G . Подобным образом возможно использование любых данных, выдающих положение в схожих нотациях. Для обновления значений Н-переменных достаточно применить функцию присваивания W (2.35):

$${}^*X_{new} = W({}^*X_{old}, {}^*X_G). \quad (2.56)$$

2.2.9. Функции интерпретации для задачи локализации по пеленгу

Некоторые из сформулированных функций присваивания (2.41)-(2.45) основаны на знании расстояния до объектов. Однако существует ряд случаев, когда измерение расстояния до объекта невозможно, или погрешности его определения настолько велики, что делают дальнейшие расчеты мало полезными. Для таких случаев предлагается ввести дополнительный набор функций интерпретации, основанный только на знании угла на объект. Данные функции интерпретации построены на геометрическом соотношении между двумя ориентирами и положением робота (Рис. 2.20).

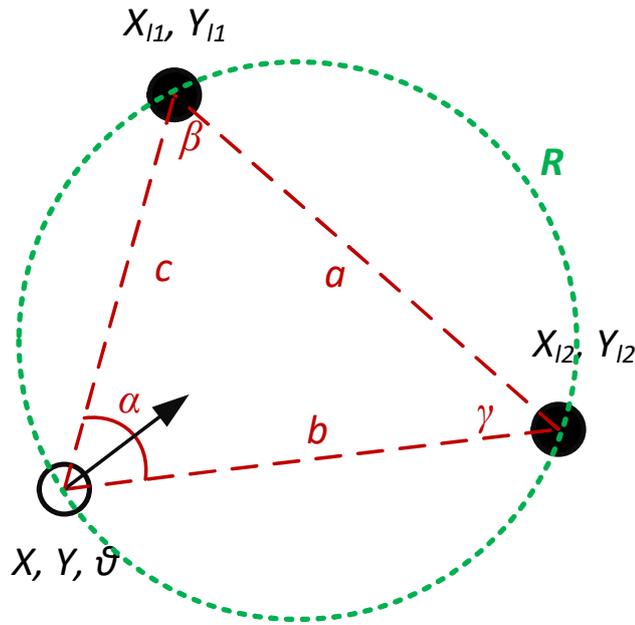


Рис. 2.20. Взаимоположение двух ориентиров и положения робота

На Рис. 2.20 показано, что положение робота и двух ориентиров могут быть помещены на окружность радиуса R (кроме случаев, когда все элементы лежат на одной прямой), тогда для треугольника, вписанного в эту окружность, можно записать теорему синусов:

$$\frac{a}{\sin \alpha} = \frac{b}{\sin \beta} = \frac{c}{\sin \gamma} = 2R. \quad (2.57)$$

В соотношении (2.57) точно известны значения a и α (также изображенные на Рис. 2.20):

$$a = \sqrt{(X_{l1} - X_{l2})^2 + (Y_{l1} - Y_{l2})^2}, \quad (2.58)$$

$$\alpha = |\alpha_{l1} - \alpha_{l2}|. \quad (2.59)$$

Из них можно выразить радиус описанной окружности:

$${}^*R = \frac{\alpha + {}^*\Delta\alpha_{l1} + {}^*\Delta\alpha_{l2}}{2a}, \quad (2.60)$$

т.к. углы определяются с погрешностями, то и значение радиуса, заданное N -переменной, зависит от этих погрешностей. Стороны b и c треугольника могут быть выражены через положения робота, описанные N -переменными:

$${}^*b = \sqrt{(X_{l2} - {}^*X)^2 + (Y_{l2} - {}^*Y)^2}, \quad (2.61)$$

$${}^*c = \sqrt{(X_{l1} - {}^*X)^2 + (Y_{l1} - {}^*Y)^2}. \quad (2.62)$$

Подставив *b и *c в (2.57), можно получить значения углов β и γ :

$$*\beta = \arcsin\left(\frac{*b}{2 *R}\right), \quad (2.63)$$

$$*\gamma = \arcsin\left(\frac{*c}{2 *R}\right). \quad (2.64)$$

Через эти значения можно выразить искомые положения Н-переменных:

$$*X = \sqrt{(2 *R \sin(*\beta))^2 - (Y_{l1} - *Y)^2} + X_{O1}, \quad (2.65)$$

$$*Y = \sqrt{(2 *R \sin(*\beta))^2 - (X_{l1} - *X)^2} + Y_{O1}, \quad (2.66)$$

$$*X = \sqrt{(2 *R \sin(*\gamma))^2 - (Y_{l2} - *Y)^2} + X_{O2}, \quad (2.67)$$

$$*Y = \sqrt{(2 *R \sin(*\gamma))^2 - (X_{l2} - *X)^2} + Y_{O2}. \quad (2.68)$$

Значения $*\beta$ (2.63) и $*\gamma$ (2.64) можно рассчитывать на каждом шаге вновь в рамках формул (2.65)-(2.68), а можно оформить как самостоятельные Н-переменные, инициированные на промежутке значений $[0, \pi]$, и ввести для них дополнительные функции интерпретации, основанные на теореме о суммах углов треугольника:

$$\alpha + *\Delta\alpha + *\beta + *\gamma = \pi. \quad (2.69)$$

Из выражения (2.69) получатся следующие функции интерпретации:

$$*\beta = \pi - *\gamma - (\alpha + *\Delta\alpha), \quad (2.70)$$

$$*\gamma = \pi - *\beta - (\alpha + *\Delta\alpha). \quad (2.71)$$

Полученные новые функции интерпретации и новые Н-переменные формируются для каждой пары входных ориентиров, что даёт квадратичную зависимость, увеличивая сложность алгоритмов локализации Табл. 2.10, Табл. 2.11) до $O(N^2)$. Однако практика показала, что наиболее ценными являются те ориентиры, которые максимально разнесены друг от друга по углу наблюдения. Данную эвристику можно использовать для сокращения вычислений, выбирая k пар, наиболее разнесённых друг от друга, для включения в модель. Эти функции, а также (2.41), (2.44) и (2.45) позволяют определять положение робота в виде ограничений даже в варианте без знания расстояния до объектов.

2.3. Архитектура глобальной локализации по визуальным ориентирам

Данный раздел описывает архитектуру глобальной локализации по ориентирам, которая включает в себя разобранные ранее методы распознавания

сцен и N-локализацию, а также классические подходы локализации с использованием дальномерных данных (облаков точек).

2.3.1. Общий подход

Как упоминалось в первой главе, в методах локализации с использованием распознавания сцен практически не используются данные с дальномеров, за исключением стадии определения расстояний до семантических объектов. Положение робота определяется сугубо по данным об этих объектах-ориентирах. Однако ввиду ряда ограничений (ошибок распознавания и локализации объектов) процесс определения положения связан с высокими погрешностями, что может приводить к низкой точности получаемых решений. Альтернативный подход с использованием данных в виде облаков точек, хоть и обеспечивает приемлемую точность локализации в локальной области, испытывает сложности с точностью и со временем на картах больших размеров, особенно в случае сред, обладающих симметрией и периодичностью.

Основная идея диссертационной работы заключается в комбинации данных подходов, где алгоритмы распознавания сцен выполняют привязку к объектам карты, N-локализация позволяет ограничить область вокруг этих объектов, в которой может находиться робот с учетом погрешностей, а классические методы локализации применяются к получившимся областям, размер которых значительно меньше полной карты пространства.

Архитектура предлагаемого процесса представлена на Рис. 2.21.

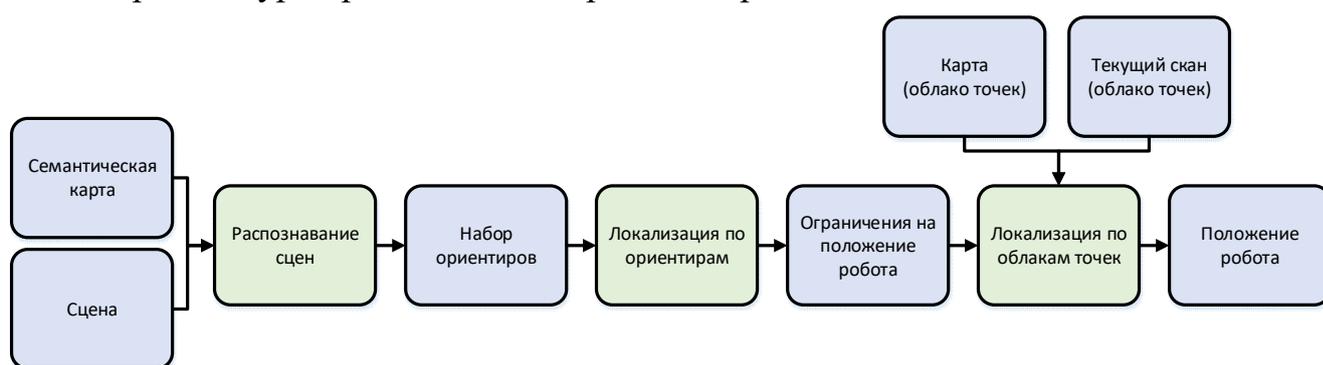


Рис. 2.21. Архитектура глобальной локализации по семантическим данным

Архитектура на Рис. 2.21 отличается от общепринятой архитектуры в исследуемой области (Рис. 1.1) именно блоком локализации по ориентирам, который позволяет получить ограничения на положение робота. Однако, методы

локализации по облакам точек требуют некоторой доработки, чтобы иметь возможность быть запущенными в полученных от Н-локализации ограничениях, выраженными набором мульти-интервалов (2.32) или декартовым их произведением (2.49). Далее в главе будут предложены способы подобного учёта.

2.3.2. Использование полученных ограничений для методов локализации по облакам точек

Методы локализации по облакам точек, например дескрипторам FPFH и связки RANSAC+ICP, работают с двумя облаками точек: картой и окружением робота. Самым простым способом учесть полученные от Н-локализации интервальные ограничения, это модифицировать входное облако точек карты по координатам x и y . Однако, стоит отметить, что полученные ограничения обычно показывают лимит на положение центра робота, поэтому, чтобы не отсечь от карты лишние области, полученные ограничения стоит расширить на рабочую дистанцию используемого сенсора D_{sens} :

$${}^*X^+ = {}^*X + [-D_{sens}, D_{sens}]. \quad (2.72)$$

Проведя аналогичную операцию для Н-переменной *Y можно использовать значения ${}^*X^+$ и ${}^*Y^+$ для обрезки облака точек в данных ограничениях, оставляя только те точки, координаты которых удовлетворяют полученным ограничениям. Т.к. Н-переменные представимы в мульти-интервалах то операцию (2.72) рекомендуется делать для результатов представления в виде декартового произведения (2.49) ${}^*X \times {}^*Y$, где для каждого элемента результата получается своё облако, для которого можно выполнить локализацию. Ограничения на угол положения робота ${}^*\theta$ не могут быть напрямую использованы при обрезке облаков, за исключением ситуаций, когда угол обзора сенсора в горизонтальной плоскости α_h сильно ограничен. В таком случае формулы расширения области принимают вид:

$$\begin{aligned} {}^*X^+ &= {}^*X + D_{sens} \cos({}^*\theta + [-\alpha_h, \alpha_h]), \\ {}^*Y^+ &= {}^*Y + D_{sens} \sin({}^*\theta + [-\alpha_h, \alpha_h]). \end{aligned} \quad (2.73)$$

Общий алгоритм использования ограничений для методов на основе сравнения двух облаков точек представлен в Табл. 2.13:

Табл. 2.13. Псевдокод алгоритма использования ограничений для локализации по облакам точек

1.	Algorithm sdm_pc_match ($V, D_{sens}, PCD_{map}, PCD_{robot}, matcher$):
2.	-- V - декартово произведение Н-перменных положения робота

3.	-- Dsens – рабочая дистанция дальномерного сенсора
4.	-- PCD_{map} – облако точек карты
5.	-- PCD_{robot} – облако точек текущего наблюдения
6.	-- matcher – функция локализации по облакам точек, возвращающая положение и численную оценку
7.	$A = \emptyset$
8.	for $*X, *Y, *\theta$ in V
9.	$+X*, +Y*$ = expand ($*X, *Y, *\theta$)
10.	PCD_{submap} = cut_pc ($PCD_{map}, +X*, +Y*$)
11.	$pose, score$ = matcher ($PCD_{submap}, PCD_{robot}$)
12.	add $pose, score$ to A
13.	return A

Алгоритм **sdm_pc_match** (Табл. 2.13) принимает на вход ограничения в виде декартова произведения, рабочую дистанцию сенсора в горизонтальной плоскости, облака точек карты и окружения и функцию поиска по двум облакам точек. Функция **expand** реализует (2.73), функция **cut_pc** реализует образку облака по полученным координатам. Алгоритм возвращает набор включающий положение робота и численную оценку. Вычислительная сложность данного алгоритма $O(N)$, где N – число элементов декартова произведения.

Также полученные ограничения, в том числе и на угол, можно использовать в алгоритмах локализации по облакам точек, которые генерируют гипотезы (например алгоритмы семейства RANSAC), добавив стадию проверки этих гипотез полученным ограничениям.

2.3.3. Использование фильтра частиц для уточнения положения в ограничениях

Другим распространенным подходом решения задачи локализации в робототехнике является применение ФЧ (методы MCL (Monte-Carlo Localization) и AMCL (Augmented\Adaptive MCL)). В основе метода лежит понятие частицы – гипотезы о положении робота. Суть алгоритма заключается в итеративном расчете весов множества таких гипотез, их отсеивании и клонировании. Эти методы привлекательны тем, что не имеют дискретного шага (например, как ГФ), и им не требуется совершать полный перебор всего пространства поиска. Вместо этого происходит случайный поиск с итеративным уточнением в наиболее перспективных областях, однако это же приводит к проблемам, связанным с попаданием в локальные минимумы, которые могут быть ввиду повторяемости и симметричности пространства. Входными данными для алгоритмов локализации на основе ФЧ могут быть как лидарные данные, так и данные о положении

ориентиров. Однако очень важным аспектом для ФЧ является начальное распределение, которое влияет как на точность так и на скорость схождения алгоритма. Ограничения, полученные алгоритмом **sdm_global_localization** (Табл. 2.10), могут быть использованы для инициализации стартового распределения частиц, тем самым ограничивая область поиска. Это обеспечивает в более быстрое схождение алгоритма, а также исключает возможность попадания в локальные минимумы, находящиеся за пределами этих ограничений. Алгоритм **restrictions_to_particles** позволяет получить такое распределение частиц:

Табл. 2.14. Псевдокод алгоритма инициализации распределения частиц по ограничениям, заданным в виде мульти-интервалов

1.	Algorithm restrictions_to_particles (<i>Restrictions</i> , <i>N</i>):
2.	-- Restrictions - набор мультиинтервалов, снабженных
	коэффициентами уверенности
3.	-- N - число частиц
4.	init <i>P</i>
5.	for <i>n</i> in 0: <i>N</i> do
6.	(<i>V</i> , <i>dc</i>) = sample_restriction (<i>Restrictions</i>)
7.	(* <i>X</i> , * <i>Y</i> , * <i>θ</i>) = <i>V</i>
8.	<i>x</i> = sample_value (* <i>X</i>)
9.	<i>y</i> = sample_value (* <i>Y</i>)
10.	<i>θ</i> = sample_value (* <i>θ</i>)
11.	add (<i>x</i> , <i>y</i> , <i>θ</i>) to <i>P</i>
12.	return <i>P</i>

Алгоритм **restrictions_to_particles** (Табл. 2.14) принимает на вход набор ограничений в виде мульти-интервалов (2.32) и число желаемых частиц *N*. Функция **sample_restriction** выбирает случайным образом один набор ограничений с учетом их коэффициентов уверенности, полученных от методов распознавания сцен: чем выше коэффициент, тем с большей вероятностью будет выбрано ограничение, ему соответствующее. Функция **sample_value** равновероятно выбирает одно случайное значение, принадлежащее мульти-интервалу. Алгоритм возвращает набор частиц, которые дальше могут быть взвешены согласно методам, основанным на ФЧ. Вычислительная сложность данного алгоритма $O(N)$. Этот алгоритм учитывает как размеры областей, так и их пересечения (Рис. 2.22).

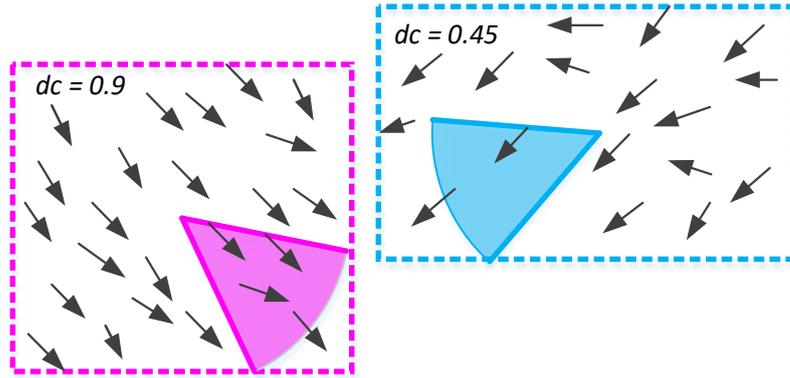


Рис. 2.22. Схематичная иллюстрация результата работы алгоритма **restrictions_to_particles** (Табл. 2.14)

Некоторые алгоритмы, например, **sdm_isis** (Табл. 2.12), возвращают не набор мульти-интервалов, а набор кортежей интервалов, получаемый в результате декартова произведения (2.49) N -переменных $*X$, $*Y$, $*\theta$. Для такого варианта ответа также требуется способ инициализации распределения частиц:

Табл. 2.15. Псевдокод алгоритма инициализации распределения частиц по декартову произведению N -переменных

1.	Algorithm cartesian_to_particles (V , N):
2.	-- V - декартово произведение ($*X*Y*\theta$)
3.	-- N - число частиц
4.	init P
5.	init W
6.	for ($*X$, $*Y$, $*\theta$) in V do
7.	add $ *X * *Y * *\theta $ to W
8.	$W = W $
9.	for n in $0:N$ do
10.	($*X$, $*Y$, $*\theta$) = sample_weighted (V , W)
11.	x = sample_value ($*X$)
12.	y = sample_value ($*Y$)
13.	θ = sample_value ($*\theta$)
14.	add (x , y , θ) to P
15.	return P

Алгоритм **cartesian_to_particles** (Табл. 2.15) принимает на вход результат декартова произведения и желаемое число частиц N . Функция **sample_weighted** выбирает из множества элемент согласно указанной вероятности, функция **sample_value** равновероятностно выбирает одно случайное значение, принадлежащее интервалу. Вычислительная сложность – $O(N)$.

Используя полученное начальное распределение частиц, можно уточнить положение робота по любым доступным данным, позволяющих оценивать

совпадения в точке пространства, например, по данным сканирующих дальномеров, алгоритм для которых представлен в виде псевдокода (Табл. 2.16).

Табл. 2.16. Псевдокод алгоритма глобальной локализации по облакам точек с использованием ФЧ

1.	Algorithm pf_global_localization ($V, \sigma, N, iter, \Delta, PCD_{map},$
2.	PCD_{robot}):
3.	-- V – ограничения на положение робота, в виде декартова произведения интервалов
4.	-- σ – чувствительность
5.	-- N – число частиц
6.	-- $iter$ – число итераций
7.	-- Δ – параметры шума
8.	-- PCD_{map} – облако точек карты
9.	-- PCD_{robot} – облако точек с робота
10.	$P = \text{cartesian_to_particles}(V, N)$
11.	for i in $iter$ do
12.	$W = \emptyset$
13.	$P' = \emptyset$
14.	for p in P do
15.	$p = \text{random_shift}(p, \Delta)$
16.	add p to P'
17.	$PCD = \text{transform_cloud}(PCD_{robot}, p)$
18.	$d = \text{get_mean_p2p}(PCD, PCD_{map})$
19.	$w = \text{get_w}(d, \sigma)$
20.	add w to W
21.	$P = \text{resampling}(P', W)$
22.	$pose_{final}, W_{final} = \text{arg_max}(P, W)$
23.	return $pose_{final}, W_{final}$

Алгоритм **pf_global_localization** (Табл. 2.16) принимает на вход набор ограничений, полученных от алгоритма **sdm_esis** (Табл. 2.12), ряд параметров, а также облака точек с робота и карты. Используется алгоритм **cartesian_to_particles** (Табл. 2.15) для формирования множества частиц. Функция **random_shift** добавляет к частицам случайный белый шум с указанными параметрами. Функция **transform_cloud** производит геометрический перенос облака точек в координаты частицы. Функция **get_mean_p2p** производит расчет кратчайших расстояний между точками первого облака до точек второго облака. Функция **get_w** рассчитывает вес частицы согласно (2.9). Функция **resampling** производит переВыборку (resampling) частиц, функция **arg_max** выбирает частицу с максимальным весом, которая выступает в роли ответа. Вычислительная сложность данного алгоритма – $O(NM)$, где N – число частиц, а M – число итераций.

2.3.4. Технология глобальной локализации по визуальным ориентирам

Предложенная архитектура (Рис. 2.21) реализуется комплексной технологией² локализации по видимым ориентирам. Некоторые части технологии допускают варианты предложенных алгоритмов, однако в целом процесс можно описать общим алгоритмом (Табл. 2.17), иллюстрирующим последовательность и передачу данных между отдельными элементами.

Табл. 2.17. Псевдокод алгоритма иллюстрирующего комплексную технологию глобальной локализации по ориентирам

1.	Algorithm complex_global_localization (<i>S</i> , <i>M</i> , <i>PCD_{map}</i> , <i>PCD_{robot}</i> , <i>M</i> , <i>params</i>):
2.	-- <i>S</i> - описание наблюдаемой сцены
3.	-- <i>M</i> - семантическая карта
4.	-- <i>PCD_{map}</i> - облако точек карты
5.	-- <i>PCD_{robot}</i> - облако точек с робота
6.	-- <i>M</i> - Н-модель (<i>V</i> , <i>C</i> , <i>C'</i> , <i>R</i> , <i>W</i>)
7.	-- <i>params</i> - параметры алгоритмов
8.	{(<i>f</i> , <i>dc</i>)} = sig_soft (<i>S</i> , <i>M</i> , <i>params</i>)
9.	<i>Res</i> = sdm_global_localization ({(<i>f</i> , <i>dc</i>)}, <i>M</i> , <i>S</i> , <i>V</i> , <i>R</i> , <i>W</i> , <i>C</i> , <i>C'</i>)
10.	<i>Rect</i> = sdm_isis (<i>Res</i> , <i>V</i> , <i>R</i> , <i>W</i> , <i>C</i> , <i>C'</i> , <i>params</i>)
11.	<i>pose</i> , <i>w</i> = pf_global_localization (<i>Rect</i> , <i>params</i> , <i>PCD_{map}</i> , <i>PCD_{robot}</i>)
12.	return <i>pose</i>

² Под «технологией» в данном контексте понимается «информационная технология» как совокупность методов, устройств и производственных процессов, используемых людьми для сбора, хранения, обработки и распространения информации.

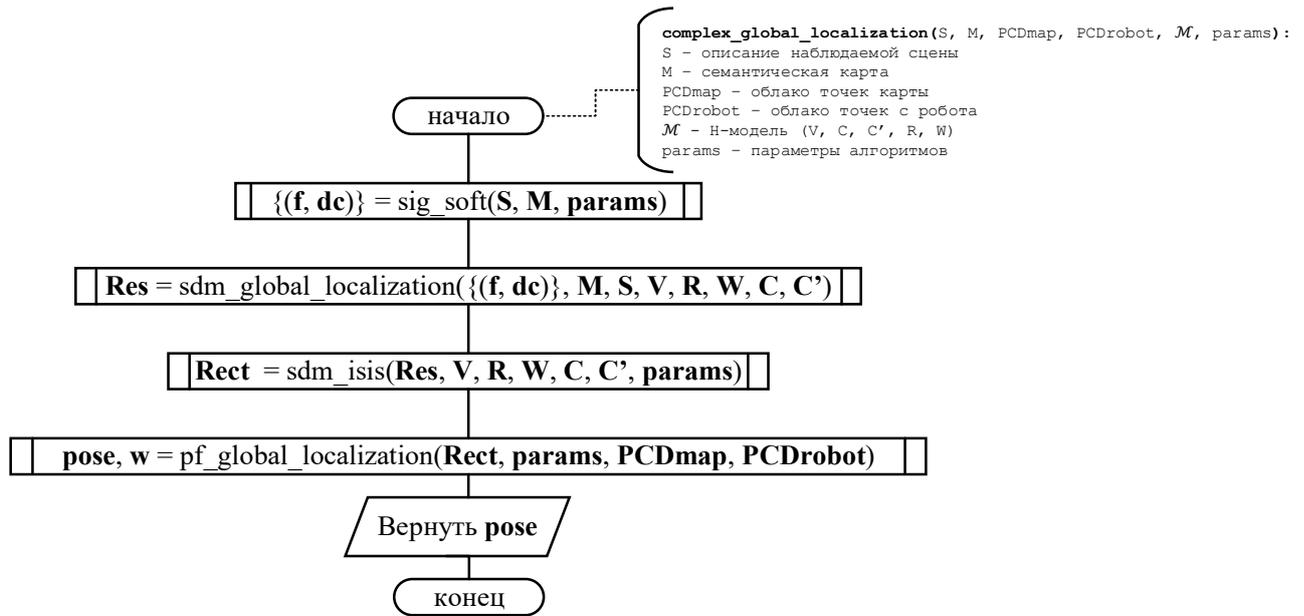


Рис. 2.23. Схема алгоритма иллюстрирующего комплексную технологию глобальной локализации по ориентирам

Алгоритм **complex_global_localization** (Табл. 2.17, Рис. 2.23) принимает на вход семантические описания карты (2.3) и сцены (2.2), соответственные облака точек, N-модель (2.31), а также набор дополнительных параметров для вызываемых алгоритмов **sig_soft** (Табл. 2.8), **sdm_global_localization** (Табл. 2.10), **sdm_isis** (Табл. 2.12) и **pf_global_localization** (Табл. 2.16). Таким образом задача глобальной локализации по визуальным ориентирам и методами уточнения по дальномерным данным решается предложенной последовательностью действий. В следующих главах будет рассмотрена практическая реализация предложенной архитектуры в нескольких вариантах исполнения, учитывающий специфику конкретной задачи и робота.

2.3.5. Требования к робототехническому комплексу

Для использования предлагаемой архитектуры, для задачи глобальной локализации робототехническая система должна обладать следующими возможностями:

1. Наличие сенсорики, способной выделять, классифицировать и локализовать статические объекты в окружении робота. Это может быть как камера, снабженная дальномерным сенсором, например стерео-зрением, так и дальноммеры типа LiDAR, по данным которых может происходить как

локализация распознанных с камер объектов или же полноценное их детектирование.

2. Подготовленная карта окружающей среды, в которой должен функционировать робот, представленная в виде слоя, на котором может производиться геометрическая локализация (2D-сетка, 3D-облако точек и т.п.), и слоя, на котором размечены статические объекты, характеризующиеся положением, классом и визуальным образом (изображением или набором выделенных визуальных признаков, например посредством CLIP или аналогов).
3. При этом локализация должна происходить в среде, в которой находится достаточное число объектов, которые можно выделить и локализовать. Наиболее подходящие среды – комнатные и городские.

Выполнение данных пунктов позволит выполнять глобальную локализацию предложенной технологией, что будет показано в четвёртой главе.

2.4. Метрики определения качества решения задачи распознавания сцен

В подразделе 1.2.4 упоминались метрики для задачи распознавания сцен, использующиеся современными исследователями в этой области. Метрики вроде *Accuracy*, *Precision* и *Recall* требуют уточнения в разрезе задачи распознавания сцен. Сцена S (2.2) может содержать лишние объекты – выбросы, для которых нет соотношения на карте M (2.3). Обозначим истинное соответствие, удовлетворяющее (2.4), как \bar{f} , где выбросы в сцене соответствуют некоторому «нулевому» объекту на карте (Рис. 2.24).

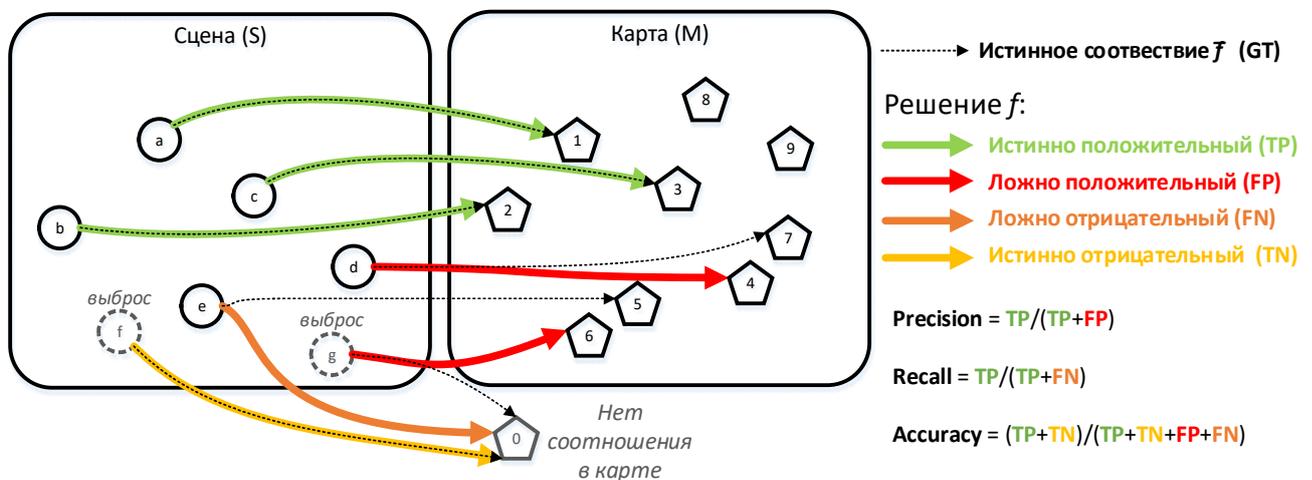


Рис. 2.24. Иллюстрация, описывающая метрики *Accuracy*, *Precision* и *Recall* для задачи распознавания сцены

Тогда для некоторого решения $f: S \rightarrow M$ можно определить указанные выше метрики через классификацию ответов:

$$TP(f, \bar{f}) = |\{f(o_i^S) = \bar{f}(o_i^S), \bar{f}(o_i^S) \neq o_0^M\}|, \quad (2.74)$$

$$TN(f, \bar{f}) = |\{f(o_i^S) = o_0^M\}|, \quad (2.75)$$

$$FP(f, \bar{f}) = |\{f(o_i^S) \neq \bar{f}(o_i^S), \bar{f}(o_i^S) \neq o_0^M\}|, \quad (2.76)$$

$$FN(f, \bar{f}) = |\{f(o_i^S) = o_0^M, \bar{f}(o_i^S) \neq o_0^M\}|, \quad (2.77)$$

где o_0^M и есть «нулевой» объект карты (соответствующий выбросам в сцене), тогда метрики принимают вид согласно общепринятым формулам:

$$Accuracy(f, \bar{f}) = \frac{TP(f, \bar{f}) + TN(f, \bar{f})}{TP(f, \bar{f}) + TN(f, \bar{f}) + FP(f, \bar{f}) + FN(f, \bar{f})}, \quad (2.78)$$

$$Precision(f, \bar{f}) = \frac{TP(f, \bar{f})}{TP(f, \bar{f}) + FP(f, \bar{f})}, \quad (2.79)$$

$$Recall(f, \bar{f}) = \frac{TP(f, \bar{f})}{TP(f, \bar{f}) + FN(f, \bar{f})}. \quad (2.80)$$

Важно отметить, что метрика *Recall* в данном случае имеет смысл только для данных, где имеются (смоделированы или размечены) выбросы, поскольку для данных без таких выбросов $Accuracy \equiv Precision$. На основе метрик *Precision* и *Recall* возможен расчет таких метрик, как *F-score* (смешение метрик *Precision* и *Recall*), а также построение графиков *PR-curve* (зависимость *Precision* от *Recall*), если варьировать параметры методов, например, τ (2.15), (3.2).

Данные метрики описывают лишь одно соответствие $f: S \rightarrow M$, однако методы распознавания сцен часто разработаны так, что возвращают набор таких соответствий $\{f\}$. Для наборов также требуются метрики оценки общего качества.

Для наборов необходимо указать не только сами соответствия f , но также их коэффициенты уверенности, и произвести упорядочивание по ним:

$$\tilde{f} = \{\{f_i, dc(f_i)\}\}: \forall i, dc(f_i) \geq dc(f_{i+1}). \quad (2.81)$$

Для оценки точности подобного набора используются метрики $Recall@k$ ($R@k$), где k – число лучших ответов из \tilde{f} , а сама метрика показывает нахождение верного ответа среди этого числа лучших:

$$recall@k(\tilde{f}, \bar{f}) = \begin{cases} 1, \bar{f} \in \{\tilde{f}_{i \in [1, k]}\} \\ 0, \bar{f} \notin \{\tilde{f}_{i \in [1, k]}\} \end{cases}. \quad (2.82)$$

Такая метрика применяется обычно к большому набору распознанных сцен и принимает вид среднего:

$$Recall@k(\{\tilde{f}\}, \bar{f}) = \frac{\sum_{\tilde{f}_i \in \{\tilde{f}\}} recall@k(\tilde{f}_i, \bar{f})}{|\{\tilde{f}\}|}. \quad (2.83)$$

Эта метрика позволяет оценить наличие правильного ответа в получаемом наборе гипотез. Поэтому предлагается метрика *Score*, показывающая и наличие, и положение истинно верного решения \bar{f} во всём наборе \tilde{f} :

$$Score(\tilde{f}, \bar{f}) = \begin{cases} \frac{1}{i}, f_i \in \tilde{f}, f_i = \bar{f} \\ 0, \bar{f} \notin \tilde{f} \end{cases}. \quad (2.84)$$

Она отражает способность методов определять корректный ответ на фоне общего набора ответов (2.81): чем ближе к 1 данное значение, тем точнее найден ответ.

Весь набор (2.81) можно определить с помощью предлагаемой метрики *Degree*, которая определяет тип такого набора среди следующих вариантов: *SuperTrue* – лучший ответ (с наибольшим *dc*) соответствует верному \bar{f} (т.е. *Score* = 1); *HasTrue* – верный ответ \bar{f} присутствует среди \tilde{f} , но не на первом месте; *Wrong* – верного ответа нет среди \tilde{f} при непустом \tilde{f} ; *NoResult* – множество \tilde{f} пусто:

$$Degree(\tilde{f}, \bar{f}) = \begin{cases} \text{"ИСТИНА"}, Score(\tilde{f}, \bar{f}) = 1, \\ \text{"ПРАВДОПОДОБО"}, \bar{f} \in \tilde{f}, Score(\tilde{f}, \bar{f}) \neq 1, \\ \text{"ЛОЖЬ"}, Score(\tilde{f}, \bar{f}) = 0, \tilde{f} \neq \emptyset, \\ \text{"ПУСТО"}, \tilde{f} = \emptyset. \end{cases} \quad (2.85)$$

Разбиение по таким подгруппам также показательно ввиду последующего применения к результатам распознавания сцен средств локализации. Это позволит оценить, насколько возможность нахождения верного ответа методами распознавания сцен влияет на локализацию робота. Данную метрику информативно применять к тестам на большом наборе сцен, как и (2.83), и анализировать размер групп данных значений.

Для всех формализованных и введённых метрик важным аспектом является такая разметка тестовых данных, чтобы было явно указан верный ответ \bar{f} включая соответствие выбросов нулевому объекту карты o_0^M .

2.5. Выводы к главе

Разработан метод распознавания сцен на основе алгоритмов теории графов: поиска изоморфного подграфа. Метод обладает всеми свойствами, которые были сформулированы исходя из специфики работы с семантическими объектами:

1. Определяются геометрические особенности взаиморасположения объектов при помощи составления графов, отражающих структуру и расстояния между объектами.
2. При сравнении расстояний между картой и сценой учитываются погрешности определения этих расстояний. Мера сравнения не зависит от абсолютных величин (2.9).
3. Учитывается визуальное сходство объектов, благодаря внедрению базисной нейросетевой модели CLIP.
4. Благодаря использованию алгоритмов теории графов, разработанный алгоритм распознавания сцен позволяет получать набор ответов, снабженный численной оценкой (коэффициенты уверенности (2.26)), который отражает как и удовлетворение геометрическим особенностям, так визуальному сходству.
5. Алгоритм адаптирован к выбросам – отсутствию объектов в сцене и карте, а также наличию лишних объектов. Данные ситуации также отражаются в расчёте коэффициента уверенности.

Эти свойства позволяют методу эффективно решать задачу распознавания сцен – сопоставления наблюдаемых объектов к семантической карте. Далее, после определения данного соответствия, становится возможной локализация на местности по визуальным ориентирам известными методами. Множественность гипотез, которые выдают разработанные методы, с одной стороны, решает проблему потери ответа в условиях, когда истинное решение оказывается близко к другим существующим.

Для определения положения по полученным от метода распознавания сцен данным, был применен механизм недоопределённых вычислений и формализована H-модель локализации (H-локализация). H-локализация оценивает положение робота в виде интервальных значений. Это позволяет не выполнять полный перебор пространства, как делается в ряде классических методов локализации. Вместо этого интервальные значения обновляются при помощи функций интерпретации, которые описывают связь положения робота с наблюдаемыми ориентирами. Был определён ряд функций присваивания, позволяющих настроить данный подход под различные имеющиеся данные.

Также важным элементом N-вычислений является наличие функций проверки корректности, которые могут определить противоречия в исходных данных. Расширив этот набор рядом дополнительных функций проверки корректности, удалось разрешить ситуации, которые не разрешались на этапе распознавания сцен, например, связанные с симметрией.

На основе предложенных методов распознавания сцен и N-локализации была предложена архитектура, позволяющая определить положение робота по визуальным ориентирам, а также уточнить его положение по данным дальномеров (облаков точек). Для этого предложены алгоритмы, позволяющие использовать полученные от N-локализации ограничения, для сокращения времени поиска классическим методом локализации по облакам точек.

Для оценки качества решения задачи распознавания сцен были предложены метрики, позволяющие оценить весь набор гипотез, которые выдают методы, а также требования к тестовым данным, чтобы эти метрики можно было реализовать. Всё это позволит сравнивать между собой различные методы распознавания сцен вне общей задачи локализации.

Глава 3. Реализация разработанных методов и моделей

3.1. Метод распознавания сцен на основе поиска изоморфного подграфа

Раздел посвящен реализации предложенного метода распознавания сцен, а также проверке работоспособности и постановке эксперимента для оценки эффективности, включая реализацию и модификации методов-аналогов.

3.1.1. Модификация метода распознавания сцен на основе поиска максимальной клики

Предложенный метод распознавания сцен (Табл. 2.7, Рис. 2.9) так или иначе базируется на алгоритмах, которые решают задачу поиска изоморфного подграфа, которая относится к классу NP-полных задач. Эта задача перекликается с задачей поиска максимальной клики (полносвязного подграфа максимального размера), которая также используется для задачи распознавания сцен [Ankenbauer и др., 2023; Matsuzaki, Tanaka, Shintani, 2024]. Отличием подхода на основе максимальной клики является то, что он не требует дополнительных расширений, чтобы учитывать выбросы. Вместо того, чтобы искать полное соответствие, в данном подходе рассматривается поиск наибольшего сходства. Тем не менее, поиск максимальной клики также относится к классу NP-полных задач. Для обоих подходов (поиска изоморфного графа и поиска максимальной клики) в плане вычислительной сложности, а также некоторых аспектов, связанных с точностью, решающей может оказаться конкретная реализация алгоритма. Поэтому целесообразно рассмотреть применение предложенных ранее механизмов оценки визуального сходства и геометрических особенностей в применении к методам распознавания сцен на основе поиска максимальной клики и провести численное сравнение полученных методов.

В качестве отправной точки был использован метод распознавания сцен на основе поиска максимальной клики, изложенный в работе [Ankenbauer и др., 2023]. Метод представляет собой построение графа специального вида, в котором в качестве вершин выступают все возможные соответствия объектов сцены к объектам карты, которые сохраняют класс объектов. Далее на этом графе две вершины соединяются рёбрами, если разница в расстоянии между объектами, соотносящимися с вершинами, не превышает некоторого порога (Рис. 3.1).

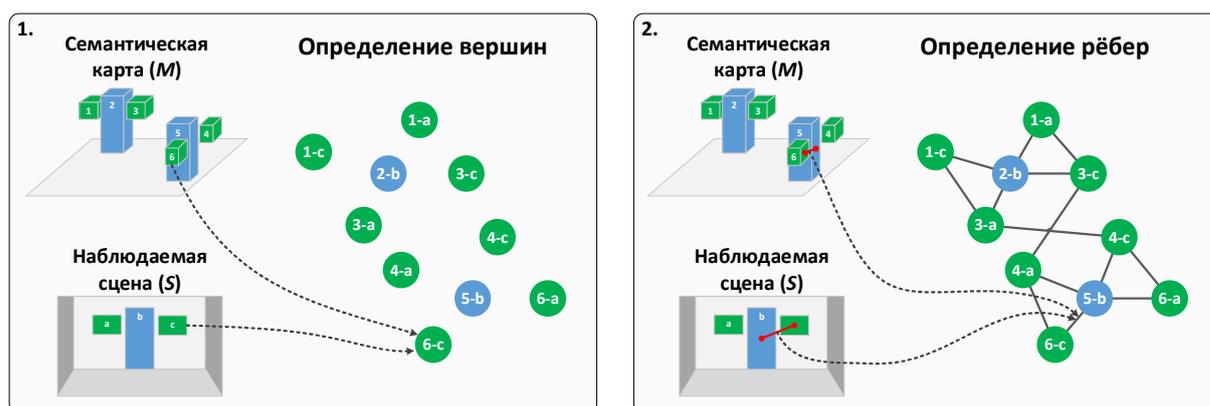


Рис. 3.1. Схема построения графа специального вида методом из работы [Ankenbauer и др., 2023]

На полученном графе (Рис. 3.1) происходит поиск максимальных клик, которые описывают наибольшее соответствие исходных множеств S (2.2) и M (2.3). Ограничения данного метода заключаются в том, что:

- не уточняется, как различить несколько вариантов ответов, если таковые есть;
- использование порога по разнице расстояний может привести к проблемам с крайними случаями, когда расстояния между объектами слишком большие или, наоборот, сопоставимы с этим порогом;
- не используются данные о погрешности измерения расстояний до объектов;
- в работе [Ankenbauer и др., 2023] не учитывалось визуальное сходство объектов.

Таким образом предлагается использование предложенной ранее меры сравнения расстояний с учётом их погрешности (2.9), интеграции средств определения визуального сходства на базе CLIP, и расчет коэффициента уверенности для каждого решения подобного (2.26).

Первый шаг предложенного метода, строящего граф G , остается прежним: все соответствия вершин S и M с совпадающим классом объектов c (2.1) добавляются как вершины:

$$G = \langle V, E \rangle, V = \{v_i = (o_i^S, o_i^M)\}, c_i^S = c_i^M. \quad (3.1)$$

Первой модификацией базового метода является интеграция меры сравнения расстояний (2.9), учитывающей погрешность определения расстояний до объектов (2.8):

$$E = \{(v_i, v_j)\}, p \left(D(o_i^S, o_j^S), D(o_i^M, o_j^M), \sigma D(o_i^S, o_j^S) \right) \geq \tau, \quad (3.2)$$

где τ настраиваемый параметр сходства отношений. Таким образом в граф добавляется ребро только между теми объектами, что удовлетворяют (3.2).

Полученный граф G может быть подан на вход алгоритмов поиска максимальной клики. Выход таких алгоритмов – набор наборов вершин $\{\{v_i\}_j\}$, образующих клики. Для полученного набора можно адаптировать расчёт коэффициента уверенности (2.26) с учётом оценки визуального сходства объектов:

$$dc(\{v_k\}) = \frac{1}{|\{v_k\}|} \sum_{((o_i^S, o_i^M), (o_j^S, o_j^M)) \in \{v_k\}} [u(H(o_i^S), H(o_i^M)) \cdot u(H(o_j^S), H(o_j^M)) \cdot p(D(o_i^S, o_j^S), D(o_i^M, o_j^M), \sigma D(o_i^S, o_j^S))]. \quad (3.3)$$

Алгоритм полученного метода представлен ниже:

Табл. 3.1. Псевдокод алгоритма распознавания сцен на основе поиска максимальной КЛИКИ

<pre> 1: 2: 3: 4: 5: 6: 7: 8: 9: 10: 11: 12: 13: 14: 15: 16: 17: 18: 19: 20: 21: 22: 23: 24: 25: 26: 27: 28: 29: 30: </pre>	<pre> Algorithm mc+_clip(S, M, τ, dc_min): -- S – сцена (множество объектов) -- M – семантическая карта (множество объектов) -- τ – порог сходства отношений -- dc_min – порог степени уверенности init V for s in S do for m in M do if $s.c = m.c$ then $v = (s, m)$ add v to V init E for $v1$ in V do for $v2$ in $V[v1:]$ do if $v1.s.c == v2.s.c$ then $r1, \sigma = \text{calc_dist_and_error}(v1.s, v2.s)$ $r2 = \text{calc_dist}(v1.m, v2.m)$ $p = \mathbf{P}(r2, r1, \sigma)$ -- (2.9) if $p < \tau$ then $e = (v1, v2)$ add e to E $G = (V, E)$ $F = \text{find_maximal_cliques}(G)$ init $Results$ for f in F do $dc = \text{calc_dc_clip}(f)$ -- (3.3) if $dc > dc_min$ then add (f, dc) to $Results$ sort $Results$ by dc return $Results$ </pre>
-----------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Алгоритм **mc+clip** (Табл. 3.1) принимает на вход множества карты M и сцены S , порог сходства отношений τ и порог степени уверенности dc_{min} . Функции **calc_dist_and_error** и **calc_dist** также считают расстояния между объектами, снабжая их среднеквадратичной ошибкой в первом случае. Функция **P** выполняет расчет согласно (2.9), функция **find_maximal_cliques** выполняет поиск наибольших клик в получившемся двудольном графе, функция **calc_dc_clip** рассчитывает коэффициент уверенности согласно (3.3), (2.26). Алгоритм возвращает отсортированный массив соответствий $f:S \rightarrow M$ с указанием их коэффициентов уверенности. Т.к. задача поиска максимальной клики также относится к классу NP-полных задач, то вычислительная сложность алгоритма **mc+clip** в худшем случае оценивается как $O(e^N)$, где N число элементов построенного графа.

Схема данного алгоритма представлена на Рис. 3.2.

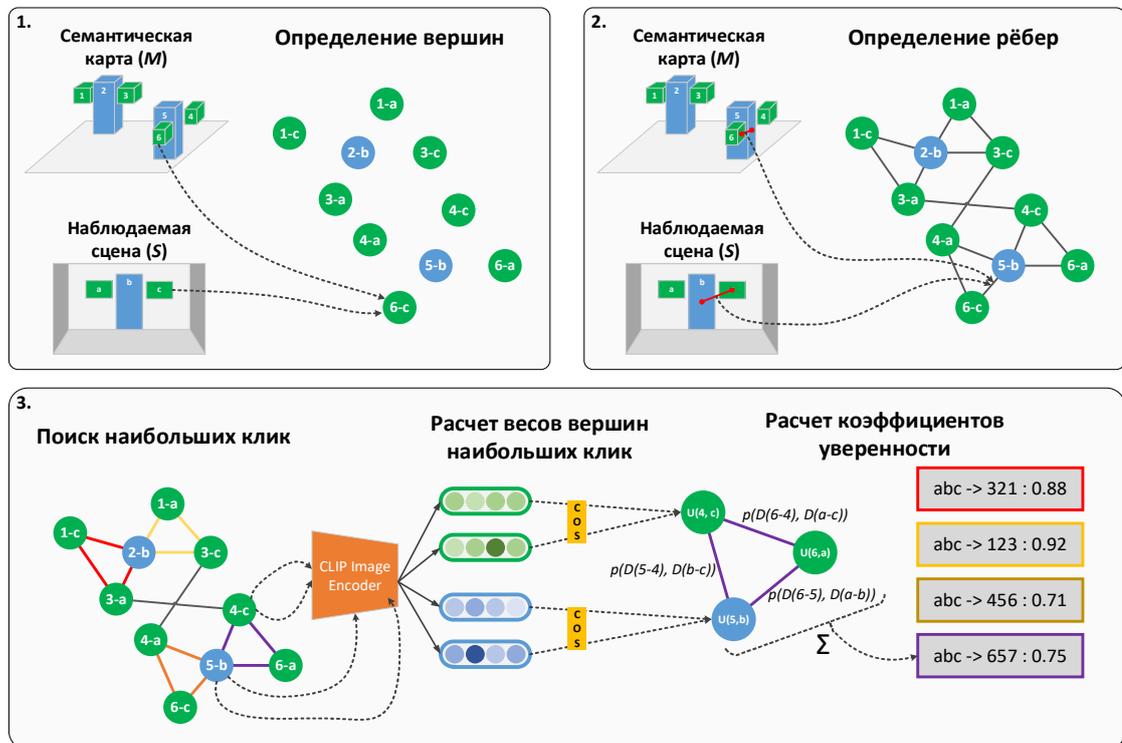


Рис. 3.2. Схема метода распознавания сцен на основе поиска максимальной клики

Таким образом, был модифицирован известный метод распознавания сцен, который в дальнейшем позволит провести более полное исследование предложенного нового метода, а также получить дополнительную информацию насколько вероятностная мера оценки отношений и учет визуального сходства повышают точность методов распознавания сцен.

3.1.2. Деление семантической карты на подкарты

Алгоритмы распознавания сцен, описанные в работе, основаны на методах из теории графов, которые традиционно относятся к классу NP-полных задач: это касается как поиска изоморфных подграфов, так и поиска максимальной клики. Ввиду экспоненциальной зависимости между временем исполнения и количеством объектов на исходном графе, целесообразно будет рассмотреть разбиение исходной карты на кластеры (подкарты) и производить распознавания сцен на них. Это обосновано тем, что идеологически сумма экспонент в степени некоторых чисел меньше экспоненты в степени суммы всех этих чисел, если эти числа превосходят единицу, что автоматически выполняется для семантических карт, насчитывающих десятки и сотни объектов:

$$e^{A+B+C} > e^A + e^B + e^C, \forall A, B, C > 1. \quad (3.4)$$

Также, имея подкарты, есть возможность обрабатывать их параллельно, что также позволит уменьшить время исполнения. Проблемой является наличие сцен, объекты которых будут иметь истинное соответствие в разных подкартах, что может привести к падению точности вблизи границ карт. Чтобы этого избежать, предлагается разбивать семантическую карту на подкарты с пересечениями такого размера, чтобы объекты сцены целиком попадали в одну из подкарт.

Для реализации такого разбиения выполняются следующие шаги (Рис. 3.3):

1. Производится начальная иерархическая кластеризация [Жамбю, Миркин, 1988] объектов карты. Число кластеров определяется на основе желаемого размера подкарты. Определяются центроиды кластеров.
2. Для каждого кластера рассчитываются диаграммы направленности. С некоторым шагом вокруг центроида определяются секторы, для каждого сектора выбирается наиболее удаленный от центроида объект и рассчитывается расстояние до него.
3. Далее все полученные расстояния увеличиваются на рабочую дальность сенсора дистанции, например, лидара или стереокамеры.
4. Все объекты других начальных кластеров, попавшие в расширенный сектор, добавляются к текущему.

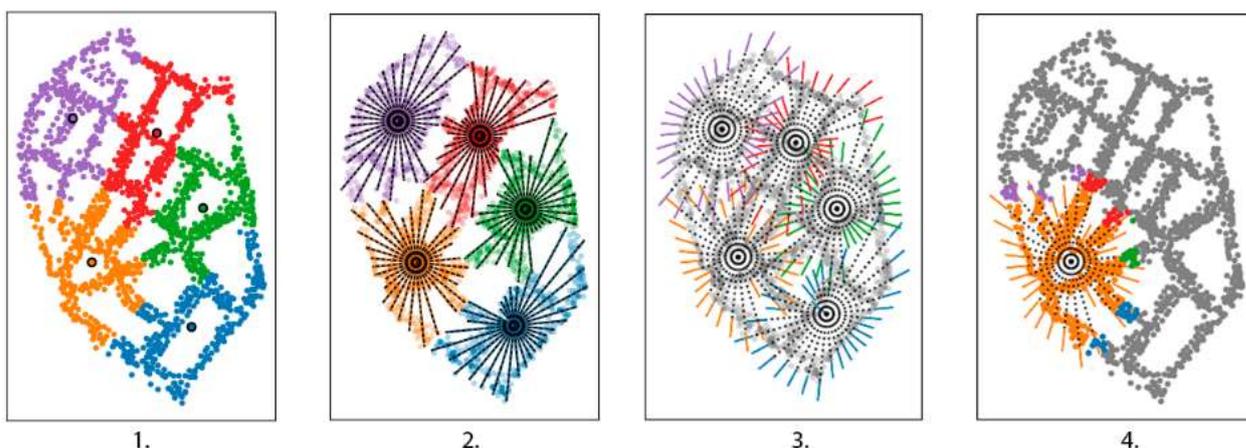


Рис. 3.3. Пример разделения на подкарты. 1 – начальная кластеризация и определение центроидов, 2 – построение диаграмм направленности, 3 – расширение диаграмм направленности, 4 – добавление новых объектов к одному из кластеров

Алгоритм деления семантической карты представлен ниже:

Табл. 3.2. Псевдокод алгоритма деления семантической карты на подкарты

1.	Algorithm <code>divide_semantic_map</code> (M , <code>submap_size</code> , <code>n_sector</code> , D_{sens})
2.	-- M - семантическая карта (множество объектов)
3.	-- <code>submap_size</code> - желаемый размер подкарт
4.	-- <code>n_sector</code> - число секторов
5.	-- D_{sens} - дальность сенсора
6.	$n_{cluster} = \text{ceil}(M / \text{submap_size})$
7.	$M' = \text{h_cluster}(M, \text{submap_size})$
8.	<code>init submaps</code>
9.	<code>for</code> m <code>in</code> M' <code>do</code>
10.	<code>init submap</code>
11.	$center = \text{mean}(m)$
12.	$\alpha_{start} = 0$
13.	$\alpha_{end} = \text{step} = 2\pi / n_{sector}$
14.	<code>for</code> i <code>in</code> $0:n_{cluster}$ <code>do</code>
15.	$m'' = \text{select_sector}(m, center, \alpha_{start}, \alpha_{end}, \infty)$
16.	$d = \text{max}(m'' - center)$
17.	<code>add</code> <code>select_sector</code> (M , $center$, α_{start} , α_{end} , $d + D_{sens}$) <code>to</code>
	<code>submap</code>
18.	$\alpha_{start} = \alpha_{start} + \text{step}$
19.	$\alpha_{end} = \alpha_{end} + \text{step}$
20.	<code>add submap</code> <code>to</code> <code>submaps</code>
21.	<code>return submaps</code>

Алгоритм `divide_semantic_map` (Табл. 3.2) принимает на вход семантическую карту M , а также параметры деления: примерный желаемый

размер подкарты $submap_size$, число разбиений на сектора n_sector и дальность сенсора D_{sens} . Функция **ceil** выполняет округление вверх до целого числа, функция **select_sector** выбирает объекты из множества попадающих в сектор, задающийся центром, углами и расстоянием. Поскольку самым вычислительно сложным моментом данного алгоритма является иерархическая кластеризация, то полученный алгоритм обладает такой же сложностью – $O(N^2 \log N)$, где N число объектов карты.

Подобное разбиение может ускорить выполнение представленных ранее алгоритмов распознавания сцен, однако это зависит от размера перекрытий подкарт t , который должен удовлетворять изменённому выражению (3.4):

$$e^{A+B+C} > e^{A+t_A} + e^{B+t_B} + e^{C+t_C}, \forall A, B, C > 1. \quad (3.5)$$

Результаты исследования влияния предложенного алгоритма на реальных данных приведены в разделе 4.1.

3.1.3. Программная реализация методов распознавания сцен

Программные средства, реализующие алгоритмы распознавания сцен были реализованы на языке Python3. Класс **SemanticMap** реализует семантическую карту, обеспечивает функции доступа к ней, загрузки из файлов, в том числе в виде подкарт. Основной класс **SceneRecognition** реализует базовый функционал (сортировка решений по коэффициенту уверенности, отрисовка входных данных, расчёт общих значений, например (2.8) и (2.9) и метрик оценки качества на основе верного ответа (раздел 2.4)) и задаёт вид функций которые должны быть унаследованы классами-реализациями методов, среди которых запуск процесса распознавания, в т.ч. с ограничением по времени работы. Были реализованы следующие алгоритмы как отдельные классы:

- Метод на основе алгоритма поиска изоморфного графа с прямым сравнением рёбер и мягким расширением (Табл. 2.8).
- Метод на основе алгоритма поиска изоморфного графа с кластеризацией отношений (Табл. 2.6) и мягким расширением.
- Оригинальный метод на основе алгоритмов поиска максимальной клики из работы [Ankenbauer и др., 2023] и его предложенная модификация с вероятностным сравнением отношений и визуальным сходством (Табл. 3.1).
- Метод распознавания сцен на основе дескрипторов случайного блуждания из работы [Gawel и др., 2018; Liu и др., 2019a].

- Метод распознавания сцен исключительно на основе визуального сходства CLIP-Clique [Matsuzaki, Tanaka, Shintani, 2024].

Вся работа с графами, в т.ч. алгоритмы поиска изоморфного подграфа и максимальной клики, была выполнена с помощью библиотеки *igraph*³. Общая структура классов позволила реализовать оптимизацию параметров и тестирование указанных методов.

3.1.4. Исследование в среде имитационного моделирования

Для первичного исследования работоспособности предложенного метода распознавания сцен была реализована симуляция на языке Python3, имитирующая процедуру распознавания объектов, и формирующая вход для исследуемых методов в виде семантической карты M (2.3) и сцены S (2.2). Сцена формировалась на основе положения робота, угла обзора (1.5 рад) и дальности (15 м) эмулированного сенсора. В расстояние до объекта и угол на него добавлялся Гауссовский шум (со стандартным отклонением 0.1 м и 0.01 рад и нулевым средним) для имитации погрешности систем распознавания. Также эмулировались выбросы в сцене путём изъятия случайных объектов из карты (после формирования сцены). Число выбросов принимало случайное значение между 5 и 15% всех объектов карты. Семантическая карта размером 100м на 100м условно описывала окружение здания Вычислительного центра НИЦ «Курчатовский институт» и содержало 62 объекта пяти различных классов.

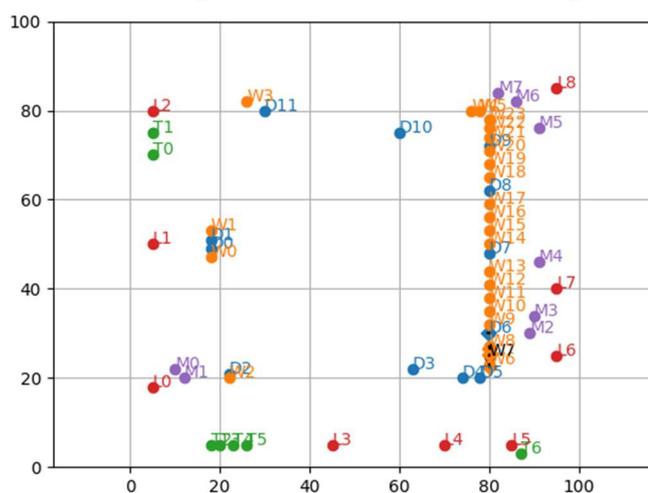


Рис. 3.4. Семантическая карта, используемая в исследовании

Исследовались следующие методы:

³ <https://igraph.org/>

- Метод на основе дескрипторов случайного блуждания *RWD*, реализованный по описанию из работ [Gawel и др., 2018; Liu и др., 2019a].
- Оригинальный метод на основе поиска максимальной клики *MC*, реализованный по описанию [Ankenbauer и др., 2023].
- Предложенное расширение метода на основе поиска максимальной клики *MC+* (раздел 3.1.1).
- Метод на основе поиска изоморфного подграфа с кластеризацией отношений (раздел 2.1.5) и мягким расширением *IG*.

Методы не использовали визуальное сходство, поскольку объекты не имели визуального образа *I*. Методы тестировались в трёх разных сценариях:

- *clear* – в данных отсутствовали выбросы;
- *holes* – во всех данных присутствовали выбросы;
- *mix* – первые два сценария были смешаны в равной пропорции.

Для каждого сценария было сгенерировано 500 сцен на основе случайного положения робота. Также для каждого сценария было сгенерировано по 100 сцен, на которых происходила оптимизация параметров методов средством *optuna*⁴. В качестве значения для максимизации был выбран усредненный критерий *Precision* (2.79) лучшего ответа. Полученные параметры представлены в Табл. 3.3.

Табл. 3.3. Оптимизированные параметры исследуемых методов

Метод	Сценарий	Параметры
<i>RWD</i>	<i>clear</i>	$d_{near}=8.66м, len=4, depth=100, spc = 0.47м, k = 5, N_{try}=300$
	<i>holes</i>	$d_{near}=10м, len=4, depth=100, spc = 0.39м, k = 5, N_{try}=300$
	<i>mix</i>	$d_{near}=9.8м, len=4, depth=100, spc = 0.51м, k = 5, N_{try}=300$
<i>MC</i>	<i>clear</i>	$d=0.55м$
	<i>holes</i>	$d=0.34м$
	<i>mix</i>	$d=0.58м$
<i>MC+</i>	<i>clear</i>	$\tau=0.45$
	<i>holes</i>	$\tau=0.34$
	<i>mix</i>	$\tau=0.46$
<i>IG</i>	<i>clear</i>	$\tau_c = 0.39, \tau_m=0.29$
	<i>holes</i>	$\tau_c = 0.31, \tau_m=0.5$

⁴ <https://optuna.org/>

	<i>mix</i>	$\tau_c = 0.46, \tau_m = 0.28$
--	------------	--------------------------------

Для метода *RWD* d_{near} обозначает максимальное расстояние между объектами, которые могут считаться соседними, len и $depth$ обозначают длину и глубину дескрипторов, spr – порог для ошибки по расстоянию между совпавшими дескрипторами, N_{try} – число попыток построения дескрипторов. Для метода *MC* d – это порог схожести отношений в метрах.

На полученных параметрах для каждого сценария исследовалась точность методов, измеряемая по критерию *Degree* (2.85), результаты представлены на Рис. 3.5 и в Табл. 3.4.

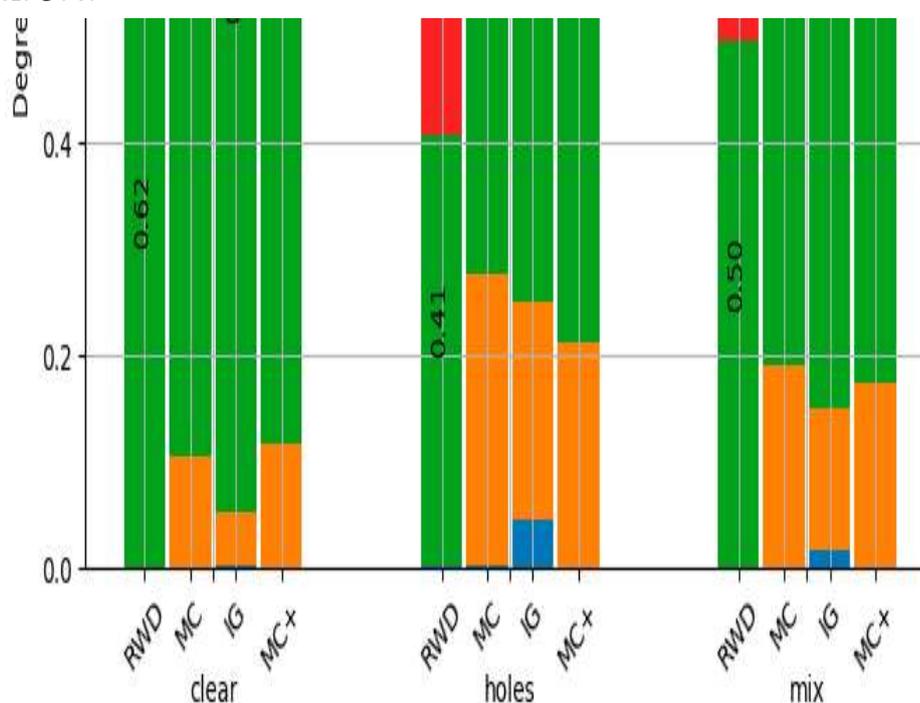


Рис. 3.5. Распределение значений *Degree* для исследованных методов и сценариев

Также к лучшему результату распознавания сцен для каждого метода был применен метод локализации SVD на основе сингулярного разложения [Arup, Huang, Blostein, 1987], сопоставляющий два набора точек, значения ошибки определения положения ΔT также представлены в Табл. 3.4.

Табл. 3.4. Численные результаты исследованных методов

Метод\Сценарий	Критерий	<i>clear</i>	<i>holes</i>	<i>mix</i>
<i>RWD</i>	%ИСТИНА, ↑:	61.6	40.6	50.4
	ΔT , м, ↓:	166.9	355.9	203.8
<i>MC</i>	%ИСТИНА, ↑:	88.2	51.0	71.6
	ΔT , м ↓:	0.015	0.015	0.019

<i>IG</i>	%ИСТИНА, ↑:	93.6	65.0	78.6
	ΔT , м ↓:	0.014	0.022	0.013
<i>MC+</i>	%ИСТИНА, ↑:	86.2	65.4	75.2
	ΔT , м ↓:	0.014	0.017	0.013

Основной вывод по результатам экспериментального исследования заключается в том, что подходы, основанные на методах аналитического сравнения графов, превосходят вероятностные. Метод на основе поиска изоморфного подграфа даёт лучшие показатели в смешанном режиме, который наиболее вероятен в реальном применении.

3.2. Метод Н-локализации

Раздел посвящен исследованию предложенного метода Н-локализации, описанного в разделе 2.2. Стоит отметить, что этот раздел рассматривает вопросы локализации по ориентирам без привязки к задаче распознавания сцен, считая, что соответствие либо найдено верно, либо используются уникальные ориентиры, для которых не требуется решать задачу распознавания сцен. Вопросы исследования работы Н-локализации по данным от задач распознавания сцен рассматриваются в разделе 3.3, посвященному комплексной технологии глобальной локализации.

3.2.1. Программная реализация

Для исследований с предложенной моделью Н-локализации все приведённые в третьей главе алгоритмы были реализованы на языке Python3, и получено свидетельство о регистрации РИД [Московский, 2019b]. Помимо алгоритмов Н-локализации программный комплекс содержит реализацию других методов локализации по ориентирам. Основной класс **LandmarkLocalization** содержит структуру для всех наследуемых классов, реализующих алгоритмы:

- Предложенной в работе Н-локализации (Табл. 2.10 и Табл. 2.11).
- Локализации на основе ФЧ (AMCL).
- Локализация с помощью ГФ.
- Локализация методов SVD.

Реализация Н-локализации позволяет внутри полученных ограничений проводить уточнение методами АМСЛ и ГФ. Для реализации всего механизма Н-вычислений были реализованы следующие дополнительные классы:

- **sd_var** реализует интервальную Н-переменную и интервальную алгебру связанную с ней, часть которой реализована с помощью библиотеки IntervalPy⁵ [Androsov, Shary, 2022].
- **sd_mi** реализует Н-переменную заданную в виде мульти-интервала (2.32) на основе интервалов **sd_var**, а также связанные с ней функции упорядочивания и интервальной алгебры.
- **SDM** реализует формальную Н-модель, с возможностями регистрации функций интерпретации и проверки корректности, а также связанные вычислительные процедуры, в т.ч. основной вычислительный процесс (Табл. 2.9) и способы его остановки.

Для взаимодействия с системой ROS Noetic реализован класс **LandmarkLocalizationRos2D** осуществляющий прием-передачу данных класса **LandmarkLocalization** с другими частями системы, включая:

- получение результатов распознавания объектов;
- публикацию положения робота в виде распространенных сообщений, а также средства tf;
- визуализацию процессов локализации, совместимых со средством rViz.

3.2.2. Исследование в среде имитационного моделирования

Для определения основных характеристик и возможностей метода Н-локализации была разработана имитационная среда на языке Python3 (Рис. 3.6).

⁵ <https://pypi.org/project/intvalpy/>

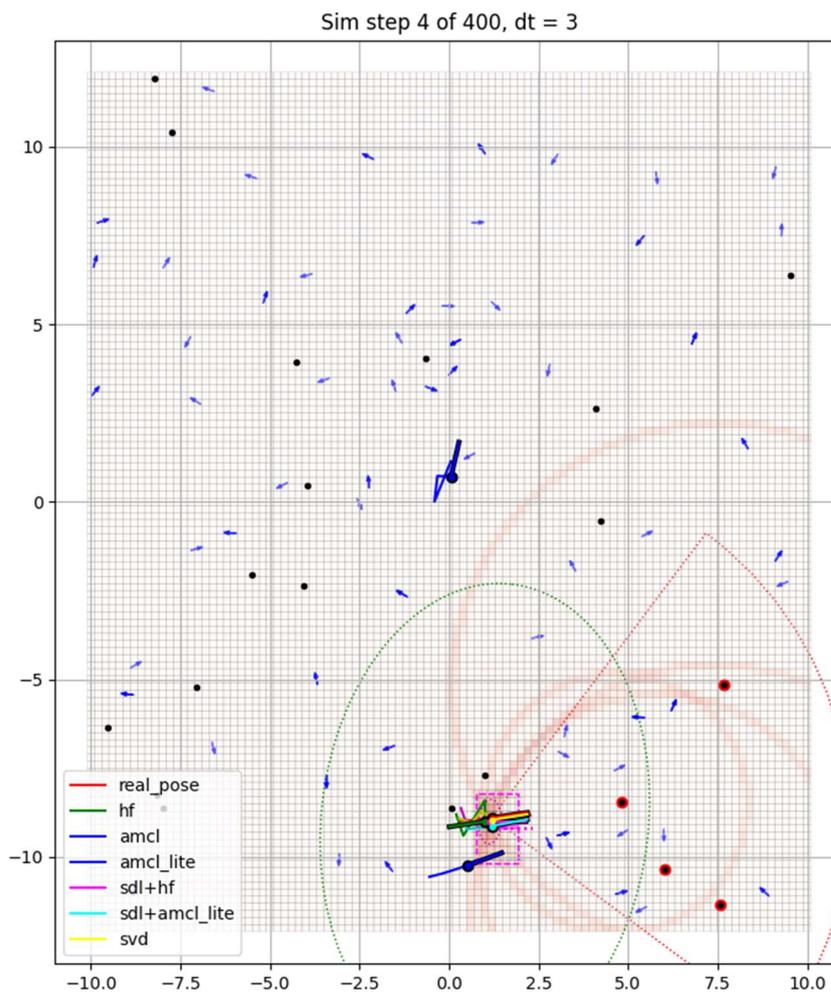


Рис. 3.6. Визуализация разработанной среды имитационного моделирования для проведения экспериментов с методами локализации: разными цветами указано текущее положение, рассчитанное разными методами (см. ниже)

В данной среде моделируется равномерное движение робота по окружности в поле ориентиров с заданным временным шагом. Моделируется сенсор робота, способный определять расстояние и/или угол на объект с погрешностью в виде нормального распределения с нулевым математическим ожиданием и заданным стандартным отклонением. Общие параметры имитационного моделирования приведены в Табл. 3.5.

Табл. 3.5. Общие параметры имитационного моделирования

Описание параметра	Значение параметра
Размер поля	10x12 м
Число ориентиров	20
Временной шаг	3 с
Число шагов	400

Линейная скорость робота	0.1 м\с
Угловая скорость робота	0.012 рад\с
Шум одометрии линейной скорости	0.01 м\с
Шум одометрии угловой скорости	0.01 рад\с
Ошибка измерения расстояний	0.1 м
Ошибка измерения угла	0.01 рад

Помимо общих параметров, приведённых в Табл. 3.5, были определены параметры для трёх разных сценариев локализации по ориентирам. Следующие сценарии рассматривались в поставленных экспериментах:

- *Camera*: локализация по данным, имитирующим данные с камеры робота, характеризуется возможностью замерять как расстояние до объекта, так и угол на него в некотором ограниченном секторе.
- *LBL (Long Baseline)*: локализация по ориентирам, до которых можно определить только расстояние в полном секторе (360°). Имитирует подводные гидроакустические системы локализации *LBL*.
- *Bearing*: локализация с возможностью определить только угол на ориентир в полном секторе, имитирует системы радио-локализации «по пеленгу».

Численные значения параметров приведены в Табл. 3.6.

Табл. 3.6. Параметры имитационного моделирования для разных сценариев эксперимента

Метод	Расстояние до ориентира		Угол на ориентир	
	Измеряемое	Ограничение	Измеряемое	Ограничение
<i>Camera</i>	✓	10 м	✓	$[-\pi/4, \pi/4]$
<i>LBL</i>	✓	20 м	✗	$[-\pi, \pi]$
<i>Bearing</i>	✗	20 м	✓	$[-\pi, \pi]$

Для тестирования метода были выбраны следующие аналоги и их вариации:

- *svd*, оптимизационный метод SVD сопоставления двух наборов точек [Arun, Huang, Blostein, 1987], рассматривается только в сценарии *Camera*, поскольку по данным других сценариев невозможно восстановить набор точек;
- *hf*, вероятностный подход на основе ГФ [Thrun, Wolfram, Fox, 2005];
- *sdl_hf*, определение интервальных ограничений методом Н-локализации и применение метода *hf* в полученных ограничениях;

- *amcl*, вероятностный метод на основе ФЧ [Thrun, Wolfram, Fox, 2005] (число частиц в диапазоне 1000-2000);
- *amcl_lite*, вероятностный метод на основе ФЧ (число частиц в диапазоне 100-200);
- *sdl_amcl_lite*, определение интервальных ограничений методом Н-локализации и запуск метода *amcl_lite* в полученных ограничениях;

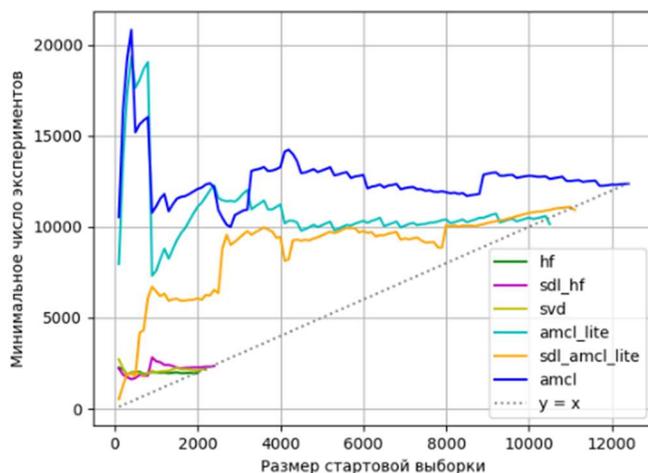
Для методов *sdl_amcl_lite* и *sdl_hf* использовался алгоритм непрерывной Н-локализации (**sdm_continuous_localization**, Табл. 2.11), а для распределения частиц в *sdl_amcl_lite* – алгоритм **restrictions_to_particles** (Табл. 2.14). В методе *sdl_amcl_lite* на каждом новом шаге после расчета новых ограничений оставались частицы, принадлежащие этим ограничениям, в то время как остальные заново перераспределялись. ГФ был реализован классическим образом, разбивая исследуемую область на ячейки с некоторым шагом по переменным X , Y и θ . В реализации *hf* значения, рассчитанные на предыдущем шаге, смещались и «размывались» согласно данным одометрии, и на новом шаге учитывались с пониженным коэффициентом. В реализации *sdl_hf* на каждой итерации запускался новый ГФ в полученных ограничениях. Параметры исследованных методов приведены в Табл. 3.7.

Табл. 3.7. Параметры исследованных методов

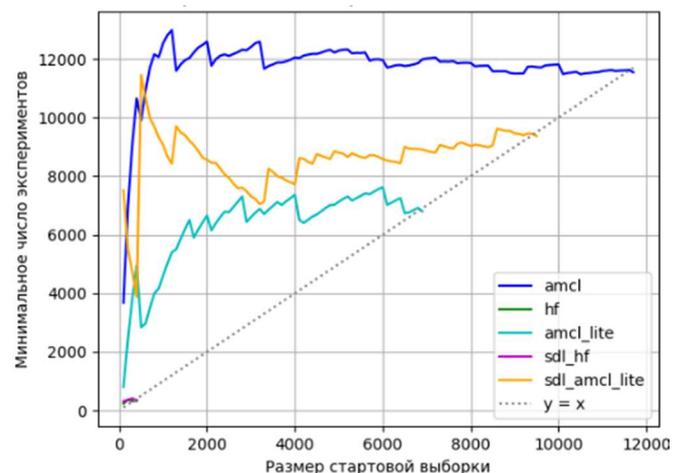
Метод	Параметр	Значение	Описание
<i>svd</i>	-	-	Метод не имеет параметров
<i>amcl</i>	N_p	1000	Стартовое число частиц
	N_{p_max}	2000	Максимальное число частиц
	$\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6$	0.1, 0.1, 0.1, 0.1, 0.1, 0.1	Параметры изменения частиц при смещении робота
	$\alpha_{fast}, \alpha_{slow}$	0.1, 0.05	Параметры оценки своего состояния для принятия решения об изменении числа частиц
	<i>calc_type</i>	MULT.	Способ одновременного учета весов из разных источников (сложение\умножение)
	<i>final_choise</i>	MEAN	Выбор финального положения (взвешенное среднее)
<i>amcl_lite</i>	N_p	100	Стартовое число частиц
	N_{p_max}	200	Максимальное число частиц
	-	-	Остальные параметры соответствуют методу <i>amcl</i>
<i>hf</i>	<i>xy_d_res</i>	0.2 м	Шаг дискретизации для координат X и Y
	<i>Y_d_res</i>	0.15 рад	Шаг дискретизации для θ
	<i>prev_step_weight</i>	0.5	Коэффициент весов ячеек с предыдущего шага
	<i>calc_type</i>	ADD.	Способ одновременного учета весов из разных источников (сложение\умножение)

	<i>final_choise</i>	MAX	Выбор финального положения (максимальная ячейка)
	<i>yaw_discount</i>	0.1	Коэффициент весов, рассчитанных по данным угла на ориентиры
	<i>sdl_hf</i>	-	Параметры соответствуют методу <i>hf</i>
	<i>sdl_<...></i>	<i>R (Camera)</i>	Функции интерпретации для сценариев <i>Camera</i>
		(2.41)- (2.45)	
		<i>R (LBL)</i>	Функции интерпретации для сценариев <i>LBL</i>
		(2.42),(2.43)	
		<i>R (Bearing)</i>	Функции интерпретации для сценария <i>Bearing</i>
		(2.41),(2.44),(2.45),(2. 63)-(2.68)	Функции (2.63)-(2.68) были сформированы для одной пары ориентиров, максимально удаленных друг от друга по углу
		<i>C' (Camera)</i>	Дополнительные функции проверки корректности
		(2.50), (2.51)	
	<i>max_steps</i>	100	Ограничение числа итераций для алгоритма sdm_proc (Табл. 2.9)
	<i>stop_acc</i>	0.01	Ограничение точности N-переменных при обновлении (в случае обновления менее данного порога, считается, что не было)
	<i>var_acc</i>	3	Число разрядов N-переменных для округления
	<i>sigma_mul</i>	4	Значение $N\sigma$ в (2.37)

Для каждого сценария было произведено несколько серий вычислений, каждая серия содержала 400 шагов движения робота в случайно сгенерированном поле, где для каждого шага происходила локализация выбранными методами. Замерялись ошибки по расстоянию (r), модулю угла ($|\alpha|$) и времени исполнения (t). Общее число требуемых экспериментов оценивалось с помощью t-критерия Стьюдента для каждого из замеряемых критериев, для всех методов и в рамках каждого сценария. Расчет происходил каждые 100 шагов эксперимента с точностью измерений 0.05 и доверительной вероятностью 0.95. Для данных параметров было оценено минимальное число экспериментов (Рис. 3.7).



а)



б)

Рис. 3.7. Пример расчётов минимального числа экспериментов для определения:
 а) ошибки положения в сценарии *Camera*, б) ошибки модуля угла в сценарии
Bearing

Для каждого сценария выбирался наибольший результат минимального числа экспериментов для измеряемых величин и округлялся в большую сторону с учетом взятого шага. Итого для сценария *Camera* было взято 13700 шагов, для сценария *LBL* 11700 шагов и для *Bearing* 19500 шагов.

Для измеряемых параметров r , $|\alpha|$, t были рассчитаны средние значения и среднеквадратичные ошибки, а также значения первых трех квартилей (где $Q2$ – это медианное среднее). Численные результаты экспериментов для исследуемых алгоритмов в сценарии *Camera* приведены в Табл. 3.8.

Табл. 3.8. Численные результаты экспериментов для сценария *Camera*

Метод	$\bar{r} \pm \sigma$, м, ↓	$Q1-Q2-Q3(r)$, м, ↓	$ \bar{\alpha} \pm \sigma \alpha $, рад, ↓	$Q1-Q2-Q3(\alpha)$, рад, ↓	$\bar{t} \pm \sigma$, с, ↓	$Q1-Q2-Q3(t)$, с, ↓
<i>svd</i>	2,73±4,23	0,08-0,22-4,19	0,64±0,90	0,01-0,04-1,18	0,0004±0,0002	3-4-5 (x10⁻⁴)
<i>hf</i>	2,30±3,32	0,26-0,60-3,08	2,60±0,77	2,38-3,00-3,08	0,26±0,09	0,19-0,25-0,31
<i>sdl hf</i>	0,63±1,26	0,17-0,27-0,51	0,26±0,44	0,08-0,13-0,23	0,01±0,01	0,004-0,01-0,01
<i>amcl</i>	0,21±0,70	0,07-0,09-0,13	0,04±0,13	0,01-0,02-0,03	0,023±0,012	0,01-0,02-0,04
<i>amcl lite</i>	0,50±1,57	0,08-0,11-0,15	0,10±0,34	0,01-0,02-0,03	0,002±0,001	0,001-0,002-0,003
<i>sdl amcl lite</i>	0,30±0,95	0,06-0,10-0,16	0,06±0,20	0,01-0,02-0,03	0,007±0,004	0,005-0,01-0,01

По данным, представленным в Табл. 3.8, метод *amcl* показывает лучшие результаты по ошибке положения, однако предлагаемый метод *sdl_amcl_lite* показывает близкие результаты по ошибке положения и модуля угла, но при этом быстрее практически на порядок. Одновременно на примере метода *amcl_lite*, который имеет такое же число частиц, как *sdl_amcl_lite*, видно повышение точности определения положения у *sdl_amcl_lite*. Варианты на основе ГФ отстают как по точности, так и по времени исполнения от методов на основе ФЧ. В то же время видно преимущество Н-локализации на примере различий в значениях *hf* и *sdl_hf* – были значительно увеличены и точность, и время исполнения. Сверхбыстрый метод *sdv* в задаче непрерывной локализации не показывает хороших значений по точности, поскольку не имеет особенностей «фильтра», позволяющих сглаживать разброс значений между шагами.

Для сценария *LBL*, где методы получают только данные о расстоянии до ориентиров, результаты экспериментов представлены в Табл. 3.9.

Табл. 3.9. Численные результаты экспериментов для сценария *LBL*

Метод	$\bar{r} \pm \sigma$, м, ↓	$Q1-Q2-Q3(r)$, м, ↓	$ \bar{\alpha} \pm \sigma \alpha $, рад, ↓	$Q1-Q2-Q3(\alpha)$, рад, ↓	$\bar{t} \pm \sigma$, с, ↓	$Q1-Q2-Q3(t)$, с, ↓
<i>hf</i>	0,17±0,08	0,11-0,16-0,23	1,60±0,91	0,81-1,60-2,39	0,13±0,01	0,12-0,13-0,13
<i>sdl hf</i>	0,37±0,12	0,3-0,38-0,46	1,65±0,93	0,83-1,71-2,48	0,016±0,002	0,01-0,02-0,02
<i>amcl</i>	0,1±0,63	0,02-0,03-0,05	0,1±0,31	0,02-0,04-0,07	0,02±0,01	0,01-0,01-0,02
<i>amcl lite</i>	0,36±1,44	0,02-0,04-0,06	0,2±0,52	0,02-0,05-0,09	0,002±0,001	0,001-0,001-0,002
<i>sdl amcl lite</i>	0,04±0,02	0,02-0,03-0,05	0,07±0,20	0,02-0,04-0,07	0,014±0,002	0,01-0,01-0,02

В сценарии *LBL* (Табл. 3.9) минимальную ошибку положения и угла показывает предложенный метод Н-локализации с ФЧ внутри полученных ограничений (*sdl_amcl_lite*), однако превышение на порядок в точности над *amcl_lite* достигается падением на порядок скорости вычислений. Также стоит отметить возможность ФЧ определять угол робота в движении, несмотря на отсутствие данных об угле на ориентиры, в то время как у методов на основе ГФ ошибка модуля угла практически равномерно распределена в диапазоне $[0, \pi]$. В сценарии *LBL* также стоит отметить возросшую ошибку метода *sdl_hf* в сравнении с *hf*, что расходится с тенденцией, полученной в сценарии *Camera* (Табл. 3.8). Происходит это из-за того, что построенная по входным данным область содержит симметричные положения робота относительно ориентиров (Рис. 2.15). Однако ввиду отсутствия данных об углах, невозможно применить дополнительную функцию проверки корректности (2.51), из-за этого метод может «скакать» от одного локального минимума к другому. Метод *hf* не имеет такой проблемы, поскольку содержит информацию о предыдущих положениях робота. Аналогично это имеется и у метода *sdl_amcl_lite*, где несмотря на расчет областей на каждом шаге, частицы, им удовлетворяющие, переносятся на новый.

Для сценария *Bearing*, где методы получают только данные об угле на ориентиры, результаты экспериментов представлены в Табл. 3.10.

Табл. 3.10. Численные результаты экспериментов для сценария *Bearing*

Метод	$\bar{r} \pm \sigma$, м, ↓	$Q1-Q2-Q3(r)$, м, ↓	$ \bar{\alpha} \pm \sigma \alpha $, рад, ↓	$Q1-Q2-Q3(\alpha)$, рад, ↓	$\bar{t} \pm \sigma$, с, ↓	$Q1-Q2-Q3(t)$, с, ↓
<i>hf</i>	0,67± 0,44	0,3-0,61-0,97	0,08± 0,05	0,04-0,08-0,12	0,73±77,34	1,21-1,26-1,28
<i>sdl hf</i>	0,86±1,77	0,33-0,55-0,85	0,09±0,19	0,04-0,06-0,09	0,31±0,53	0,02-0,02-0,23
<i>amcl</i>	0,92±2,32	0,10-0,16-0,28	0,14±0,51	0,002-0,01-0,01	0,03±0,02	0,01-0,02-0,04
<i>amcl lite</i>	5,09±4,01	0,34-7,57-8,55	0,93±1,04	0,01-0,47-1,81	0,002±0,001	0,001-0,002-0,002
<i>sdl amcl lite</i>	0,66±1,77	0,04-0,08-0,2	0,07±0,29	0,002-0,005-0,01	0,02±0,004	0,01-0,02-0,02

Сценарий *Bearing* является наиболее сложным из перечисленных, для методов на основе ФЧ требуется достаточно плотное покрытие области, чтобы точно определить местоположение. Падение точности методов *amcl* и *amcl_lite* в сравнении с *sdl_amcl_lite* можно видеть в данных из Табл. 3.10. Стоит также

отметить результаты методов *hf* и *sdl_hf*, у которых просматривается та же тенденция, что в сценарии *LBL*, где использование Н-локализации понижает точность. В данном случае точность падает не так значительно, как в сценарии *LBL*, но при этом достигается сильное уменьшение времени расчёта.

На Рис. 3.8 приведены данные для средней ошибки положения (\bar{r}) и среднего времени работы исследованных методов (\bar{t}).

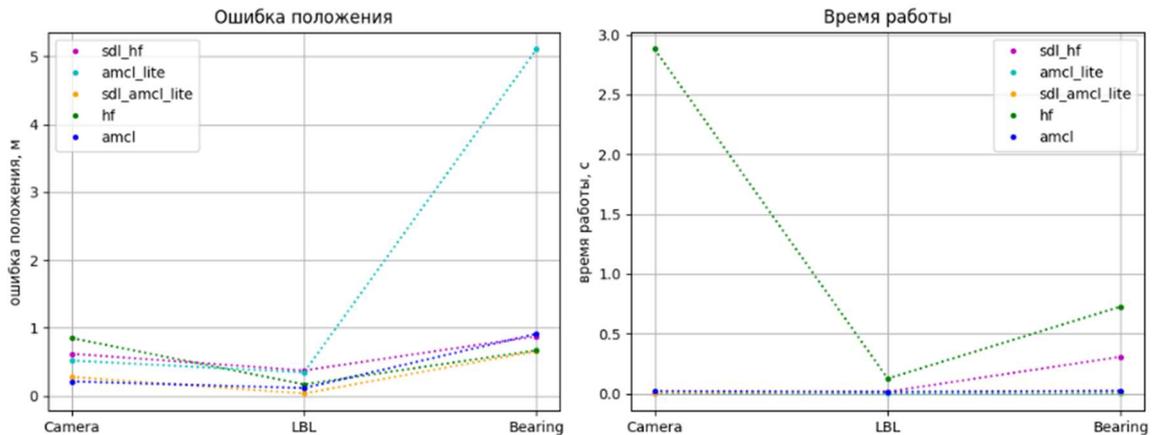


Рис. 3.8. Сравнительные графики для исследованных методов для различных сценариев

По результатам проведённых экспериментов можно сделать следующие выводы:

Метод Н-локализации обладает заявленными свойствами по повышению качества и уменьшению времени расчёта при работе в связке с вероятностными методами, такими как ГФ и ФЧ. Во всех сценариях наблюдалось уменьшение времени расчёта ГФ, который работал в ограничениях, полученных от Н-локализации. Повышение точности было продемонстрировано только в сценарии *Camera* ввиду особенностей реализации. Также во всех сценариях наблюдалось увеличение точности ФЧ, работающего в ограничениях Н-локализации (*sdl_amcl_lite*) в сравнении с таким же фильтром (*amcl_lite*), повышение быстродействия, и в общем схожие показатели точности с фильтром, имеющим на порядок больше частиц (*amcl*).

Метод Н-локализации по ориентирам обладает универсальной вычислительной процедурой, что подтверждается работой с разными композициями входных данных, не меняя основную логику работы, а только вводя нужные функции интерпретации. Это позволяет использовать предлагаемый метод в разных сценариях, в том числе и комбинированных.

3.3. Технология глобальной локализации по визуальным ориентирам

3.3.1. Система распознавания объектов

Для проверки применимости предложенных алгоритмов необходимо провести эксперименты по распознаванию сцен и локализации по визуальным ориентирам. Для выполнения некоторых экспериментов требуется распознавать объекты. Обилие современных подходов, несмотря на открытость большинства библиотек, приводит к тому, что программные средства технического зрения каждый раз проектируются практически с нуля для новой задачи, ввиду её специфики. С другой стороны, в мобильной робототехнике можно определить ряд требуемых качеств системы распознавания объектов, которая позволяла бы проводить настройку для отдельных подзадач. Для этого в рамках диссертационной работы было разработано средство Extended Object Detection (EOD), которое реализует широкие возможности в области распознавания объектов [Moscowsky, 2022a].

Мульти-атрибутное распознавание объектов.

Базовым элементом системы является атрибут (признак) a . Идеологически каждый атрибут отражает какую-то характеристику объекта, которую требуется распознать (форму, образ, цвет и т.п.). Каждому атрибуту в системе соответствует некоторый программный модуль. А описание распознаваемого объекта o в таком случае задаётся списком атрибутов:

$$o = \langle c, \{a_i\} \rangle. \quad (3.6)$$

где c – имя (тип) объекта. Атрибуты в системе бывают следующих типов:

- детектирования (выполняет поиск брусков на изображении);
- проверки (определяет соответствие текущего объекта требуемым характеристикам);
- извлечения информации (определяет для текущего объекта интересующие характеристики).

Брус r является самым распространенным способом описания объектов на изображении. Поэтому атрибуты в первую очередь связаны с этой характеристикой, а также с изображением I , на котором идёт детектирование:

$$\begin{aligned} a^{detect}(I) &= \{ \langle r, m, s \rangle \}, \\ a^{check}(I, \langle r, m \rangle) &= s, \\ a^{extract}(I, \langle r, m \rangle) &= m', \end{aligned} \quad (3.7)$$

где s – численная оценка качества детектирования, m – некоторая дополнительная информация об объекте, которая может содержать любую информацию, включая различные представления данных об объекте (сегментационные маски, ключевые точки), семантические метки, трёхмерное положение и т. п. Однако брус r используется в качестве единицы, по которой происходит соединение данных отдельных атрибутов в единый объект. В качестве меры сравнения двух брусов r используется распространенная метрика IoU (Intersection over union) – индекс Жаккара, примененный к изображениям:

$$IoU(r_1, r_2) = \frac{r_1 \cap r_2}{r_1 \cup r_2}. \quad (3.8)$$

Поскольку каждый атрибут детектирования a^{detect} выдаёт набор брусов (3.7), то для распознавания объекта (3.6) нужен механизм агрегации данных брусов, в качестве которого может выступать «жадный» выбор максимального значения из всех возможных комбинаций брусов двух атрибутов, или же Венгерский алгоритм [Munkres, 1957]. Последовательно «соединяя» таким образом списки полученных брусов от разных атрибутов, можно найти области на изображении, которые удовлетворяют каждому атрибуту. Однако ввиду сложностей окружающей среды (шумы, освещение, проблема ракурса, перекрытие объектов и т. п.) ряд атрибутов может не распознать нужные области. Это приведёт к тому, что не будет распознан объект целиком. Для устранения этой проблемы предлагается такой алгоритм, который позволит «пропускать» такие атрибуты, понижая финальную числовую оценку распознанного объекта. На основе этого алгоритм распознавания объекта o (3.6) выглядит следующим образом:

Табл. 3.11. Псевдокод алгоритма мульти-атрибутного распознавания объектов

1.	Algorithm object_detection(I, o, τ_{iou}):
2.	-- I – изображение
3.	-- o – описание объекта (3.6)
4.	-- τ_{iou} – порог критерия IoU (3.8)
5.	$\{r_f, m_f, s_f\} = a^{o_0}(I)$
6.	for i in $[1, \text{len}(o)]$
7.	if a^{o_i} is detect
8.	$\{r_c, m_c, s_c\} = a^{o_i}(I)$
9.	$int_c, int_f, extra_c, extra_f = \text{match}(\{r_c, m_c, s_c\}, \{r_f, m_f,$
10.	$s_f\}, \tau_{iou})$
11.	$\{r_f, m_f, s_f\} = \emptyset$
12.	for $r_c, m_c, s_c, r_f, m_f, s_f$ in int_c, int_f
13.	$r = r_c \cap r_f$
14.	$m = m_c \cup m_f$
15.	$s = s_c + s_f$

16.	add $\langle r, m, s \rangle$ to $\{r_f, m_f, s_f\}$
17.	for r_c, m_c, s_c in $extra_c$
18.	add $\langle r_c, m_c, s_c \rangle$ to $\{r_f, m_f, s_f\}$
19.	for r_f, m_f, s_f in $extra_f$
20.	add $\langle r_f, m_f, s_f \rangle$ to $\{r_f, m_f, s_f\}$
21.	if a^{o_i} is <i>check</i>
22.	for r_f, m_f, s_f in $\{r_f, m_f, s_f\}$
23.	$s_f = s_f + a^{o_i}(I, r_f, m_f)$
24.	if a^{o_i} is <i>extract</i>
25.	for r_f, m_f, s_f in $\{r_f, m_f, s_f\}$
26.	$m_f = m_f \cup a^{o_i}(I, r_f, m_f)$
27.	for r_f, m_f, s_f in $\{r_f, m_f, s_f\}$
28.	$s_f = s_f / \text{len}(o)$
29.	return $\{r_f, m_f, s_f\}$

Алгоритм **object_detection** (Табл. 3.11) принимает на вход изображение I , описание объекта o , и порог τ_{iou} . Вспомогательная функция **match** на основе двух наборов брусков формирует два множества брусков, которые имеют IoU (3.8), не превосходящее заданного порога τ_{iou} и два множества остальных брусков. В зависимости от типов детекторов происходит слияние информации об объекте. На выходе алгоритм возвращает набор брусков, дополнительную информацию и численную оценку, которая рассчитывается как средняя оценка всех атрибутов, беря за 0 те, что не внесли вклад в конкретный объект. Вычислительная сложность алгоритма зависит от реализации функции **match**, сложность которой является $O(N^2)$ в случае жадного алгоритма и $O(N^3)$ в случае Венгерского, где N наибольшее количество сравниваемых брусков. Однако это число зависит исключительно от возможностей отдельных детекторов и входных данных. По числу атрибутов данных алгоритм обладает линейной вычислительной сложностью.

Иллюстративный пример работы алгоритма **object_detection** (Табл. 3.11) изображен на Рис. 3.9, где распознаётся объект, имеющий три атрибута:

- атрибут детектирования, определяющий чашки;
- атрибут детектирования эмблемы в виде мальчика в феске;
- атрибут проверки, что объект красного цвета.

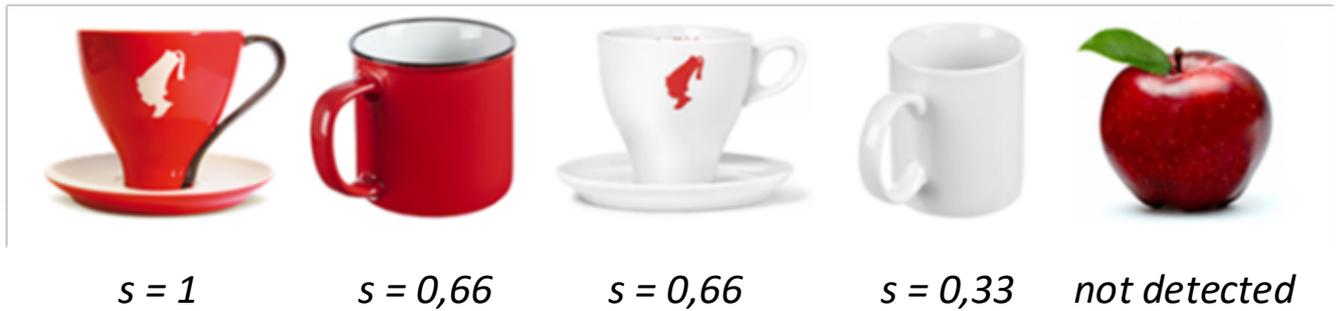


Рис. 3.9. Иллюстративный пример работы алгоритма распознавания объектов

В первом объекте на Рис. 3.9 совпали все атрибуты, поэтому его численная оценка равна 1, на втором и третьем только два из трёх, что уменьшает оценку до 0.66, на предпоследнем только один атрибут был найден, поэтому его численная оценка равна 0.33, последний объект не был распознан, поскольку для срабатывания атрибута проверки красного цвета не было подходящих брусков.

Распознавание сложных объектов.

Мульти-атрибутное описание объекта (3.6) несмотря на ряд удобств не даёт возможность описать объекты, имеющие широкий набор признаков. Чтобы расширить возможности системы, был предложен метод распознавания сложных объектов co , которые описываются группой «простых» объектов o (3.6) и связями (отношениями) между ними. Поиск композиций был также выполнен с помощью решения задачи поиска изоморфного подграфа [Московский, 2018]. Идеи, сформулированные и реализованные в рамках этого подхода, позже легли в основу метода распознавания сцен, изложенного в подразделе 2.1.3.

Для описания связей между объектами в составе сложного объекта вводятся отношения, которые могут быть пространственными («спереди», «над», «справа»), сравнительными («больше», «дальше», «длиннее»), принадлежности («быть частью») или иметь специфичную семантику («держат в руках», «ехать на»). Тогда сложный объект может быть описан как набор объектов и списка отношений:

$$co = \langle c, \{o_i\}, \{r_i\}, M \rangle, \quad (3.9)$$

где c – имя (тип) объекта, r_i – отношения, M – матрица связи объектов и отношений, если отношение r_n связывает объекты o_i и o_j , то $M_{ij} = n$. Отношения не обязаны быть заданными для каждой пары объектов. Каждому отношению соответствует программный детектор, способный определить его выполнимость на основе информации об объектах:

$$r(\langle r_i, m_i \rangle, \langle r_j, m_j \rangle) = \begin{cases} true \\ false \end{cases}. \quad (3.10)$$

Для распознавания сложных объектов, заданным таким способом, предлагается воспользоваться алгоритмами поиска изоморфного подграфа, поддерживающими разметку вершин и рёбер (например, семейством VF). На основании описания (3.9) строится граф сложного объекта G_{co} , где вершинам соответствуют объекты, а ребрам отношения. В качестве разметки этого графа каждой вершине в соответствие ставится имя (тип) объекта, и аналогично происходит для рёбер, которым ставится в соответствие тип отношения. Это частный вид разметки графов, называемой раскраской, когда вершинам и рёбрам сопоставляются натуральные числа (идентификаторы типов).

В процессе распознавания строится граф общего наблюдения G_{view} , на который наносятся объекты всех типов, представленные в графе сложного объекта, а также проверяются отношения (3.10), и в случае их выполнения удовлетворяющие этим отношениям вершины соединяются рёбрами. На графе G_{view} используется такая же раскраска, как и на графе G_{co} . После того как сформированы два графа, возможно применить алгоритмы поиска изоморфного подграфа, используя граф G_{view} как тот, в котором происходит поиск, и граф G_{co} как тот, который ищется. Найденные подграфы G_{view} будут изоморфны G_{co} и совпадать по раскраске, а следовательно, объекты будут образовывать заданный сложный объект (3.9). Однако, как и в случае с объектами с мульти-атрибутным описанием, в случае ошибок распознавания могут не распознаться отдельные объекты, это приведёт к полному нераспознаванию сложного объекта, поскольку нарушен изоморфизм. Решить эту проблему можно так же, как в методах распознавания сцен (подраздел 2.1.6): путем добавления фиктивных объектов. На граф общего наблюдения G_{view} добавляется по одному объекту каждого типа, что представлены в описании сложного объекта с меткой «фиктивный». Все эти объекты соединяются ребрами с объектами тех типов, что имеют отношения в описании сложного объекта. Таким образом, в результирующие ответы будут добавлены распознанные фиктивные объекты. Для того, чтобы учесть это при оценивании результатов, предложена следующая мера:

$$s_{co} = \frac{\sum_{r_k \sim (o_i, o_j)} (s_{o_i} + s_{o_j})}{2|\{r_i\}|}, \quad (3.11)$$

где s_{o_i} численная оценка объекта o_i , равная нулю, если объект имеет метку фиктивности. В случае нормирования оценок s_{o_i} на интервал $[0; 1]$, оценка (3.11) также нормирована на этот интервал. Алгоритм распознавания сложных объектов представлен в виде псевдокода:

Табл. 3.12. Псевдокод алгоритма распознавания сложных объектов

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31.	Algorithm complex_object_detection (I, CO, τ_{iou}) -- I - изображение -- co - описание сложного объекта (3.9) -- τ_{iou} - порог критерия IoU (3.8) $G_{co} = \mathbf{build_graph}(co)$ $V_{view}, E_{view} = \emptyset, \emptyset$ for o in $co.\{o\}$ $\{r_f, m_f, s_f\} = \mathbf{object_detection}(I, o, \tau_{iou})$ (Табл. 3.11) add $\{r_f, m_f, s_f\}$ as vertices to V_{view} with color $o.c$ for o_i in V_{view} for o_j in $V_{view}, i > j$ if $co.M[i][j] \neq 0$ if $co.r[co.M[i][j]](o_i, o_j)$ add edge(i, j) to E_s with color $co.r[co.M[i][j]]$ for o in $co.\{o\}$ if not fake in V_{view} with color $o.c$ add fake to V_{view} with color $o.c$ for all fakes in V_{view} for o_i in V_{view} if relation ($fake.c, o_i.c$) in M add edge(fake, i) to E_s with color of relation $G_{view} = \langle V_{view}, E_{view} \rangle$ $F = \mathbf{VF}(G_{view}, G_{co})$ $CO = \emptyset$ for f in F init co for (i, j) in f $co = co \cup V_{view}[i]$ $co.s = \mathbf{calc_s}(co)$ (3.11) add co to CO return CO
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Алгоритм **complex_object_detection** (Табл. 3.12) принимает на вход изображение I , описание сложного объекта co и порог критерия IoU . Вспомогательная функция **build_graph** строит граф сложного объекта. Функция **object_detection** реализует алгоритм мульти-атрибутного распознавания (Табл. 3.11). Функция **VF** реализует одноименный алгоритм поиска изоморфного подграфа с учетом раскраски графов. Функция **calc_s** рассчитывает численную оценку объекта согласно (4.6). Алгоритм возвращает набор брусков, дополнительную информацию и численные оценки. Вычислительная сложность алгоритма в худшем случае является экспоненциальной относительно числа распознанных объектов, поскольку опирается на решение задачи поиска изоморфного подграфа.

Предложенный алгоритм позволяет распознавать группы объектов благодаря введению отношений между ними, что расширяет возможности разработанной системы распознавания объектов.

В качестве иллюстрации работы системы можно привести несколько простых примеров. Объект «biker», состоящий из объектов «person» и «motorcycle» и отношения «mount», которое можно составить из простейших геометрических соотношений «быть над» и «пересекаться». Такой объект задаётся как:

$$CO_{biker} = \langle \{o_{person}, o_{motorcycle}\}, \{r_{mount}\}, \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix} \rangle. \quad (3.12)$$

Результат распознавания такого объекта представлен на Рис. 3.10.



Рис. 3.10. Пример распознавания сложного объекта «biker», приведён результат, где отсеяны сложные объекты с весом ниже 0.5

Каждый из «простых» объектов в примере с Рис. 3.10 состоял из двух атрибутов: нейросетевого детектора образа и атрибута проверки конкретной семантической метки.

Другой пример отражает объект «TwoPersonsOnMotorcycle», и расширяется из первого (3.12) следующим образом:

$CO_{two\ persons\dots}$ (3.13)

$$= \langle \{o_{person}, o_{person}, o_{motorcycle}\}, \{r_{mount}, r_{not\ same}\}, \begin{vmatrix} 0 & 2 & 1 \\ 2 & 0 & 1 \\ 1 & 1 & 0 \end{vmatrix} \rangle.$$

Тут задействуется отношение «not same», указывающее на разные объекты «person», это связано с особенностью поиска изоморфного графа, при которой одна вершина найденного подграфа может соответствовать нескольким вершинам исходного. Результат распознавания такого объекта представлен на Рис. 3.11.

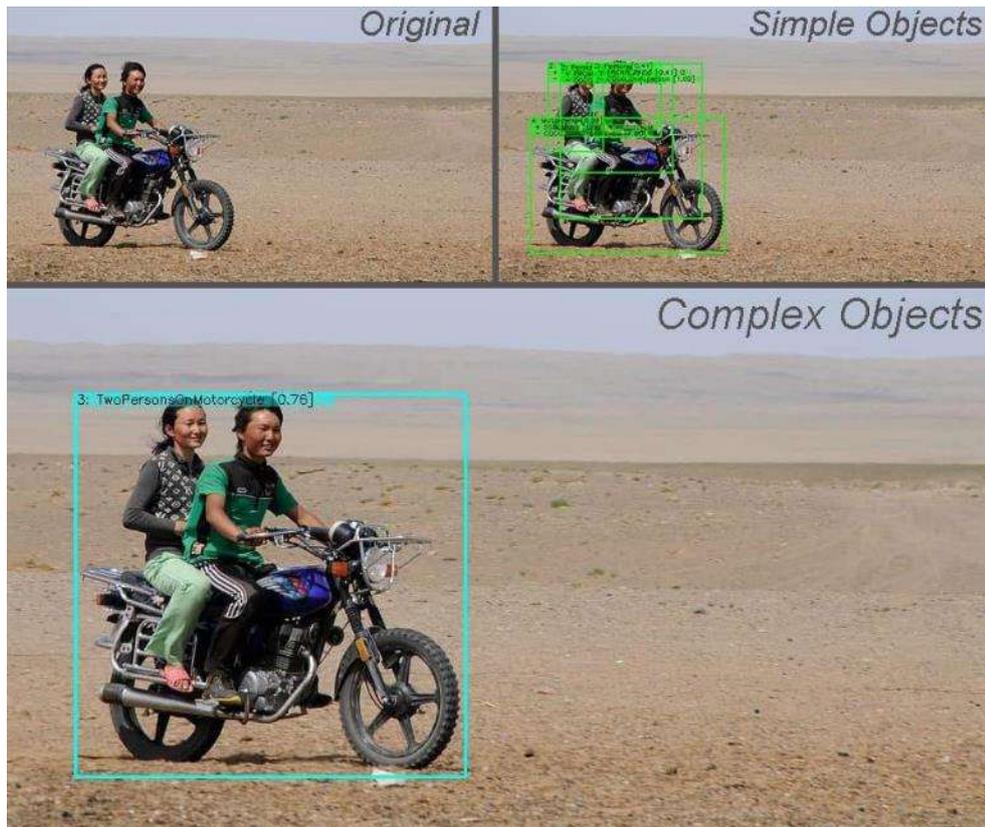


Рис. 3.11. Пример распознавания сложного объекта «TwoPersonsOnMotorcycle», приведён результат, где отсеяны сложные объекты с весом ниже 0.5

Вид графов сложных объектов из примеров (3.12) приведен (3.13) на Рис. 3.12.

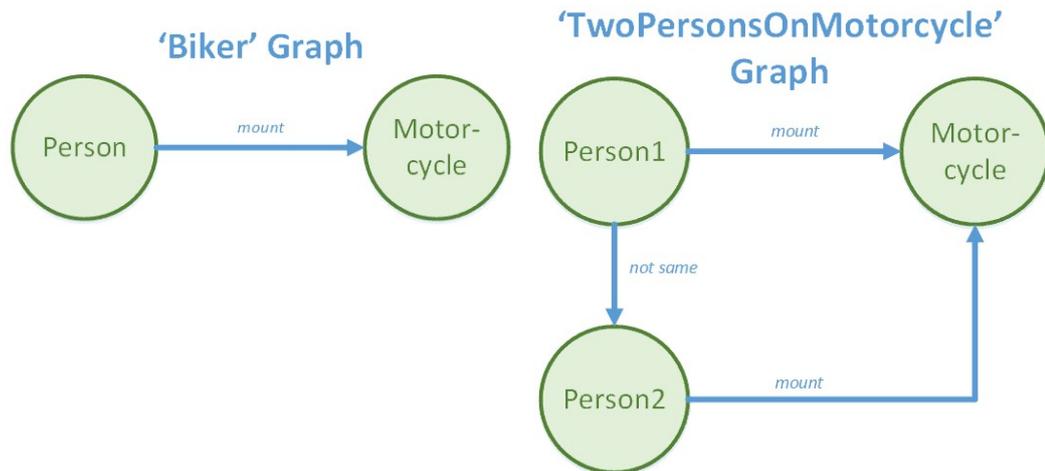


Рис. 3.12. Вид графов сложных объектов из примера (3.12) (слева) и (3.13) (справа)

Реализация и апробация.

Данное средство реализовано на языке C++ и, помимо основной библиотеки, имеет ряд реализаций, использующих её, включая узел для системы ROS. Программный код, связанный с этим, снабжен подробной документацией и выложен в открытый доступ⁶. Всё описание объектов, требуемое для системы распознавания, задаётся посредством конфигурационного XML-файла, это позволяет оперативно менять параметры без изменения исходного кода.

Средство EOD заняло первое место на конкурсе открытых пакетов⁷ для среды ROS, проходившем в рамках практической конференции «ROS Meetup» (декабрь 2020), а также первое место в конкурсе статей LearnOpenCV Blog Olympic 2021⁸ [Moscowsky, 2022b].

Средство EOD использовалось в экспериментах в среде Gazebo и на реальных роботах, описанных далее в работе.

3.3.2. Реализация технологии глобальной локализации

Т.к. предложенная технология допускает некоторые вариации в своём процессе, особенно в пункте выбора методов классической локализации по дальномерным данным, то реализация должна поддерживать модульность, которая может быть естественно реализована за счет системы ROS при условии сохранения интерфейсов между отдельными узлами.

⁶ <https://github.com/Extended-Object-Detection-ROS>

⁷ <https://habr.com/ru/articles/541876/>

⁸ <https://learnopencv.com/blog-olympics/>

ROS-обертка системы распознавания EOD имеет стандартизированной вход и свой тип сообщения о распознанных объектах. Поскольку связка методов распознавания сцен и N-локализации в предлагаемой технологии неизменна, то они реализованы в виде отдельного модуля, который принимает на вход данные от системы распознавания и публикует далее полученные ограничения. Далее полученные ограничения могут быть использованы каким-либо вариантом средства локализации. Общая схема представлена на Рис. 3.13.

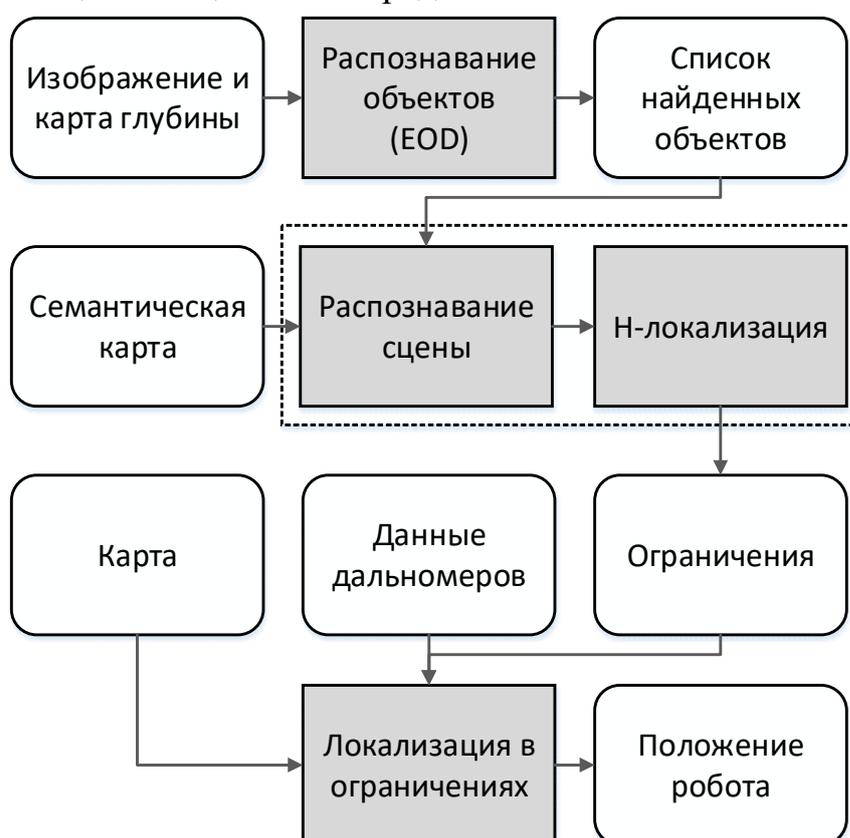


Рис. 3.13. Общая схема реализации комплексной технологии в рамках системы ROS

В блоке «Локализация в ограничениях» на Рис. 3.13 может быть ряд различных подходов, среди которых реализованы:

- Алгоритм локализации методом RANSAC+ICP по обрезанному в полученных ограничениях облаку точек (Табл. 2.13).
- Алгоритм локализации ФЧ в ограничениях (Табл. 2.16).
- Модификация стандартного для системы ROS узла AMCL⁹ по данным двумерного дальномера и плоской карте. Метод AMCL был модифицирован так¹⁰, что посредством ROS-сервиса принимает

⁹ <https://wiki.ros.org/amcl>

¹⁰ https://github.com/MoscowskyAnton/amcl_limited

ограничения в виде мульти-интервалов и при запросе на глобальную локализацию распределяет частицы только внутри данной области алгоритмом **restriction_to_particles** (Табл. 2.14), а не по всей карте.

3.3.3. Тестирование технологии глобальной локализации в среде Gazebo

Первичное тестирование работы комплексной технологии глобальной локализации проводилась в модели холла лаборатории робототехники НИЦ «Курчатовский институт» в симуляторе Gazebo (Рис. 3.14).



Рис. 3.14. Модель помещения и робота в среде Gazebo

Робот RItRover, оснащенный стерео камерой и 2D-лидаром, использует метод AMCL, основанный на ФЧ для локализации в помещении. Проблема глобальной локализации этим методом заключается в том, что помещение относится к категории симметричных, и представляет собой квадрат с 4-мя колоннами, расположенными по центру. В таком виде метод AMCL для каждого положения робота имеет минимум 3 положения, в которых скан «накладывается» на карту так же хорошо, как и в истинном положении. Когда робот выполняет глобальную локализацию, равномерно распределив частицы по помещению, то он будет локализован в одном из таких мест практически с равной долей вероятности. Таким образом, используя объекты, расположенные в помещении, можно выполнить задачу распознавания сцен, провести Н-локализацию и ограничить область, в которой инициализируются частицы. Схема варианта комплексной технологии с визуализацией процесса изображена на Рис. 3.15.

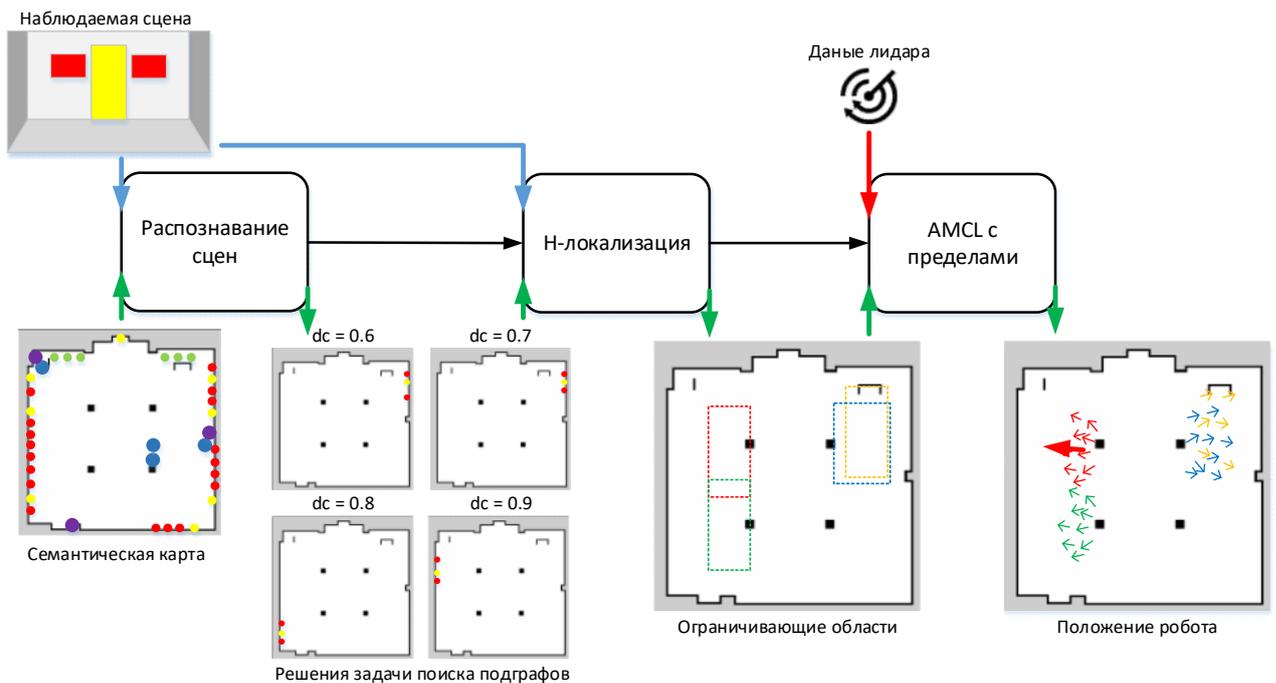


Рис. 3.15. Вариант комплексной технологии глобальной локализации с помощью AMCL

Распознавание объектов было выполнено с помощью описанной системы EOD (раздел 3.3.1), которая выполняла запуск свёрточной нейросети YOLOv7 [Wang, Bochkovskiy, Liao, 2023], доученной на объектах из симуляции, и определяла расстояния до объектов с помощью карты глубины со стереокамеры. Использование такого подхода в среде симуляции позволило смоделировать ряд проблем, связанных с ошибками распознавания и ошибками определения положения объектов.

Первичная двумерная карта помещения была построена с помощью средства `gmapping`¹¹ и отредактирована вручную. Семантический слой был реализован с помощью разработанного средства картирования объектов [Московский, 2020], выполняющего нанесение распознанных объектов на карту и дальнейшую кластеризацию иерархическими методами. Данное средство работает при готовой карте и локализации методом AMCL с произведенной вручную глобальной локализацией. Семантическая карта содержала 5 объектов разных типов: дверь, плакат, стол, шкаф, растение. Двумерный и семантический слой карты представлены в схеме на Рис. 3.15.

В данном эксперименте был применен метод распознавания сцен на основе поиска изоморфного подграфа с кластеризацией отношений и без использования визуального сходства и мягкого расширения (алгоритм `sig_cluster`, Табл. 2.6).

¹¹ <http://wiki.ros.org/gmapping>

Метод Н-локализации использовал базовый набор функций интерпретации (2.41)-(2.45) и дополнительную функцию проверки корректности (2.51) и реализовывался алгоритмом `sdm_global_localization` (Табл. 2.10). Реализация метода AMCL с пределами описана в разделе 3.3.2.

В ходе эксперимента робот помещался в случайное доступное место на карте, сначала проводилась локализация при помощи AMCL с равномерным распределением частиц по всему помещению, затем предлагаемым подходом (Рис. 3.15). Было проведено 300 подобных случайных перемещений и рассчитана ошибка определения положения (Рис. 3.16).

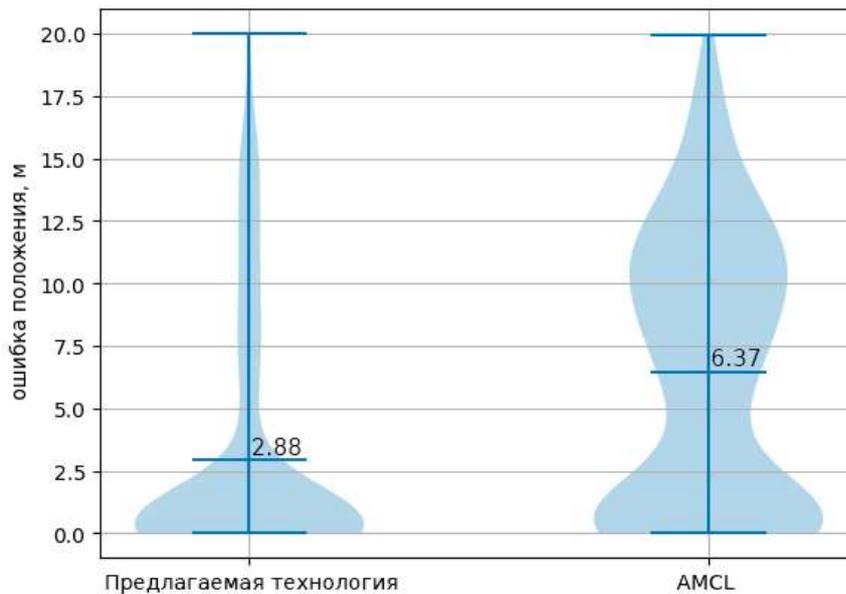


Рис. 3.16. Ошибка глобальной локализации в поставленном эксперименте

По форме графиков плотности распределения на Рис. 3.16 видно, что при использовании предлагаемой технологии положения более сконцентрированы в окрестности истинного значения, нежели при работе AMCL без модификаций. Полученную ошибку потенциально также можно уменьшить за счёт использования учёта визуального сходства согласно подразделу 2.1.7, однако в симуляции не было возможности использовать данные улучшения, поскольку визуально большинство объектов одного класса были идентичны, т.к. представляли копии друг друга.

3.4. Создание набора данных для тестирования

Как уже упоминалось, в настоящее время отсутствуют открытые наборы данных для тестирования задач распознавания сцен, которые обладали бы

перечисленными характеристиками (подраздел 1.2.3). Также требуются фотореалистичные объекты, охват больших площадей, наличие семантической карты, набора сцен и, что особенно важно, сопоставления между ними – верного решения \bar{f} , о котором упоминалось в разделе 2.4, для расчёта введённых метрик. Современные исследователи в области задач распознавания сцен прибегают к модификации наборов данных для смежных задач в области локализации и технического зрения, например, путём ручной разметки, или создают свои наборы с нуля. К сожалению, подобные наборы данных также не выкладывались в открытый доступ. Таким образом, встаёт задача подготовки набора данных для тестирования, удовлетворяющего перечисленным критериям.

В качестве основы для данного набора данных был использован набор KITTI-360 [Liao, Xie, Geiger, 2023], который удовлетворяет всем критериям (Табл. 1.1), кроме сопоставления между сценами и семантической картой. Данный набор собран при движении автомобиля, оснащенного рядом сенсоров, включая несколько камер и лидаров, а также системами глобального позиционирования с помощью точного GPS. Автомобиль движется по городской среде (в т.ч. частному сектору), иногда съезжая в пригородную зону. В наборе имеются 9 различных по продолжительности последовательностей, также отличающихся характером движения и типом местности. Далее по тексту отсылка к той или иной последовательности набора данных будет осуществляться путем указания двузначного номера этой последовательности, согласно оригинальному набору данных KITTI-360, например, «последовательность 03».

В наборе KITTI-360 имеется трёхмерная глобальная разметка объектов в виде параллелепипедов, которая фактически является готовой семантической картой (2.3), необходимой для задачи распознавания сцен. Единственное, что требуется – это определить визуальный образ данных объектов. Также в этом наборе данных существует двумерная разметка объектов для задачи семантической сегментации изображений, что можно перевести в описание сцены (2.2). При этом для некоторых типов объектов имеется разметка экземплярной сегментации, т.е. разделение на отдельные объекты. Там, где произведена экземплярная сегментация, также выполнена глобальная привязка к идентификатору объекта, т.е. он сохраняется от кадра к кадру для одного и того же объекта. Однако эта глобальная идентификация никак не связана с упомянутой ранее трёхмерной глобальной разметкой объектов, что не позволяет использовать этот набор для задачи распознавания сцен в первоизданном виде. Только наличие связи в идентификаторах объектов между картой и сценой позволит оценить точность предлагаемых методов распознавания сцен. Для решения данной

проблемы и установки этого соответствия была реализована процедура сопоставления (Рис. 3.17).

Предлагаемая процедура была выполнена для ограниченного набора объектов. Т.к. предлагаемые методы работают со статическими и компактными в пространстве объектами, то были исключены все объекты, помеченные как динамические, а также протяженные объекты классов «дорога», «тротуар», «ограждения», «забор» и другие. Также был удалён класс «растительность», т.к. помимо подходящих отдельно стоящих деревьев, содержит и протяженный кустарник.

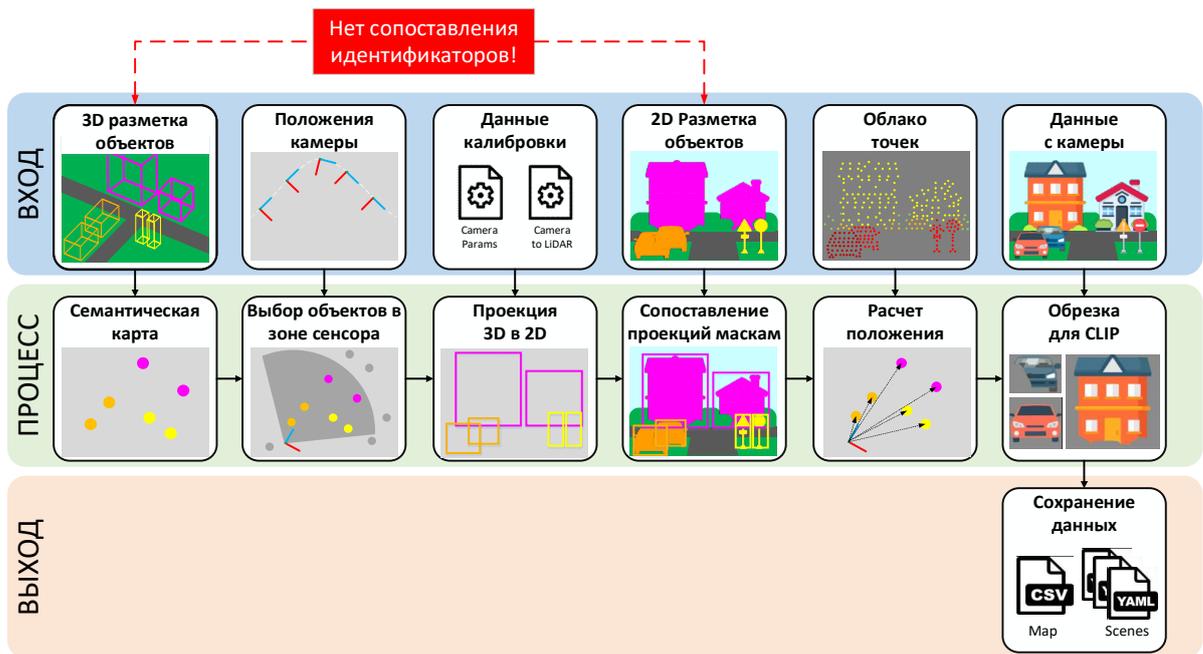


Рис. 3.17. Схема подготовки набора данных KITTI-360 для задачи распознавания сцен

На первом шаге трёхмерная разметка объектов переводится в семантическую карту посредством расчёта центров этих объектов. Далее для каждого положения камеры определяются видимые объекты на основании выбранной дальности и угла раствора используемой камеры. Параллелепипеды этих объектов с трёхмерной разметки далее проецируются в плоскость изображения и описываются прямоугольниками со сторонами, параллельными границам изображения (брусами). Для объектов, для которых есть экземплярная сегментация, происходит сопоставление с полученными прямоугольниками при помощи классического Венгерского алгоритма на основе критерия совпадения площадей этого прямоугольника и маски объекта. Для объектов без экземплярной сегментации общая маска класса объектов обрезается по полученному прямоугольнику. Полученные в результате двумерные объекты имеют общие с

семантической картой идентификаторы, что позволяет численно оценить качество сопоставления в задаче распознавания сцен.

На основе полученной маски определяется расстояние до объектов по данным лидара. В данном подходе определяются точки лидара, попавшие в маску объекта, фильтруются точки с низкой интенсивностью. Далее несколькими методами рассчитывается расстояние до объекта: по ближайшей точке, среднее, медианное среднее, первый квартиль, а также взвешенное среднее с учетом интенсивности точек. На основе посчитанного сопоставления был проведен анализ ошибок для каждого из предложенных методов, на Рис. 3.18 приведены результаты для некоторых объектов.

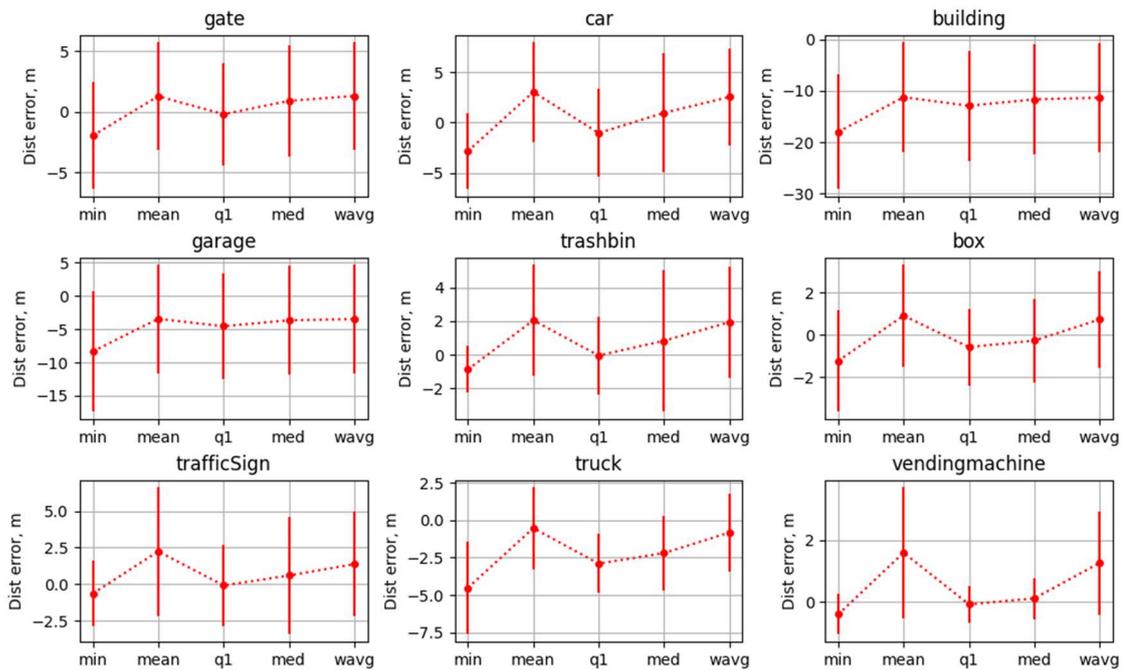


Рис. 3.18. Значения среднего и среднеквадратичного отклонения для каждого способа расчёта ошибки определения расстояния для некоторых типов объектов из последовательности 00

Медианное расстояние (med) на Рис. 3.18 показывает меньшую ошибку, однако у первого квартиля (q1) меньше среднеквадратичное отклонение, поэтому далее в работе во всех экспериментах использовался данный метод расчёта расстояния до объектов (как было описано в разделе 2.1.2). Помимо расстояния до объектов, определяются углы на него, используя центр обрамляющего прямоугольника, который описывает полученную маску объекта. Для углов также происходит расчёт ошибок, однако ввиду симметричности средняя ошибка выбирается равной нулю.

Также на основе обрамляющих прямоугольников и маски подготавливается изображение для входа на графический кодировщик CLIP¹². Все пиксели обрамляющего прямоугольника, не попавшие в маску, закрашиваются нейтральным серым цветом, т.к. случайно попавшие в брус другие объекты могут дать свой отрицательный вклад в результирующий вектор. Для семантической карты вектор объекта рассчитывается как нормированное среднее среди всех векторов этого объекта со всех сцен (2.24).

Объекты, попавшие в сцену, но имеющие малую площадь маски или малое число точек лидара, удаляются как ненадёжные. На финальном шаге полученная семантическая карта очищается от объектов, которые не были замечены ни на одной из сцен. Полученные таким образом данные, а также скрипт, их генерирующий, доступны онлайн¹³. Описанный алгоритм сопоставления (Табл. 3.13, Рис. 3.19) семантических данных был выполнен для каждой из девяти последовательностей.

Табл. 3.13. Псевдокод алгоритма восстановления соответствия сцен и карты для набора KITTI-360

1.	Algorithm scene_recognition_kitti360 (3Dbboxes, CamCalib, Poses, RGBImages, SegImages, LidarFrames, params):
2.	-- 3Dbboxes - 3D разметка объектов
3.	-- CamCalib - параметры камеры
4.	-- Poses - положение кадров (GPS)
5.	-- RGBImages - изображение с камер
6.	-- SegImages - сегментированные изображения
7.	-- LidarFrames - данные с лидара
8.	-- params - параметры извлечения
9.	init Mtemp
10.	init {S}
11.	for 3Dbbox in 3Dbboxes then
12.	o_m = (3Dbbox.GID, 3Dbbox.p, 3Dbbox.c, 3Dbbox.3Dbox)
13.	add o_m to Mtemp
14.	camHFoV = getHFoV (CamCalib)
15.	for pose in Poses then
16.	M_local = get_local_objects (Mtemp, pose, camHFoV, params.Dsens)
17.	for o_m in M_local then
18.	o_m.2Dbox = project3Dbox (pose, CamCalib, o_m.3Dbox)
19.	Stemp = extract_objects_from_seg (SegImages[pose.id], params.min_area_px)
20.	S = hungarian_match (M_local, Stemp, params.IoU)
21.	for o_s in S then

¹² <https://huggingface.co/openai/clip-vit-base-patch32>

¹³ https://github.com/MoscowskyAnton/scene_recognition_kitti_360

22.	<code>o_s.dist_params = get_dist(LidarFrames[pose.id], o_s, params.min_lidar_points)</code>
23.	<code>o_s.clip_feature = CLIP(o_s)</code>
24.	<code>add S to {S}</code>
25.	<code>M = filter_map(Mtemp, {S})</code>
26.	<code>return M, S</code>

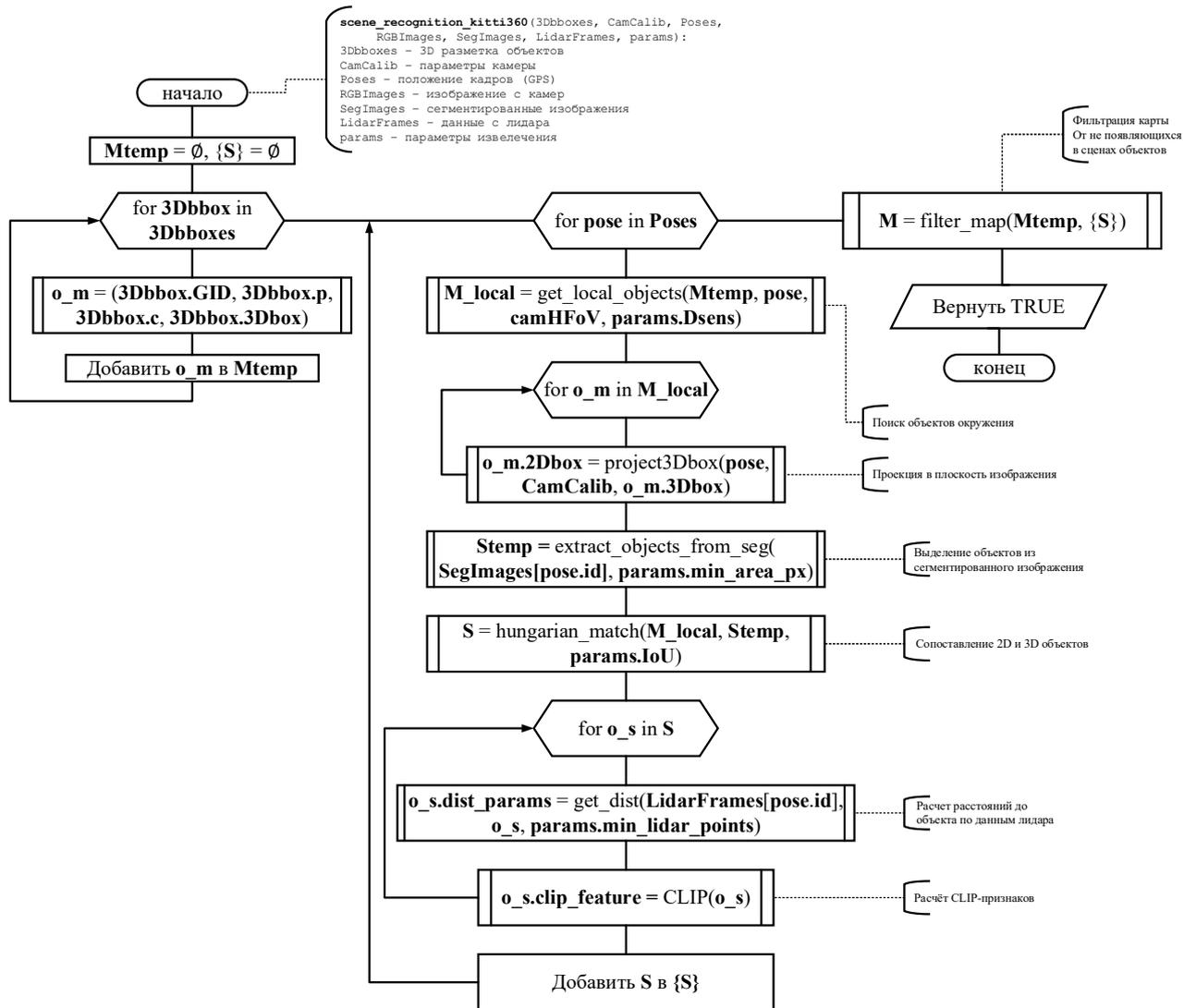


Рис. 3.19. Схема алгоритма восстановления соответствия сцен и карты для набора KITTI-360

Алгоритм `scene_recognition_kitti360` (Табл. 3.13, Рис. 3.19) принимает на вход элементы набора данных KITTI-360 и параметры извлечения. Функция `getHFOV` рассчитывает угол обзора камеры на основе данных о калибровке, функция `get_local_objects` выбирает объекты с карты на основе дальности сенсора и угла раствора, функция `project3Dbox` проецирует параллелепипед в брус (обрамляющий прямоугольник), функция `extract_objects_from_seg`

выделяет маски, идентификаторы и обрамляющие прямоугольники из сегментированных изображений, функция **hungarian_match** выполняет сопоставление спроецированных объектов, выделенных сегментацией, функция **get_dist** рассчитывает расстояние до объектов на основе данных лидара, функция **CLIP** рассчитывает признаки объекта на основе его маски, функция **filter_map** оставляет только те объекты, что имеются в сценах, и рассчитывает усреднённые признаки для объектов карты. Вычислительную сложность можно оценить, как $O(NM)$, где N – число объектов на семантической трехмерной разметке, а M – число положений камеры в наборе данных. На Рис. 3.20 приведен пример обработанного кадра, а на Рис. 3.21 вид сверху на этот кадр.



Рис. 3.20. Пример кадра №1548 из последовательности 00 набора данных KITTI-360, слева изображен вид с камеры с размеченными объектами (маска + брус), серым цветом отмечены невидимые с данного положения камеры объекты, но попадающие в область обзора

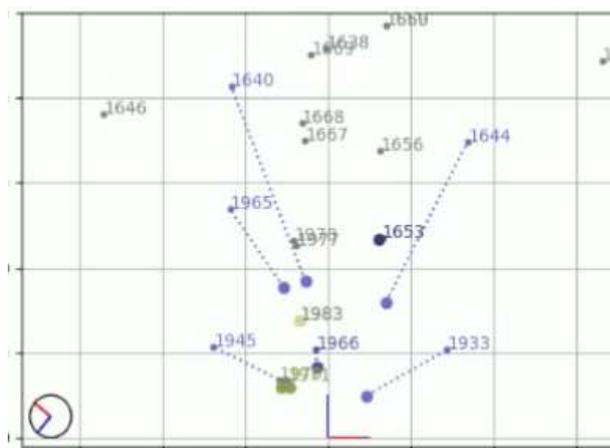


Рис. 3.21. «Вид сверху» на кадр №1548, изображенный на Рис. 3.20, где размечены как объекты карты (звездочками), так и сцены (кружками), а также их соответствие (пунктирной линией). Серым цветом отмечены невидимые с данного положения камеры объекты, но попадающие в область обзора

Также была произведена процедура деления на подкарты, согласно описанию в разделе 3.1.2, результаты такого разбиения приведены в Табл. 3.14.

Табл. 3.14. Численные характеристики полученного набора данных

№ последовательности	Объектов на карте	Число сцен	Число подкарт	Прирост общего числа объектов на подкартах
00	2333	9974	8	+22%
02	1720	8304	5	+7%
03	133	453	-	-
05	1576	6892	6	+7%
05	830	4798	3	+8%
06	1619	7552	6	+14%
07	476	1267	-	-
09	1978	10330	5	+22%
10	1298	2174	3	+0%

Для последовательностей 03 и 07 разбиение на подкарты не производилось ввиду небольшого числа объектов на них.

Таким образом был подготовлен набор данных, позволяющий проводить исследования с методами распознавания сцен, содержащий верный ответ, на основании которого возможно оценить качество полученных решений предложенными метриками (раздел 2.4).

3.5. Выводы к главе

В главе представлены реализации предложенных ранее алгоритмов распознавания сцен, N-локализации, методов локализации по дальномерным данным в ограничениях, которые объединены в общую комплексную технологию, включающую в себя также и распознавание объектов, предложенной системой EOD. Технология локализации реализована в рамках системы ROS, что позволяет поддерживать модульность. Были описаны начальные эксперименты в имитационных средах для отдельных частей технологии, показавшие как работоспособность алгоритмов так и результаты превосходящие аналоги. Работа технологии в целом была протестирована в системе физического моделирования Gazebo, где показала свою базовую работоспособность. Для проверки на реальных данных был подготовлен набор на основе KITTI-360, обладающий

всеми данными для проведения экспериментов и дальнейшей оценке полученных решений.

Глава 4. Экспериментальные исследования

Глава 4 посвящена экспериментальным исследованиям с изложенными в предыдущих главах методами и сравнению с их ближайшими аналогами. Методы распознавания сцен и метод N-локализации будут рассмотрены как по отдельности, так и в составе предложенной комплексной технологии глобальной локализации. Эксперименты проводились как в системах физического симулирования, так и на открытых наборах данных и реальных роботах.

4.1. Исследование метода распознавания сцен на реальном наборе данных

Как уже упоминалось, во многих работах, посвященных визуальной локализации с использованием задач распознавания сцен, не оценивается отдельно качество решений данной задачи. Зачастую оценкам подвергается общее решение, в которое вносят вклад как задача распознавания сцен, так и методы локализации робота. В данном разделе будет рассмотрена отдельно именно задача распознавания сцен, включая как предложенные в диссертации алгоритмы, так и взятые из литературы. Такое рассмотрение важно провести, поскольку это позволит выявить сильные и слабые стороны каждого из методов более явно. Результаты распознавания сцен вкупе с результатами методов локализации будут рассмотрены в разделе 4.3.

Одной из целей исследования на реальных визуальных данных являлась оценка влияния определения визуального сходства. Исследовался ряд методов как с учётом визуального сходства так и без:

- Метод на основе поиска изоморфного подграфа с кластеризацией отношений *sig* (Табл. 2.6).
- Расширение метода на основе поиска изоморфного подграфа с кластеризацией отношений (Табл. 2.6) с использованием CLIP-архитектуры *sig_clip*.
- Вариация метода на основе поиска изоморфного подграфа с прямым сравнением отношений (Табл. 2.3) и использованием CLIP *sig_lite_clip*.
- Мягкое расширение метода на основе поиска изоморфного подграфа с прямым сравнением отношений *sig_lite_soft1_clip* (Табл. 2.8) и с одним фиктивным объектом (Табл. 2.8).
- Метод на основе поиска максимальной клики (*mc*), реализованный по описанию из [Ankenbauer и др., 2023].

- Расширение метода на основе поиска максимальной клики ($mc+$) с добавлением меры сравнения отношений (3.2).
- Дальнейшее расширение метода на основе поиска максимальной клики с использованием оценки визуального сходства при помощи CLIP ($mc+_{clip}$, Табл. 3.1).
- Метод без использования поиска геометрических особенностей *CLIP-Loc*, выполненный по описанию из [Matsuzaki и др., 2024b] ($clip_loc$). Для данного метода был введен такой критерий оценки решений, как среднее сходство полученных соответствий.

Эксперименты проводились на наборе данных описанном в разделе 3.4, который имеет верное сопоставление \bar{f} (раздел 2.4), поэтому можно использовать следующие описанные метрики для оценки точности решения задачи распознавания сцен: *Precision* (2.79) лучшего ответа, *Score* (2.84) и *Degree* (2.85) всего множества ответов. Поскольку явных выбросов в полученном наборе данных не моделировалось, то метрики *Accuracy* и *Recall* не использовались, т.к. тождественны *Precision*.

Также замерялось время работы методов. В поставленных экспериментах время работы было ограничено сверху 120-ю секундами. Ограничение по времени было выбрано из тех соображений, что задача глобальной локализации производится достаточно редко (обычно в начале работы и в проблемные моменты), поэтому можно выделить намного больше времени, чем обычно требуется от непрерывных систем локализации, вынужденных работать с субсекундным временем. Также данное ограничение было выбрано ввиду большого количества данных, требующих обработки. Если время работы метода превышало данный временной порог, то его *Precision* и *Score* равнялись нулю, а *Degree* принимала значение *NoResult*.

Поскольку методы имеют ряд настраиваемых параметров, влияющих как на точность, так и на время исполнения, то проводилась оптимизация параметров на отобранном наборе из 100 сцен средством *optuna*¹⁴. Отобранный набор из последовательности 02 содержал сцены размером от двух до двенадцати объектов в равной пропорции, т.е. по десять каждого вида. Оптимизация длилась 100 шагов и максимизировала *Precision* лучшего ответа и одновременно минимизировала время работы алгоритма. Форма полученных графиков процедуры оптимизации (Рис. 4.1) говорит о том, что выбранные решения (самая правая точка, максимальная по *Precision*) находятся в области оптимума.

¹⁴ <https://optuna.org/>

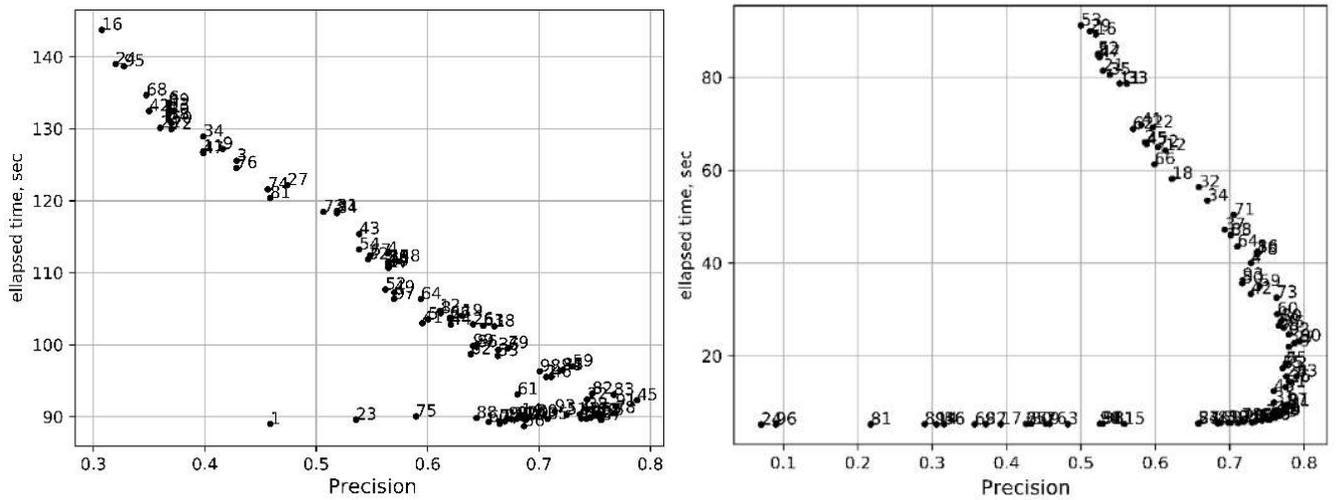


Рис. 4.1. Процесс оптимизации параметров методов *mc+_clip* (слева) и *sig_clip* (справа) на последовательности 02

Полученные параметры методов использовались во всех дальнейших экспериментах и приведены в Табл. 4.1.

Табл. 4.1. Параметры исследуемых методов и их показатели

Метод	Параметры	$ Precision $	$ t , c$
<i>sig</i>	$\tau_c=0.12, \tau_m=0.36, m_\sigma=1.29$	0.57	3.37
<i>sig_clip</i>	$\tau_c=0.05, \tau_m=0.39, m_\sigma=1.81$	0.79	9.02
<i>sig_lite_clip</i>	$\tau=0.79, m_\sigma=4.03$	0.79	23.21
<i>mc</i>	$d=7.63$ м	0.38	85.01
<i>mc+</i>	$\tau=0.24, m_\sigma=1.07$	0.35	75.59
<i>mc+_clip</i>	$\tau=0.88, m_\sigma=4.53$	0.79	92.31

В Табл. 4.1 вводится параметр m_σ , который позволяет увеличить чувствительность введённой формулы сравнения расстояний (2.9), рассчитывая не $p(r_1, r_2, \sigma r)$, а $p(r_1, r_2, m_\sigma \sigma r)$. Для метода *sig_lite_soft1_clip* не производилась оптимизация, его параметры были взяты идентичными *sig_lite_clip*, только с числом фиктивных объектов $k=1$. Для метода *clip_loc* также не проводилось оптимизации, поскольку единственный его параметр, отвечающий за число ответов (взятый равным 15), не влияет на *Precision* лучшего ответа и время исполнения.

В поставленных экспериментах объекты «строение» и «гараж» были исключены, т.к. используемый метод локализации объектов даёт слишком высокую погрешность для данных классов (Рис. 3.18). Важно отметить, что были оставлены все типы припаркованных транспортных средств («автомобиль»,

«грузовик», «велосипед» и т.п.) с целью увеличения среднего числа объектов на сценах. Конечно, в реальном использовании невозможно опираться на эти типы объектов при составлении семантических карт на долгосрочный период, однако в данном исследовании в первую очередь ставилась цель проверки работоспособности предложенных методов на реальных визуальных данных. В реальном использовании увеличить число объектов в сцене можно за счет таких, как «окно», «дверь», «канализационный люк», «дерево» и т.п.

Расчёты производились на компьютере с процессором AMD Ryzen7 2700X Eight-Core 3.70GHz для всех исследуемых методов и для каждой последовательности из набора данных KITTI-360 в отдельности.

Размер семантической карты (по числу объектов) для каждой последовательности свой, поэтому было проведено исследование зависимости *Precision* (Рис. 4.2) и *Score* (Рис. 4.3) от этого числа.

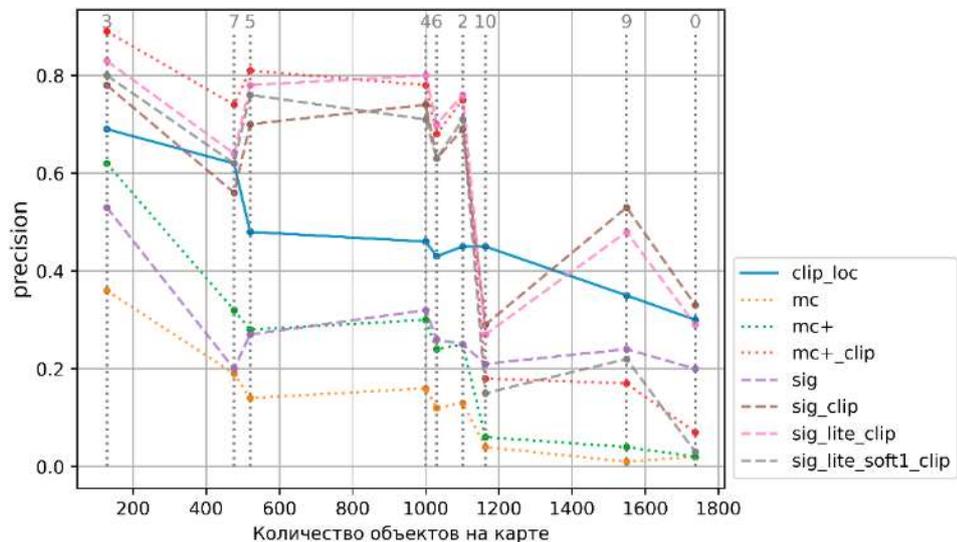


Рис. 4.2. Значения среднего *Precision* лучшего ответа от размера карты для каждой последовательности, цифрами сверху отмечен номер последовательности

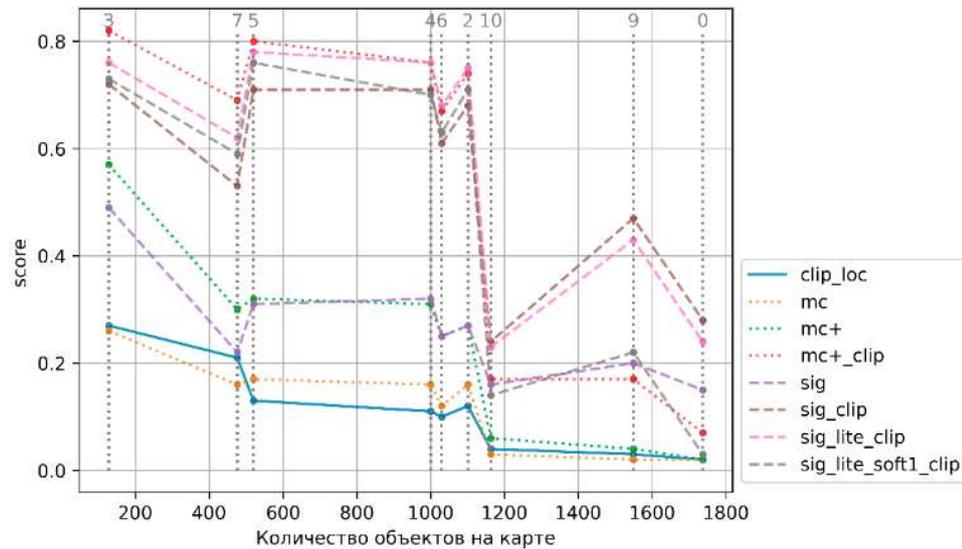


Рис. 4.3. Значения среднего *Score* набора ответов от размера карты для каждой последовательности, цифрами сверху отмечен номер последовательности

За исключением некоторых случаев на Рис. 4.2 и Рис. 4.3, видно убывание обоих критериев точности при увеличении числа объектов. Очевидно, что чем больше карта, тем больше вероятность, что на ней будут похожие сцены, уменьшающие точность. Также размер карты влияет на время исполнения методов, что важно при введенных ограничениях. Временные показатели приведены на Рис. 4.4.

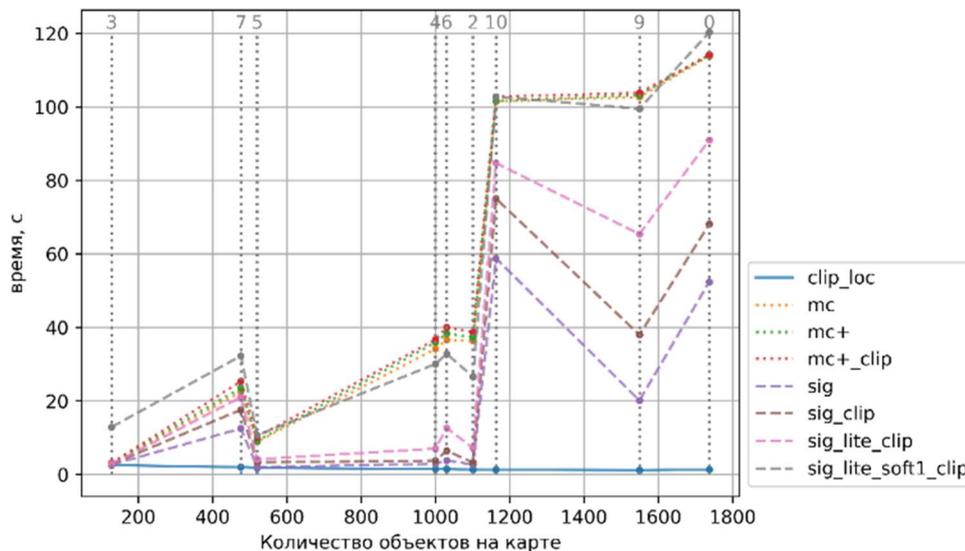


Рис. 4.4. Среднее время распознавания одной сцены в зависимости от размера карты для каждой последовательности, цифрами сверху отмечен номер последовательности

Общая тенденция на графиках точности (Рис. 4.2, Рис. 4.3) показывает, что методы с использованием визуального сходства всегда превышают свои же реализации, но без использования визуального сходства, причём значительно.

Методы семейства *mc* показывают лучшие результаты аналогичных им (с точки зрения использования\не использования визуального сходства) методов семейства *sig*, за исключением трёх самых больших карт. Это тоже может быть следствием того, что с выбранным ограничением по времени методы семейства *sig* работают быстрее (Рис. 4.4), т.е. методы *mc* чаще не успевают обработать данные. Данная скорость, конечно, может быть результатом конкретной реализации используемых методов теории графов. Методы семейства *mc* также примечательны тем, что каждая последовательная их модификация (*mc+* и *mc+_clip*) увеличивает точность по обоим параметрам, и при этом практически не увеличивая время.

Последовательные модификации метода *sig* вдобавок увеличивают точность за несколькими исключениями. Добавление визуального сходства с помощью CLIP в равной мере дало существенный скачок в точности. Вариант метода с прямым сравнением отношений (*sig_lite_clip*) увеличивает точность и уменьшает скорость. Мягкое расширение метода (*sig_lite_soft1_clip*) уменьшает точность, но за счёт ожидаемого увеличения времени исполнения. Тут стоит напомнить, что мягкое расширение разработано специально для условий выбросов (которые отсутствуют в исследуемом наборе данных), поэтому точность априори не может превзойти вариант метода без расширения, однако из-за увеличения числа объектов в графе карты увеличивается время. Эксперименты, показывающие применимость этого метода в данных с выбросами было показано в подразделе 0.

Отдельного обсуждения заслуживает метод *clip_loc*, поскольку он единственный из исследуемых, который использует только визуальное сходство и применяется несколько иначе, чем в оригинальной работе [Matsuzaki и др., 2024b], и к значительно большему числу объектов на карте. По критерию *Precision* (Рис. 4.2) этот метод показывает весьма приемлемые результаты, опережая все подходы только на геометрическом сравнении и иногда превосходя даже комбинированные. Тем не менее, по критерию *Score* (Рис. 4.3) метод показывает одни из самых низких показателей. Это говорит о том, что в целом метод в среднем верно распознаёт около половины объектов в сцене, однако верное распознавание сцены целиком для него достаточно редкая ситуация. При этом он показывает наилучшие показатели по времени, очень слабо зависящие от числа объектов на карте.

На графике среднего времени исполнения (Рис. 4.4) примечателен скачок для последовательности 07 при сравнимом с последовательностью 05 числом объектов для всех методов кроме метода *clip_loc*. Этот же скачок в виде резкого падения наблюдается и на графиках точности (Рис. 4.2, Рис. 4.3). Несмотря на близкое число объектов и схожую вытянутую топологию, карты этих последовательностей разнятся по наполнению. Последовательность 07 (загородная трасса) содержит много однотипных объектов типа *smallPole* – это «столбики» вдоль дорог, установленные с одинаковым интервалом, а т.к. они произведены заводским способом, то и выглядят одинаково. Поэтому методы, основывающиеся на геометрических особенностях и визуальном сходстве, испытывают проблемы с такими окружениями как по точности, так и по времени.

Результаты *Precision*, *Score* и времени распознавания одной сцены, усреднённые по всему набору данных KITTI-360, а также их среднеквадратичные ошибки приведены в Табл. 4.2.

Табл. 4.2. Численные результаты методов на полном наборе модифицированных данных KITTI-360, знаком σ отмечена среднеквадратичная ошибка метрик

Метод	$ Precision $, \uparrow	$\sigma Precision$	$ Score $, \uparrow	$\sigma Score$	$ time $, с, \downarrow	$\sigma time$
<i>clip_loc</i>	0,4	0,28	0,08	0,26	1,39	9,63
<i>mc</i>	0,08	0,25	0,09	0,22	66,57	48,58
<i>mc+</i>	0,16	0,35	0,17	0,33	67,39	48,52
<i>sig</i>	0,25	0,41	0,23	0,37	19,72	37,23
<i>sig_clip</i>	0,56	0,47	0,53	0,46	28,4	44,97
<i>mc+_clip</i>	0,46	0,48	0,45	0,48	68,28	48,42
<i>sig_lite_clip</i>	0,57	0,46	0,54	0,46	41,55	46,5
<i>sig_lite_soft1_clip</i>	0,43	0,48	0,43	0,47	65,3	51,09

Высокие значения ошибок ($\sigma Precision$, $\sigma Score$ и $\sigma time$) в Табл. 4.2 объясняются тем, что, помимо размера карты, размер сцены также влияет на скорость и качество распознавания. На графиках ниже показаны зависимости времени распознавания одной сцены (Рис. 4.5) и *Precision* (Рис. 4.6).

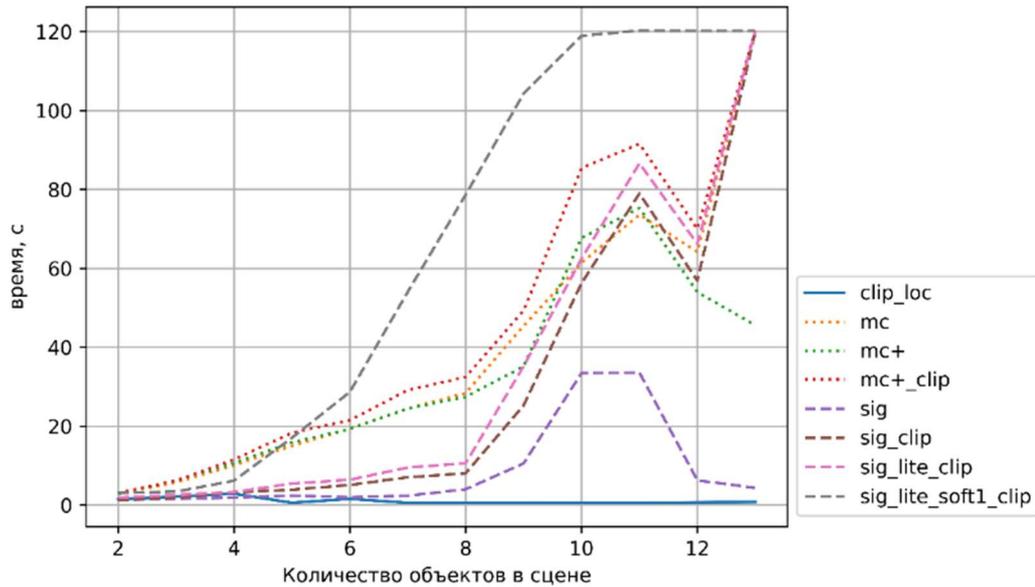
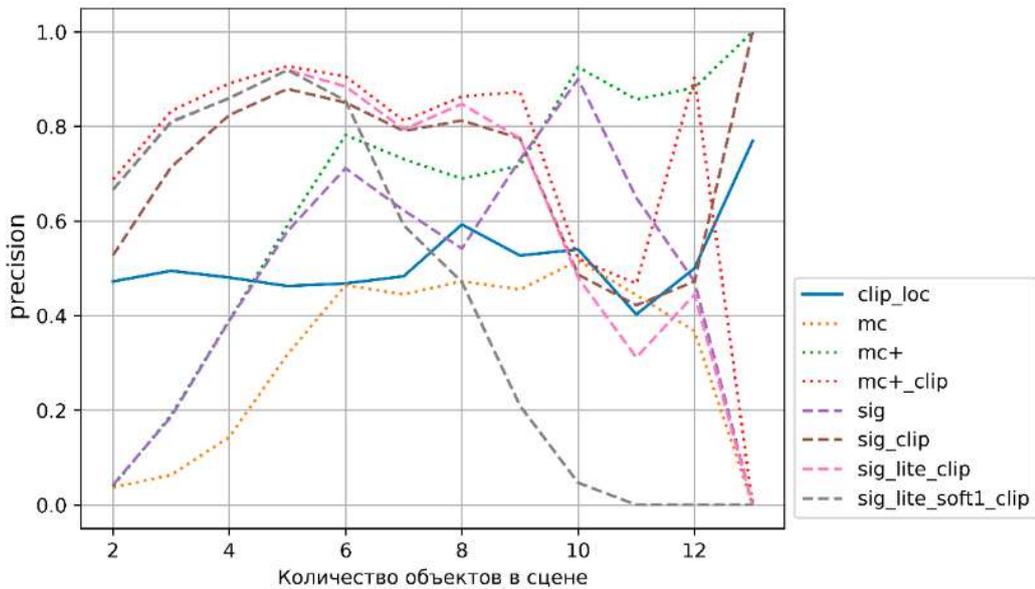


Рис. 4.5. Среднее время распознавания от размера сцены

Рис. 4.6. Средние показатели *Precision* от размера сцены

На Рис. 4.6 падение точности ряда методов в зависимости от числа объектов сцены также может быть следствием ограничения методов по времени. В то же время пониженная точность методов (особенно без использования визуального сходства) при небольшом числе объектов говорит о множественности похожих друг на друга решений. Небольшая разница в точности на промежутке 4-9 объектов для методов с использованием визуального сходства и геометрических особенностей говорит о том, что этого числа объектов может быть достаточно для решения задачи, и в некоторых случаях имеет смысл ограничить по числу объектов сцены.

На Рис. 4.7 указан критерий *Degree* для всего исследуемого набора данных.

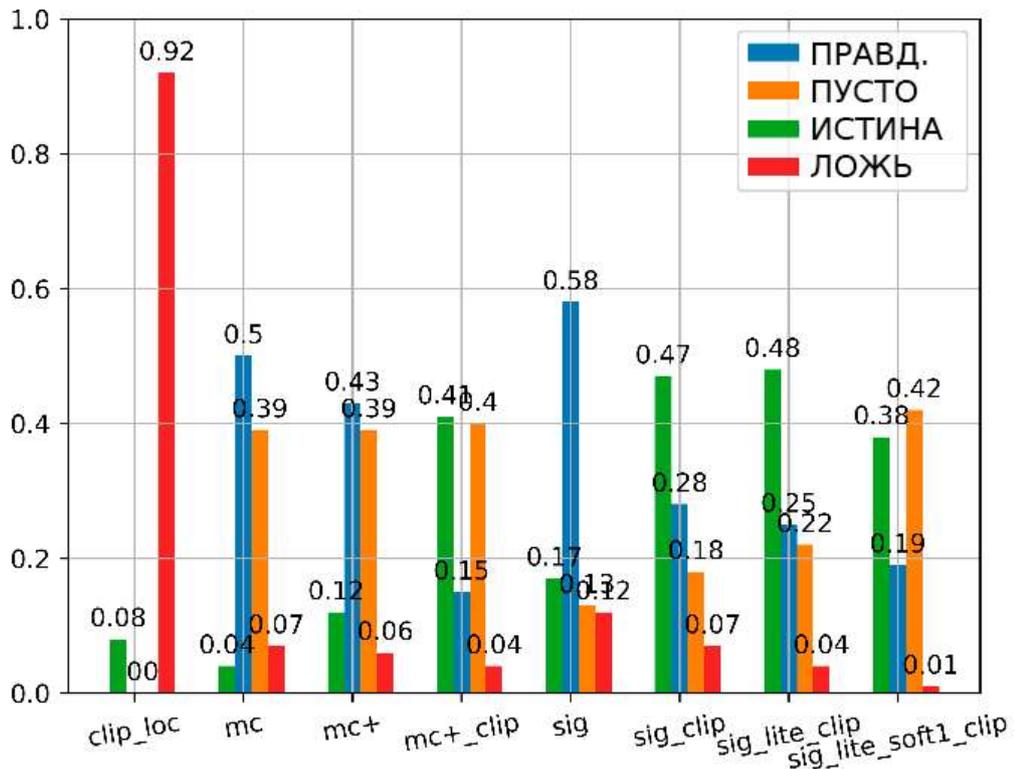


Рис. 4.7. *Degree* (2.85) методов, посчитанная на полном наборе данных

Degree методов, изображенная на Рис. 4.7, призвана показать возможность методов находить абсолютно верный ответ. От этого будет зависеть дальнейшая стратегия по использованию результатов распознавания сцен. Данный график также показывает, что соединение подходов на основе геометрических особенностей и визуального сходства приводит к наилучшим результатам. Однако можно отметить, что идеальное совпадение лучшего и верного ответа по метрике *Degree* (2.85) (*ИСТИНА*) не достигает даже 50% случаев для лучших методов. Немаленький процент наличия верного ответа среди всех полученных (*ПРАВДОПОДОБИЕ*), а также имеющийся процент неверных решений (*ЛОЖЬ*) подтверждает ранее высказанное предположение, что требуется внедрение дополнительных методов, способных оценить качество полученных решений. Такие методы должны обладать возможностью оценивать положение робота на основе нескольких гипотез и определять ошибочные объекты в ответах.

Ввиду неутешительных временных характеристик для самых больших карт (например, для последовательности 00 на Рис. 4.4) было проведено исследование влияния процедуры разбиения карты на подкарты, согласно описанию в разделе 3.1.2. Карта последовательности 00 была разбита на 11 подкарт с пересечениями, которые привели к общему повышению числа объектов на 29%. Тестировались методы, показавшие лучшие результаты: *mc+_clip* и *sig_lite_clip*. Несмотря на возможность параллельной обработки подкарт, для чистоты эксперимента

расчёты проводились последовательно. Значения *Precision* и времени исполнения приведены на Рис. 4.8.

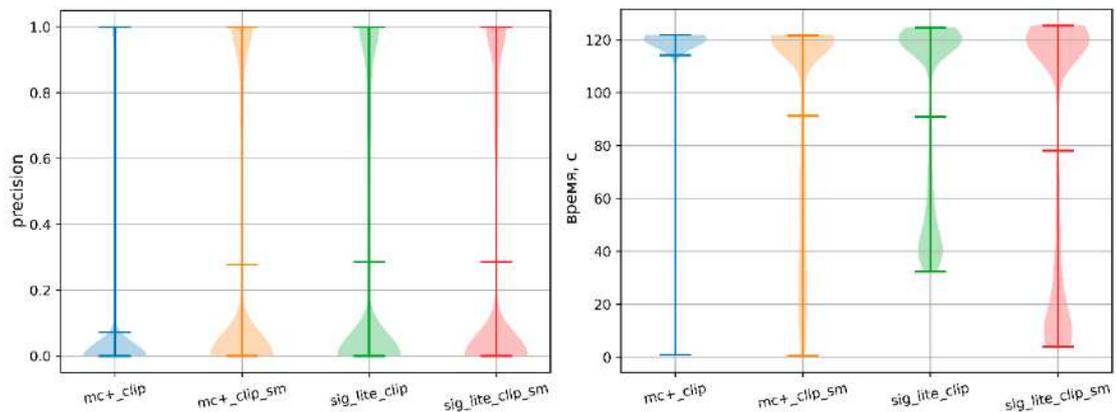


Рис. 4.8. Сравнение *Precision* и времени методов работы с полной картой и подкартами для последовательности 00, суффикс *_sm* в названии методов указывает на работу с подкартами

У метода *mc+_clip_sm* наблюдается значительный прирост по точности (+285%), выходящий на уровень *sig_lite_clip*, а также уменьшение затраченного времени на 20%. У метода *sig_lite_clip_sm* особого изменения в точности не наблюдается, однако также фиксируется ускорение (-14%). Тут стоит отметить, что в случае, когда любой из методов на подкартах не успевал закончить обработку за отведённое время, фиксировался полученный на этот момент ответ, который мог содержать как реальный ответ, так и ошибочный, но близкий к требуемому. Поэтому на графике *Degree* (Рис. 15) также видно увеличение неверных решений (*ЛОЖЬ*) у методов семейства *mc*.

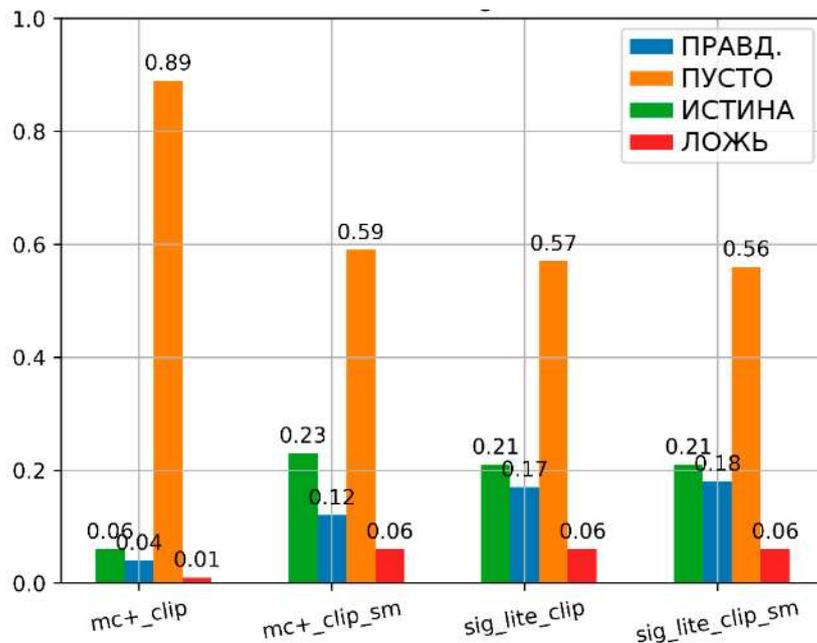


Рис. 4.9. Сравнение *Degree* для методов распознавания сцен для работы с полной картой и разбиением на подкарты

Полученные результаты говорят о том, что используемая реализация поиска максимальной клики в большей степени подвержена экспоненциальному росту времени вычисления, нежели методы на основе поиска изоморфного подграфа.

По результатам исследования методов распознавания сцен на полученном наборе данных, можно сделать следующие выводы:

1. Одновременное использование подходов на основе геометрических особенностей и визуального сходства дают наилучший результат.
2. Увеличение точности достигается путем допустимого увеличения времени выполнения методов, а также требует особого формата семантической карты и предварительного расчёта погрешностей.
3. Метод *sig_lite_clip* показал лучшую точность, однако его «мягкое» расширение *sig_lite_clip1_soft*, адаптированное под сложные ситуации, показывает сравнимые результаты с методом *mc+_clip*, который исходно адаптирован к этим ситуациям. Поэтому выбор «лучшего» метода зависит от условий конкретного применения.
4. Поскольку существует прямая зависимость времени выполнения от размера семантической карты, то была предложена процедура разбиения на подкарты, ускоряющая процессы распознавания сцен.
5. Показано падение производительности и точности в зависимости от размера сцены. Возможно, процедура обрезки сцены до некоторого числа объектов позволит ускорить процесс без потери точности.

6. Помимо зависимости от размеров сцены и карты, на исследуемые методы также влияет наполнение их объектами. Множество одинаковых и периодически расположенных объектов также является трудностью ввиду отсутствия геометрических особенностей и визуальных различий.
7. Полученной точности недостаточно для использования результата распознавания сцен без дальнейшей обработки и отсеивания неверных результатов.

Проведённые эксперименты показывают, что предложенные алгоритмы значительно повышают точность решения задачи распознавания сцен за счёт применения фундаментальных визуально-языковых моделей. Полностью оригинальный алгоритм на основе поиска изоморфного подграфа (**sig**, Табл. 2.3) показывает в целом лучшие показатели по точности, чем предложенное расширение известного алгоритма на основе поиска максимальной клики (**mc+_clip**, Табл. 3.1). Однако для работы с данными, имеющими выбросы, требуется расширение **sig_soft** (Табл. 2.8), у которого скорость падает с увеличением числа «фиктивных объектов», в то время как **mc+_clip** изначально адаптирован к сценарию с выбросами. Таким образом, оба предложенных алгоритма могут быть использованы на практике, и выбор зависит от условий применения. Однако точность определения верного ответа свидетельствует о том, что необходимо применять дополнительные методы верификации. Примером такого метода является процедура N-локализации, которая будет рассмотрена в последующих разделах.

4.2. Исследование N-локализации

4.2.1. Тестирование в среде Gazebo

Следующим шагом стала апробация разработанных алгоритмов в распространенной среде симуляции для роботов Gazebo. Был исследован метод N-локализации для модели группового робота YARP8 [Воробьев, 2023]. YARP8 (Yet Another Robot Platform) функционирует в среде, в которой расположены ArUco-маркеры, положение которых известно (Рис. 4.10).

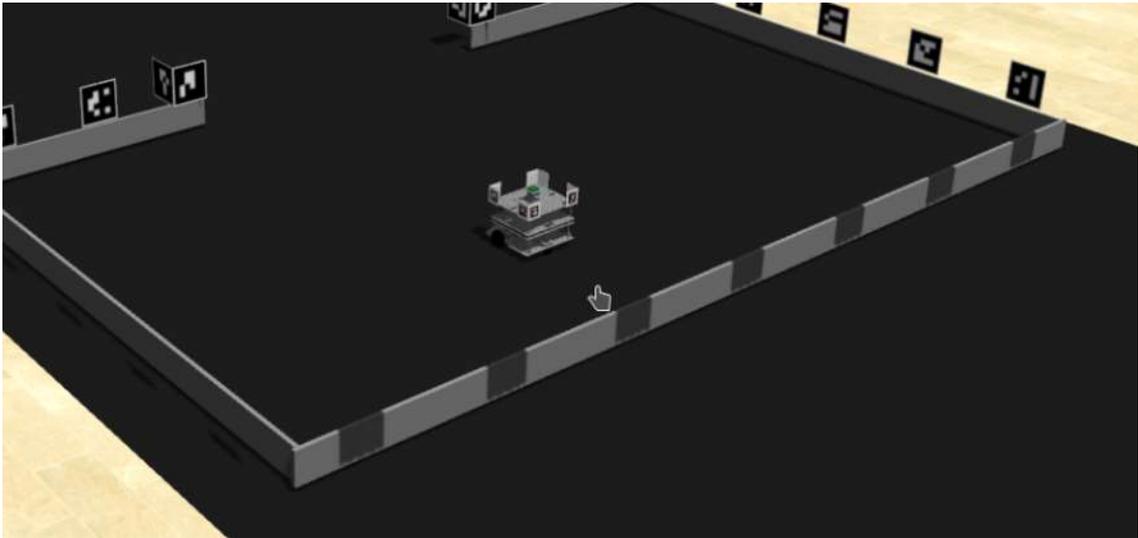


Рис. 4.10. Робот YARP8 с среде Gazebo

Робот оснащён моно-камерой и с помощью средства EOD (раздел 3.3.1) распознаёт маркеры и определяет расстояние до них и угол, производя локализацию аналогично сценарию *Camera* из подраздела 3.2.2. Использование среды Gazebo позволило провести исследование, более приближенное к реальности, чем предыдущее, за счёт следующих факторов: приближенные к реальности шумы в распознавании и локализации ориентиров, временная задержка за счёт затрат времени на распознавание, приближенная к реальности одометрия. Для тестирования были выбраны следующие методы:

- *amcl_100*, метод на основе ФЧ, с числом частиц от 100 до 200;
- *amcl_500*, аналогично *amcl_100*, но с числом частиц от 500 до 1000;
- *amcl_sdl_100*, Н-локализация (**sdm_continuous_localization**, Табл. 2.11) с ФЧ (с инициализацией **restrictions_to_particles**, Табл. 2.14) с числом частиц от 100 до 200;
- *amcl_sdl_500*, аналогично *amcl_sdl_100*, но с числом частиц от 500 до 1000;
- *amcl_sdl_isis_100*, Н-локализация (**sdm_continuous_localization**, Табл. 2.11) с последующим применением алгоритма исчерпывающего оценивания (**sdm_isis**, Табл. 2.12) с ФЧ с инициализацией **cartesian_to_particles** (Табл. 2.15) и числом частиц от 100 до 200.

Методы *amcl_100* и *amcl_500* были реализованы аналогично исследованию в разделе 3.2.2. Методы с использованием Н-локализации в дополнение к данным о расстоянии и угле использовали и данные об ориентации маркеров. Таким образом, к ставшему стандартным для локализации по камере набору функций интерпретации (2.41)-(2.45) добавилась функция (2.48).

Для тестирования была запущена программа, реализующая процедуру фуражировки (сбор ресурсов) роботом [Захарова, Московский, 2023]. Робот двигался из стартовой зоны («гнездо») в зону сбора ресурсов («кормовой участок») и выполнял поиск и захват элементов «пищи», доставлял этот элемент в гнездо, выгружал и начинал заново. Такая последовательность была трижды повторена и был записан *rosbag*-файл, позволяющий проигрывать данный объезд повторно. В файле записывались данные одометрии и данные от системы распознавания EOD, а также истинное положение робота, получаемое от симулятора, для проверки рассчитанных положений. Все исследуемые методы запускались на данных из полученного проезда и замерялись ошибки положения r и угла α .

Предварительно на том же *rosbag*-файле были измерены ошибки локализации объектов, включая расстояние до объекта r_l (0.128 ± 0.174 м), угол на объект α_l (0.005 ± 0.005 рад) и его ориентацию γ_l (0.248 ± 0.55 рад). Все эти данные использовались методами при локализации, результаты приведены в Табл. 4.3.

Табл. 4.3. Численные результаты исследуемых методов в среде Gazebo

Метод	$\bar{r} \pm \sigma_r$, м	$Q1-Q2-Q3(r)$, м	$ \bar{\alpha} \pm \sigma_{ \alpha }$, рад	$Q1-Q2-Q3(\alpha)$, рад
<i>amcl_100</i>	$0,74 \pm 0,86$	0,10-0,37-1,17	$0,49 \pm 0,80$	0,03-0,09-0,43
<i>amcl_500</i>	$0,49 \pm 0,64$	0,13-0,25-0,56	$0,24 \pm 0,53$	0,03-0,07-0,15
<i>amcl_sdl_100</i>	$0,25 \pm \mathbf{0,34}$	0,08-0,15-0,27	$0,09 \pm 0,17$	0,02-0,05-0,10
<i>amcl_sdl_500</i>	$0,20 \pm 0,34$	0,06-0,11-0,21	$0,08 \pm 0,12$	0,02-0,05-0,09
<i>amcl_sdl_isis_100</i>	$0,29 \pm 0,42$	0,08-0,17-0,34	$0,10 \pm 0,17$	0,03-0,06-0,11

Результаты также представлены в виде распределений на Рис. 4.11.

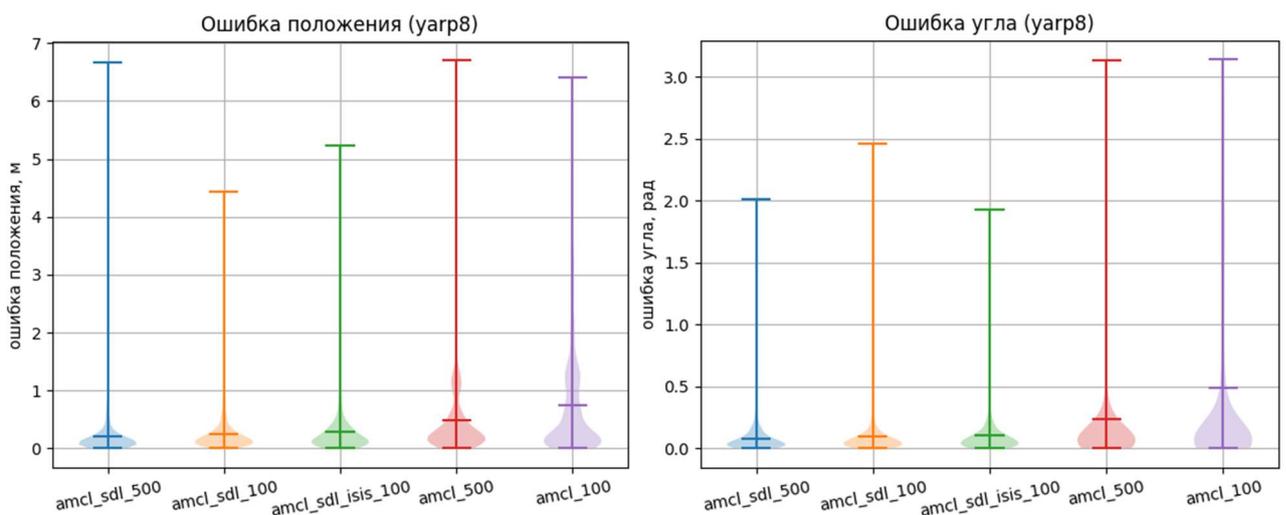


Рис. 4.11. Ошибки локализации робота YARP8 в среде Gazebo исследованными методами

Из полученных результатов, представленных в Табл. 4.3 и на Рис. 4.11, можно сделать вывод, что применение Н-локализации для задач непрерывной локализации по визуальным ориентирам уменьшает ошибки локализации практически в два раза, а иногда и более. Однако стоит отметить, что применение механизма исчерпывающего оценивания (алгоритм **sdm_isis**, Табл. 2.12) для задачи непрерывной локализации ухудшает точность. Это ухудшение можно объяснить рядом факторов, первым из которых является зависимость ошибки расстояния от самого расстояния до объекта (Рис. 4.12).

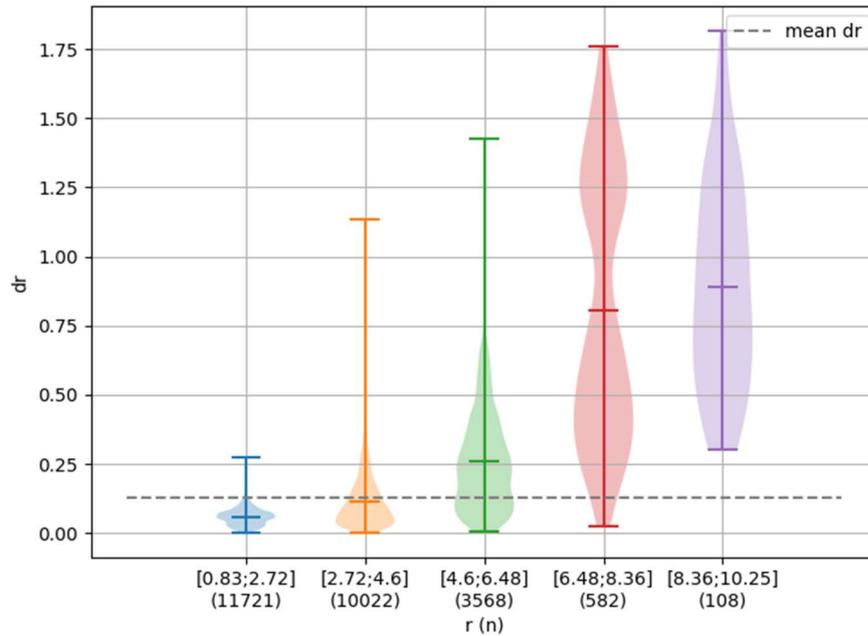


Рис. 4.12. Зависимость ошибки определения расстояния (dr) от расстояния (r) до объекта, серой пунктирной линией ($mean\ dr$) отображена средняя ошибка по всей выборке

Данная неравномерность приводит к тому, что введённые оценки погрешности в виде интервалов (2.37) не выполняются, из-за чего следует отсеивание лишних областей в ходе выполнения алгоритма **sdm_isis** (Табл. 2.12). В методах *amcl_sdl_100* и *amcl_sdl_500* данный фактор влияет меньше, ввиду большего размера стартовых областей. Другим фактором является то, что во время движения значительно перестраиваются полученные области, что приводит к сильному перераспределению частиц внутри областей, что, в свою очередь, не позволяет опираться на данные предыдущего шага в полной мере.

Тем не менее, данное исследование демонстрирует применимость метода Н-локализации к задаче непрерывной локализации по визуальным ориентирам, увеличивая точность вероятностного подхода, который пользуется полученными ограничениями.

4.2.2. Применение метода на реальных роботах

Предложенный метод N-локализации также был апробирован на ряде реальных роботов. Однако стоит отметить, что для этих роботов не было реализовано «дублирующих» систем локализации, которые могли бы быть взяты за истину, и позволили бы оценить качество определения положения. Поэтому качество определялось визуально и по косвенным признакам, включающим успешное выполнение роботами поставленных задач.

МПП (Макет подвижной платформы, Рис. 4.13). В рамках выполнения НИР «Разработка экспериментального образца аппаратно-программного комплекса сбора и обработки данных для контроля характеристик и калибровки целевой аппараты КА ДЗЗ в части создания макета подвижной платформы для проведения подспутниковых наземных радиометрических измерений» (шифр «Мониторинг-СГ-1.3.1.1.1-МПП») была применена ранняя версия метода N-локализации.



Рис. 4.13. Общий вид МПП с установленным манипулятором (слева), МПП без манипулятора в тестовом заезде рядом с калибровочным полигоном

В тестовых заездах, локализуясь по уникальным маркерам, расположенным в среде, робот успешно выполнял задачу подъезда в область калибровочного полигона (Рис. 4.13), в дальнейшем выполняя движение ориентируясь на этот полигон. Был получен акт о внедрении ПО (см. Приложение 1.).

Группа роботов YARP8 лаборатории робототехники НИЦ «Курчатовский институт». Реальный робот, модель которого применялось в исследовании, описанном в подразделе 4.2.1. Условия применения и сценарий идентичны условиям в среде моделирования Gazebo. Использовалась группа из 4х роботов YARP8. Каждый робот локализуется методом *amcl_sdl_100*, точность которого

(Табл. 4.3) была сочтена приемлемой. Перед роботами также стояла задача сбора «пищевых элементов» (Рис. 4.14), движение робота между «гнездом» и «кормовым участком» осуществлялось при помощи навигационного стека системы ROS¹⁵, полностью опирающегося на данные системы локализации.

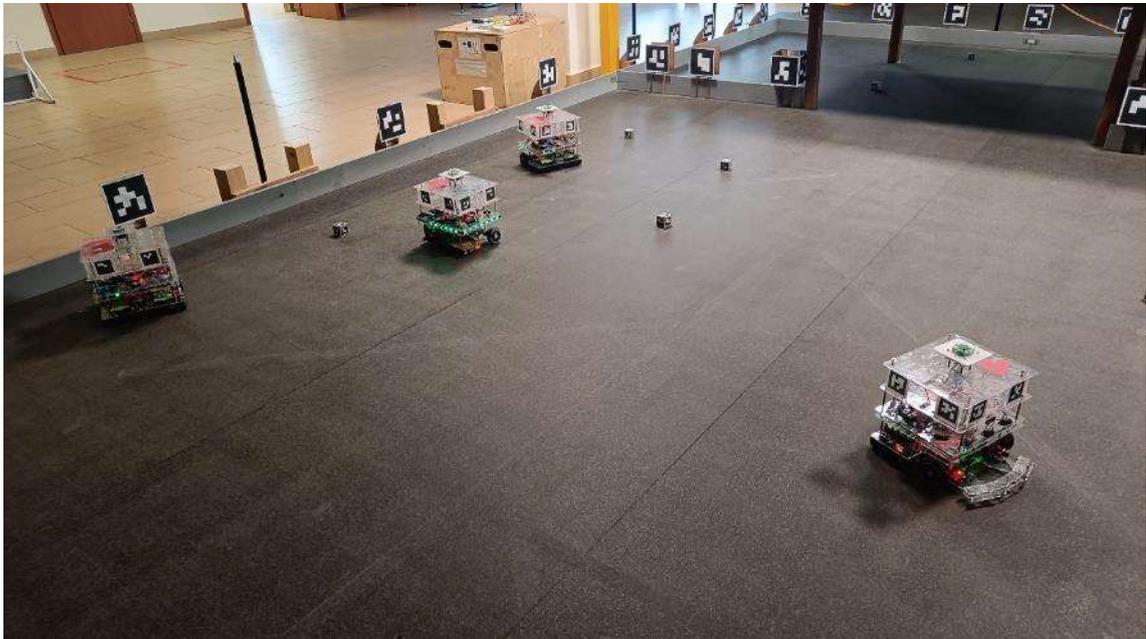


Рис. 4.14. Моделирование процессов фуражировки (сбора и транспортировки ресурсов) группой роботов YARP8, использующих метод N-локализации

Группа роботов, локализуясь предложенным методом, успешно собирает все пищевые объекты, расположенные на кормовом участке, и транспортирует их в гнездо.

Исследовательский робот YARP13, применяющийся в исследованиях по направлению БОС-2 (биологическая обратная связь второго рода) [Бова, Леушина, 2022; Бова, Леушина, 2024], проводящихся в лаборатории НИЦ «Курчатовский институт». Основная задача YARP13 – проведение экспериментов по оценке качества удалённого управления роботом в игровых задачах. Ввиду особенностей реализации данных задач роботу требуется знать своё местоположение. Поскольку робот является достаточно простым и малогабаритным и не несет на себе достаточно сенсоров и вычислительных ресурсов. Исходя из этого, было принято решение развернуть процедуру N-локализации с поддерживающим ФЧ по данным ArUco-маркеров, размещённых на потолке (Рис. 4.15).

¹⁵ <http://wiki.ros.org/navigation>

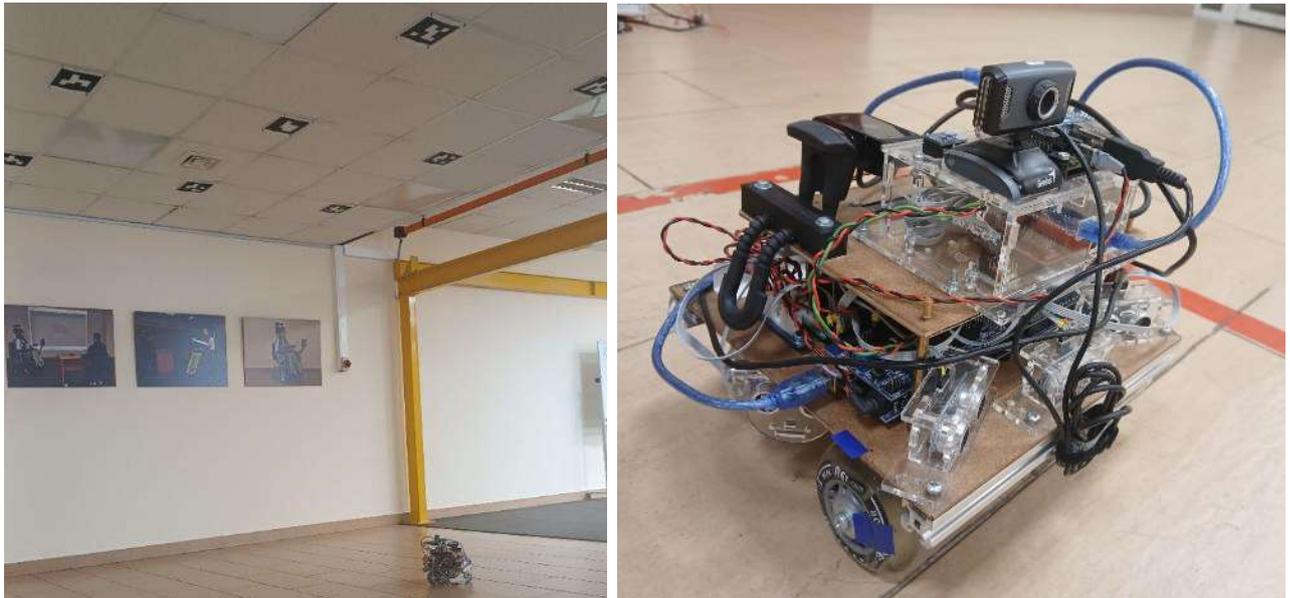


Рис. 4.15. Робот YARP13 в рабочем окружении

Расположение маркеров на потолке нисколько не изменило общую парадигму локализации, поскольку все измерения проходили в двумерной плоскости в координатах робота. Таким образом, фактически отработывался распространенный сценарий, где доступны такие данные, как расстояние до ориентира, угол на него и его ориентация. Угол обзора в данном случае становится равным 360° и дополнительные функции проверки корректности (2.50) не применялись, т.к. не имеют смысла при таком угле. Также в реальности была проверена гипотеза, объясняющая неприменимость алгоритма, исчерпывающего оценивания (**sdm_isis** Табл. 2.12) для задачи непрерывной локализации (раздел 4.2). Визуальная оценка результатов локализации показала, что предпочтительнее использовать вариант без алгоритма исчерпывающего оценивания (Рис. 4.16), поскольку сильное сужение интервалов в динамике влияет негативно, время от времени «запирая» робота в полученных ограничениях, до того, как алгоритм обнаружит противоречие и выполнит сброс полученных ограничений до стартовых значений.

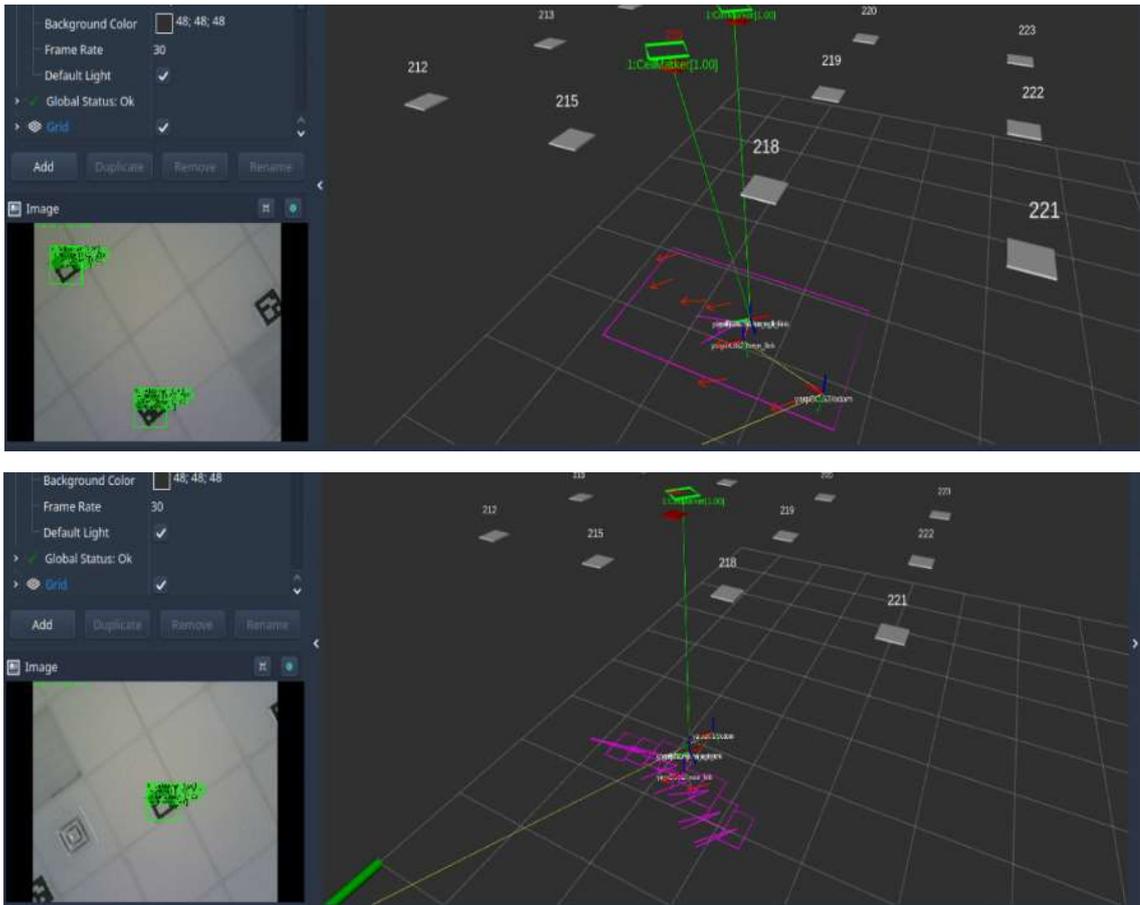


Рис. 4.16. Визуализация локализации робота YARP13 в средстве rviz без алгоритма исчерпывающего оценивания (сверху) и с алгоритмом исчерпывающего оценивания (снизу)

Робот-экскурсовод MENTOR-1 [Малышев, Московский, 2023]. Робот MENTOR-1 (Mobile Excursion Not Tele Operated Robot) используется в лаборатории робототехники НИЦ «Курчатовский институт» для проведения экскурсий, преимущественно для школьников старших классов. Задача робота – рассказывать посетителям о проектах лаборатории. Для этого робот перемещается между стендами, водя за собой группы. Для перемещения требуются системы навигации, опирающиеся на системы локализации. Будучи оснащённым двумерным лазерным сканирующим дальномером, робот использует метод AMCL¹⁶ из стека ROS1. Однако в ходе эксплуатации была обнаружена проблема, связанная с ограничением угла обзора дальномером посетителями и ложной релокализацией метода AMCL. Особенно сильно это проявлялось в удаленных от стен точках лаборатории (Рис. 4.17).

¹⁶ <http://wiki.ros.org/amcl>

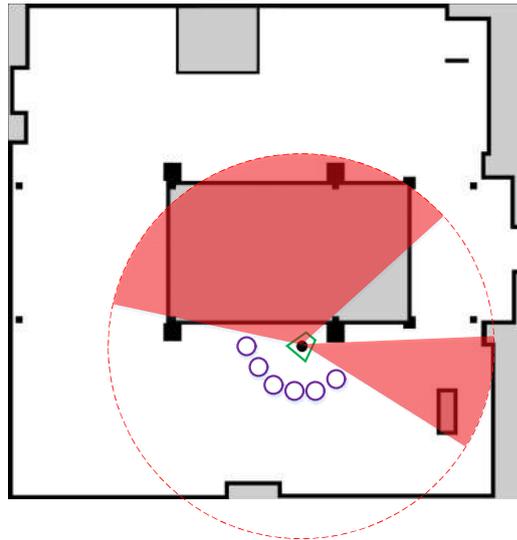


Рис. 4.17. Блокировка наиболее информативной части обзора. Зеленым отмечен робот, фиолетовым – люди, красным – зона работы лидара

Для решения данной проблемы было решено использовать метод N-локализации, который бы ограничивал работу метода AMCL в пространстве (Рис. 4.18).

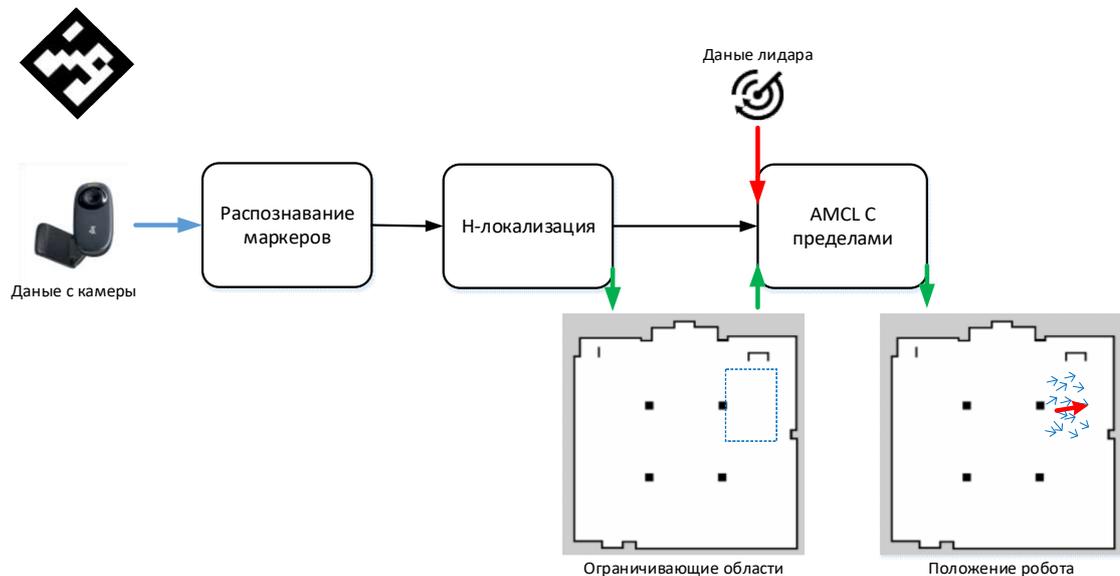


Рис. 4.18. Схема использования метода N-локализации для ограничения работы метода AMCL в проблемных местах

Подобно тому, как была реализована локализация робота YARP13 (описанная выше), на потолке были размещены ArUco-маркеры, и роботу была известна карта помещения с их расположением (Рис. 4.19).

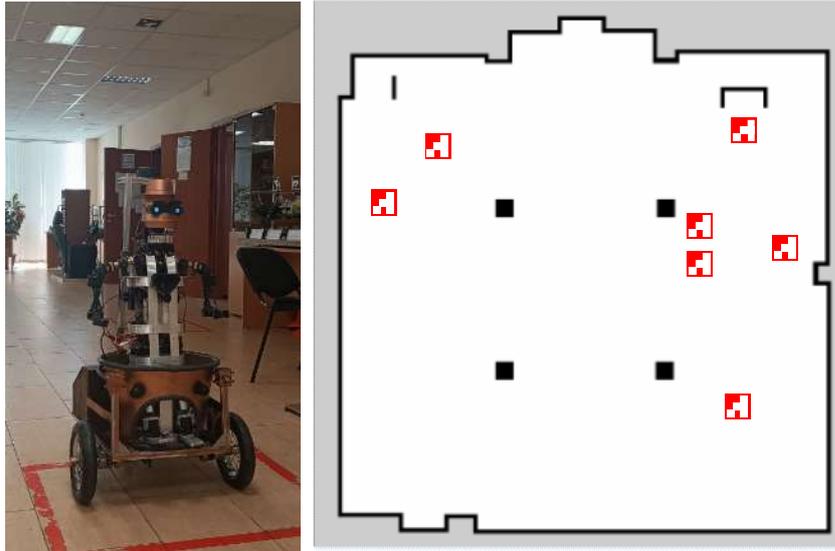


Рис. 4.19. Робот MENTOR-1 (слева), карта помещения с отмеченными положениями маркеров красным (справа)

Маркеры были размещены только в проблемных местах, у тех экспонатов, где были ранее замечены проблемы, в то время как в моменты движения между ними робот не наблюдал их и опирался исключительно на локализацию AMCL.

До внедрения данной системы, во время ошибок локализации сотрудникам, страхующим работу робота, приходилось вмешиваться в ход экскурсии: вручную выправлять движение робота при помощи джойстика и указывать роботу его истинное положение с помощью графического интерфейса. После внедрения данной системы эта проблема отпала, что косвенно (поскольку вносились и другие изменения в ПО и конструкцию робота) подтверждается статистикой успешности экскурсий (Рис. 3.20).

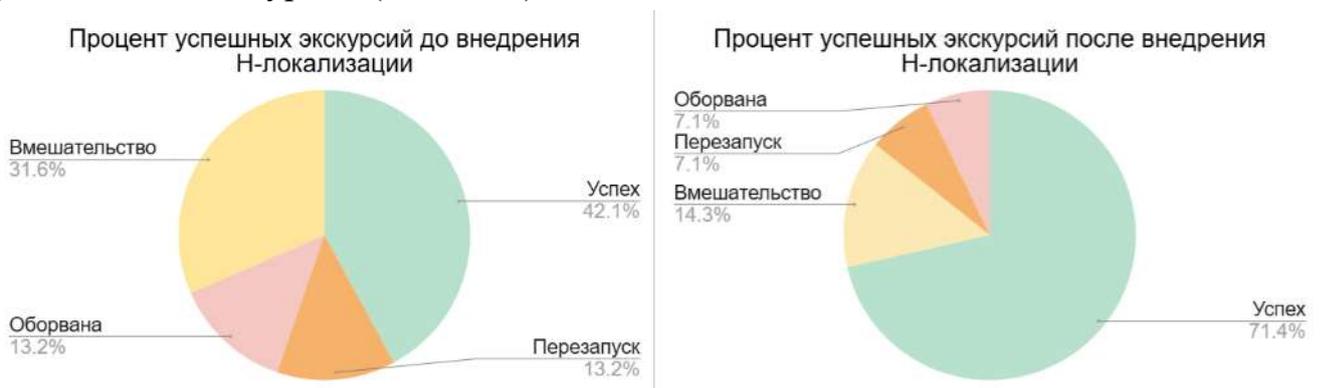


Рис. 4.20. Успешность экскурсий до (слева, 37 экскурсий) и после (справа, 29 экскурсий) внедрения метода N-локализации

Проблемы, связанные с локализацией в толпе, обычно приводили к ситуации «Вмешательство» или «Перезапуск». На графиках на Рис. 4.20 видно,

что их процент которых значительно уменьшился после внедрения Н-локализации, но не исчез совсем, поскольку присутствовал и ряд других проблем, например, связанный с навигацией или отказом какого-то оборудования. Также, помимо ошибок локализации, внедрение метола позволило решить проблему неоднозначности глобальной локализации в симметричном помещении, которой посвящена диссертационная работа. Аналогичная проблема глобальной локализации в том же помещении, но решаемая с помощью «естественных» объектов, а не искусственных маркеров, рассматривается в подразделе 3.3.3.

Интеграция разработанного метода Н-локализации в роботов YARP8, YARP13 и MENTOR-1 была выполнена в рамках выполнения работ по тематике 1п.1.2 подтемы 1П «Разработка робототехнических комплексов, в том числе энергетически автономных» по приказу №86 от 20.01.2023 г. «О проведении фундаментальной научно-исследовательской работы по комплексной теме "Прикладные междисциплинарные исследования в области создания природоподобных технологий" НИЦ «Курчатовский институт», получен соответствующий акт о внедрении (см. Приложение 1).

4.3. Исследование комплексной технологии глобальной локализации

Данный раздел посвящен вопросам тестирования и апробации комплексной технологии локализации целиком, включая как эксперименты на открытом наборе данных так и на реальном роботе.

4.3.1. Глобальная локализация на полученном наборе данных KITTI-360

Эксперименты по глобальной локализации с использованием предложенных алгоритмов распознавания сцен и методов Н-локализации также были проведены на наборе данных KITTI-360 (раздел 3.4). В качестве входных данных был использован выход алгоритма распознавания сцен на основе поиска изоморфного подграфа с прямым сравнением отношений и визуальным сходством на основе CLIP (*sig_lite_clip* из раздела 4.1), как показавшего одни из лучших результатов по введённым критериям точности (Рис. 4.7). Эксперименты были проведены на последовательности 02 из набора KITTI-360, который отражает движение в городской среде. Среди ответов метода распознавания сцен было выбрано 10 первых (лучших) результатов \tilde{f} (2.81). Все входные результаты были разбиты на

группы по значениям *Degree* (2.85) – *ИСТИНА*, *ПРАВДОПОДОБИЕ* и *ЛОЖЬ*. Группа *ПУСТО* не рассматривалась в исследовании, поскольку без результатов от системы распознавания невозможно сформулировать задачу локализации по ориентирам (раздел 2.2.1).

Поскольку одной из ключевых идей применения Н-локализации является сокращение области поиска, то были рассчитаны следующие оценки полученного ответа в виде мульти-интервалов:

- *точность интервалов* – содержание верного ответа (положения робота) в полученных ограничениях;
- *охват интервалов* – относительный размер полученных ограничений к общей площади всей карты.

Для расчёта *охвата интервалов* была использована следующая методика:

1. Рабочая область случайно-равномерно была заполнена частицами-положениями ($N = 1\,000\,000$), описываемыми тройкой (x, y, γ) – положением и углом. Рабочая область определялась в радиусе 50 м вокруг каждого положения камеры в исходном наборе данных.
2. Определялись частицы, принадлежащие полученным мульти-интервалам. Отношение определялось как число принадлежащих частиц к общему их числу.

Также замерялось время работы. Рассматривалась работа алгоритма глобальной Н-локализации (Табл. 2.10) с параметром $N\sigma = 4$ (2.37), функциями интерпретации (2.41)-(2.45), дополнительными функциями проверки корректности (2.50) и (2.51). Отдельно был рассмотрен вариант без модификаций и вариант с дополнительным применением алгоритма исчерпывающего оценивания (*sdm_esis*, Табл. 2.12) с $xu_lim = 10$ м. Средние значения исследуемых параметров для обоих алгоритмов приведены в Табл. 4.4, параметры рассчитывались для объединения результатов для всех 10 гипотез распознанных сцен.

Табл. 4.4. Численные результаты тестирования метода Н-локализации

Величина	Точность интервалов, %, ↑			Охват интервалов, %, ↓			t ,с,↓
	ИСТИНА	ПРАВД.	ЛОЖЬ	ИСТИНА	ПРАВД.	ЛОЖЬ	
Алгоритм\ группа							все
<i>sdm_global_ localization</i>	99.8	100	37	3.22	4.44	0.37	0.03
<i>sdm_global_ localization</i>	97.1	98	17.6	0.23	0.41	0.65	0.69

+ <i>sdm_esis</i>							
-------------------	--	--	--	--	--	--	--

По данным, приведённым в Табл. 4.4, применение алгоритма *sdm_esis* значительно уменьшает полученную область ограничений, до десятых процента от исходного размера, однако увеличивает время исполнения практически на порядок. Также присутствует незначительная разница в содержании верного ответа для групп *ИСТИНА* и *ПРАВДОПОДОБИЕ*.

Далее была исследована способность на основе полученных ограничений определять положение робота с использованием методов локализации, основывающихся на работе с облаками точек. Был предложен следующий алгоритм глобальной локализации по облакам точек на основе ФЧ:

Таким образом, комплексная технология локализации с использованием методов распознавания сцен, метода N-локализации и ФЧ в ограничениях иллюстрируется схемой на Рис. 4.21.

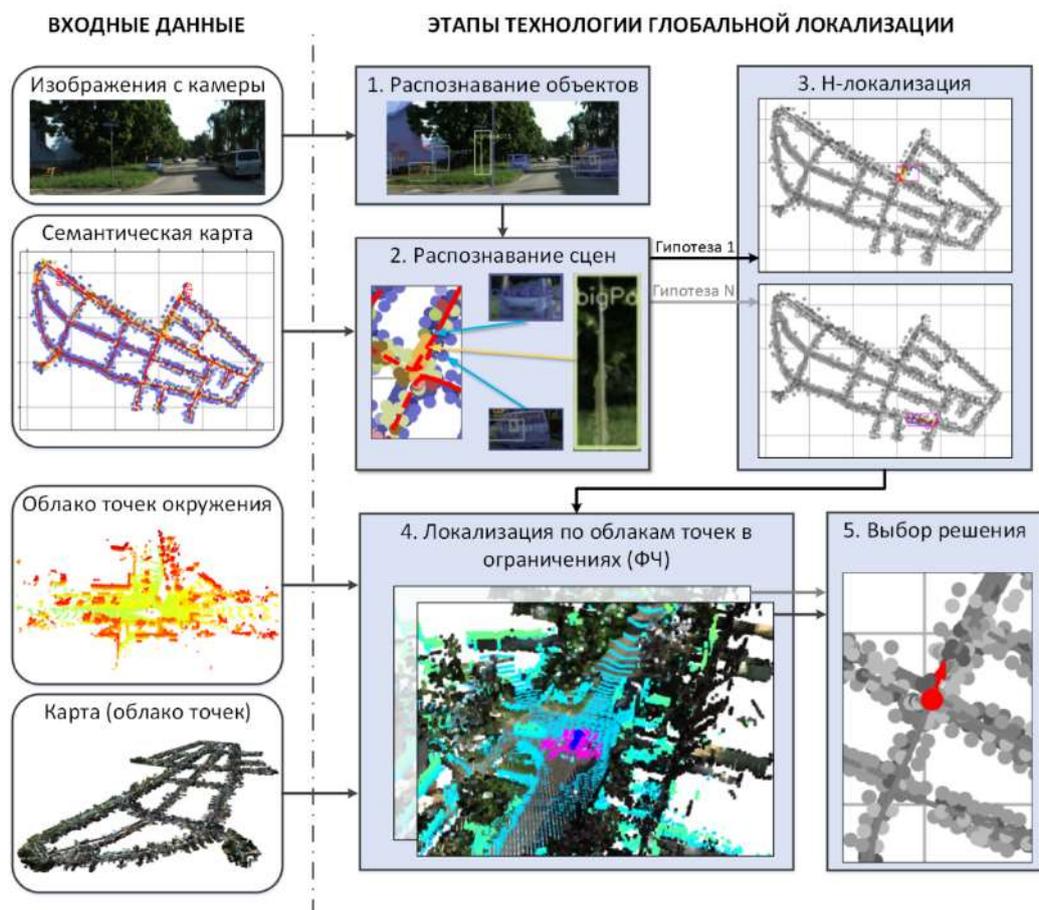


Рис. 4.21. Схема предлагаемой комплексной технологии решения задачи глобальной локализации

Помимо поиска ФЧ также предлагается на основе полученных от N-локализации ограничений напрямую выделить часть карты и применить к ней

классические методы глобальной локализации. Для обрезки карты таким образом к полученным ограничениям на $*X$ и $*Y$ добавляется дальность работы сенсора и определяются те точки карты, что принадлежат новым получившимся ограничениям. В данном варианте такое расширение и игнорирование ограничений на $*\theta$ может привести к тому, что ответ не будет принадлежать полученным ограничениям.

Для тех методов, что работают с выходом систем распознавания сцен, был применен общий алгоритм выделения единственного ответа:

1. Расчёт положения для каждого варианта из 10 отобранных значений.
2. Выбор одного положения на основе численной оценки решения, предлагаемого методом.

В рассмотрение попали следующие варианты процедуры локализации:

- **RANSAC+ICP**. Классический подход к локализации исключительно по облакам точек с расчетом FPFH-дескрипторов [Rusu, Blodow, Beetz, 2009], поиском их соответствий методом RANSAC [Fishler, Bolles, 1980] и дальнейшее уточнение с помощью ICP [Rusinkiewicz, Levoy, 2001]. Данный подход был выбран для рассмотрения ввиду доступных открытых реализаций. Т.к. карта местности в наборе данных KITTI-360 изначально разбита на подкарты, то соответствия методом RANSAC проводились для каждой подкарты, а уточнение ICP для той позиции, что показало наибольшую меру *fitness* (численная оценка качества регистрации облаков точек, используемая в методах RANSAC и ICP).
- **SVD**. Локализация по данным распознавания сцен методом SVD [Arun, Huang, Blostein, 1987]. Качество решения для разных результатов распознавания сцен определялось средней ошибкой положения каждого объекта между сценой и картой.
- **SVD+ICP**. Локализация по данным распознавания сцен методом SVD и дальнейшее уточнение полученного положения с помощью ICP по облакам точек. В качестве численной оценки использовался параметр *fitness* метода ICP.
- **SDM+PF**. Предлагаемая в работе технология глобальной N-локализации по ориентирам алгоритмами **sdm_global_localization** (Табл. 2.10, $N\sigma = 4$) с алгоритмом исчерпывающего оценивания **sdm_esis** (Табл. 2.12, $xu_{lim} = 10\text{м}$) и ФЧ в ограничениях **pf_global_localization** (Табл. 2.16, $N = 150$, $\sigma = 0.1$, $iter = 5$, $\Delta = (0.5, 0.1)$).

- **SDM+RANSAC+ICP**. Подход **RANSAC+ICP** (Табл. 2.13), который применялся к облаку точек, полученному на основе рассчитанных ограничений алгоритмом **sdm_global_localization** (Табл. 2.10, $N\sigma = 4$). Рассчитанные ограничения были расширены на дальность работы дальномера (20м). В качестве меры оценки решений также бралось значение *fitness* метода ICP.

Алгоритмы работы с облаками точек были реализованы на языке Python с использованием библиотеки обработки облаков точек Open3D¹⁷. Замерялись такие параметры, как средняя ошибка по расстоянию, медианное среднее ошибки расстояния, среднее и медианное время исполнения, а также значения $recall@k_L$ ($R@k$) с доверительным интервалом $L = 20$ м. Такая метрика часто используется в работах по глобальной локализации, она идеологически похожа на одноименную метрику для оценки качества распознавания сцен (2.83), и рассчитывается как:

$$recall@k_L(\{p\}, \tilde{p}) = \begin{cases} 1, & \min(\{d(p_i, \tilde{p})\}_{p_i \in \{p\}}) \leq L, \\ 0, & otherwise \end{cases}, \quad (4.1)$$

где p – положение, рассчитанное методом, \tilde{p} – истинное положение, d – функция расчета ошибки. Численные результаты выбранных критериев для группы *ИСТИНА* представлены в Табл. 4.5.

Табл. 4.5. Численные результаты методов для группы решений *ИСТИНА*

Метод\величина	$ r $, м, ↓	$M(r)$, м, ↓	$ \alpha $, рад, ↓	$M(\alpha)$, рад, ↓	$ t $, сек, ↓	$M(t)$, сек, ↓	$R@1$, %, ↑	$R@5$, %, ↑
RANSAC+ICP	233.3	190.1	0.71	0.12	4418.1	4171.5	37	-
SVD	4.7	2.3	0.26	0.06	0.005	0.004	94.4	95.6
SVD+ICP	39.5	1.6	0.29	0.06	18.6	17.7	94.9	96.1
SDM+PF	83.7	2.1	0.39	0.06	95.1	97.1	98.4	99.3
SDM+RANSAC+ ICP	100.6	1.4	0.28	0.05	1009	976	69.4	83.7

На основе данных из Табл. 4.5 можно сделать вывод, что распространенный подход SVD без привязки к облакам точек значительно превосходит остальные методы практически по всем параметрам. Это связано с тем, что коэффициент уверенности, использующийся для нормирования ответов в методах распознавания сцен, идеологически совпадает с метрикой оценки метода SVD. Поэтому для группы *ИСТИНА* этот метод практически не ошибается с выбором гипотезы, беря самую высокую по коэффициенту уверенности, которая также является и правильной. Это тоже видно в сравнении с расширением этого метода SVD+ICP, где значительно выросла средняя ошибка, т.к. появились сцены, где

¹⁷ <https://www.open3d.org>

была предпочтена другая гипотеза. Однако медианная ошибка в сравнении с SVD была уменьшена. Это говорит о том, что более чем в половине случаев ICP смог улучшить положение по данным с облаков точек. Другие методы, использующие результаты распознавания сцен, показывают сравнимую медиану, но сильно увеличенное среднее за счёт неверно выбранных гипотез, которые могут находиться на других концах карты, значительно влияя на среднее. Это подтверждается также и превосходящими значениями $R@k$ для метода SDM+PF.

Результаты для группы *ПРАВДОПОДОБИЕ* представляют больший интерес в рамках данной работы, поскольку отражают возможность находить более верный ответ в наборе близких гипотез. Численные результаты для данной группы приведены в Табл. 4.6.

Табл. 4.6. Численные результаты методов для группы решений

ПРАВДОПОДОБИЕ

Метод\величина	$ r $, м, ↓	$M(r)$, м, ↓	$ \alpha $, рад, ↓	$M(\alpha)$, рад, ↓	$ t $, сек, ↓	$M(t)$, сек, ↓	$R@1$, %, ↑	$R@5$, %, ↑
RANSAC+ICP	207.5	158.9	1	0.46	4038.3	4002	27.7	-
SVD	191.1	28.9	1.15	0.83	0.005	0.004	44.3	85.8
SVD+ICP	38.4	2	0.35	0.04	19.6	19.6	44.5	85.9
SDM+PF	80.4	1.5	0.29	0.05	94.8	95.6	62.2	63.9
SDM+RANSAC+ ICP	71.5	6.7	0.22	0.03	911.2	878.3	61.5	84.6

В Табл. 4.6 в сравнении с Табл. 4.5 метод SVD, который показывал лучшие результаты, значительно потерял в точности и демонстрирует одни из худших показателей. Это происходит из-за того, что он не использует дополнительные данные в виде облаков точек, как это делают остальные методы, что подтверждает важность предлагаемой комплексной технологии. Однако добавление коррекции с помощью ICP позволило значительно поднять точность, получив практически лучшие результаты по всем параметрам и хорошие временные показатели. Параметры точности методов, основанных на результатах распознавания сцен и уточнения положения по облакам точек, в целом не сильно отличаются от группы *ИСТИНА* (Табл. 4.5), что говорит о возможности нахождения решения среди множества близких гипотез. Сильное падение метрики $R@1$ обусловлено неправильностью лучшего ответа в группе *ПРАВДОПОДОБИЕ*, падение $R@5$ также происходит потому, что правильный ответ мог не попасть в лучшие 5 решений.

Группа решений, отнесенных к категории *ЛОЖЬ*, где нет полностью верного ответа, также интересна в задаче локализации по визуальным ориентирам, поскольку постановка содержит заведомо ложные данные. Численные результаты представлены в Табл. 4.7.

Табл. 4.7. Численные результаты методов для группы решений *ЛОЖЬ*

Метод\величина	$ r $, м, ↓	$M(r)$, м, ↓	$ \alpha $, рад, ↓	$M(\alpha)$, рад, ↓	$ t $, сек, ↓	$M(t)$, сек, ↓	$R@1$, %, ↑	$R@5$, %, ↑
RANSAC+ICP	289	262.3	0.87	0.06	4501	4255	33.3	-
SVD	351	298.3	1.22	0.92	0.004	0.004	24.3	30.7
SVD+ICP	206.7	95.2	1.13	0.86	17.8	17.2	24.5	30.7
SDM+PF	203.4	150.8	1.34	1	79.1	83.4	25.7	25.7
SDM+RANSAC+ ICP	210.5	285.5	0.59	0.07	1076.6	1030.9	13.3	20

Из данных в Табл. 4.7 можно видеть, что отсутствие верного ответа от системы распознавания сцен отрицательно влияет на результаты, что в очередной раз показывает важность создания робастных методов распознавания сцен. Предложенный метод N-локализации на основе функций проверки корректности позволяет отбрасывать некоторые из гипотез: в среднем было отброшено 17.5% всех обработанных гипотез, и эта величина не сильно варьируется в пределах исследуемых групп (16.6, 20.1 и 18.5% соответственно).

Общие показатели $recall@1$ и $recall@5$ без деления на подгруппы представлены на Рис. 4.22.

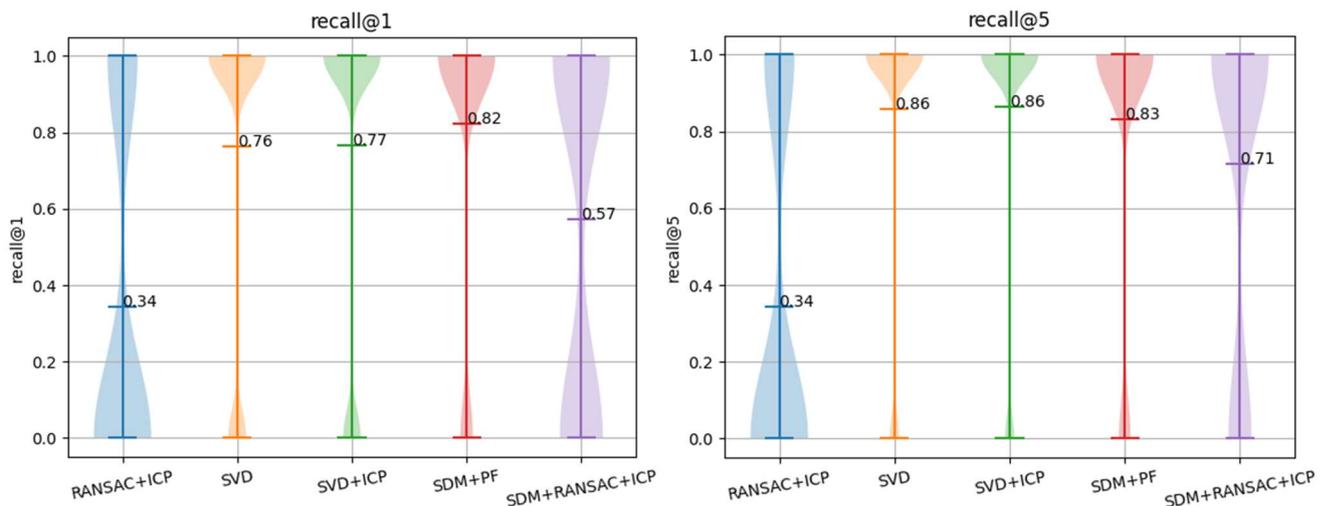


Рис. 4.22. Распределения и средние значения $recall@k$ для исследованных методов

Используя смысл, заложенный в метрике $recall@1$, были рассчитаны средние ошибки положения, таким образом, что значения, превышающие 20м были опущены до этой величины (Рис. 4.23).

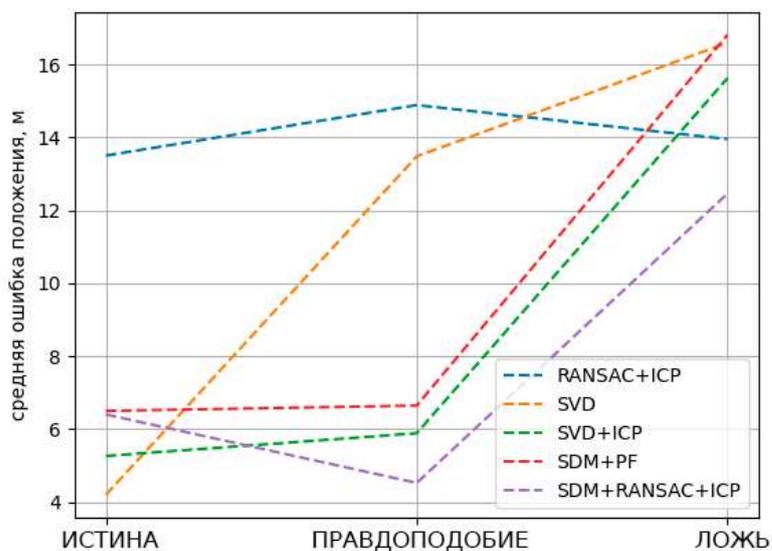


Рис. 4.23. Средняя ошибка методов при пороге ошибки, равном 20 м

График на Рис. 4.23 ещё раз показывает сильное падение точности метода SVD, который не пользуется данными об облаках точек, в зависимости от группы. В то время как методы, использующие облака точек в дополнение к распознаванию сцен, имеют схожие результаты для групп *ИСТИНА* и *ПРАВДОПОДОБИЕ*.

Таким образом, предложенные методы на основе сокращения области поиска (SDM+PF и SDM+RANSAC+ICP) показывают сравнимые по точности результаты с подходом на основе SVD, в некоторых случаях превосходя их. Это говорит о перспективности данного направления для задач глобальной локализации. Временные параметры, приведённые в данном эксперименте, носят скорее ознакомительный характер, поскольку при реализации данных методов не вставали вопросы, связанные с программной оптимизацией. Подход на основе ФЧ с использованием расстояния между ближайшими точками для сравнения облаков обладает ограничением, что показывает хорошие результаты только вблизи верного ответа и с небольшими отклонениями по углу. Исследование и применение более чувствительных метрик сравнения облаков точек, в т.ч. с учетом их цветовых характеристик, также потенциально улучшит результат. Также стоит отметить, что ввиду ограничения доступных вычислительных ресурсов не была проведена многокритериальная оптимизация параметров используемых методов, и они были выбраны эмпирическим путём. Метод SDM+RANSAC+ICP показал как значительное увеличение точности, так и

уменьшение времени вычисления в сравнении с классическим подходом RANSAC+ICP. Это говорит о возможности использования данной технологии для других методов глобальной локализации по облакам точек, исследование которых, однако, находится за рамками данной диссертационной работы.

4.3.2. Применение комплексной технологии на реальном роботе

Предлагаемая комплексная технология была апробирована на роботе RItRover (Robotic Intellectual Rover, Рис. 4.24), использующегося в лаборатории робототехники НИЦ «Курчатовский институт» как платформа для различных исследований, например человеко-машинного взаимодействия [Карпов, Ровбо, Сорокоумов, 2023].



Рис. 4.24. Робот RItRover

Робот оснащен трехмерным лидаром Unitree L1 и стереокамерой ОАК-D. Сенсорное обеспечение позволяет проверить предлагаемую комплексную технологию глобальной локализации в полной мере: выполнить распознавание и локализацию объектов, распознать сцену, построить области ограничений и выполнить в них поиск положения с использованием облаков точек.

Для распознавания объектов была использована свёрточная модель YOLOv7 [Wang, Vochkovskiy, Liao, 2023], которая была обучена распознавать объекты, находящиеся в помещении, использованном для тестов. Были выбраны следующие объекты: стол, стул, растение, дверь, плакат, шкаф, ступени. Для обучения использовались данные из Open Image Dataset [Kuznetsova и др., 2020], а также небольшое число вручную размеченных фотографий помещения. Запуск

сети происходил через средство EOD (раздел 3.3.1). Все объекты состояли из атрибута детектирования образа посредством указанной нейросети, атрибута проверки семантической метки, а также атрибута расстояния, на основе карты глубины, полученной от стерео-камеры ОАК-D. Трёхмерная карта помещения была построена при помощи средства Point-LIO¹⁸. На основе трехмерного слоя был вручную сформирован семантический слой (Рис. 4.25), содержащие выбранные объекты за исключением стульев (часто меняют своё положение) и растений (высокие погрешности локализации ввиду их «разреженной» структуры).

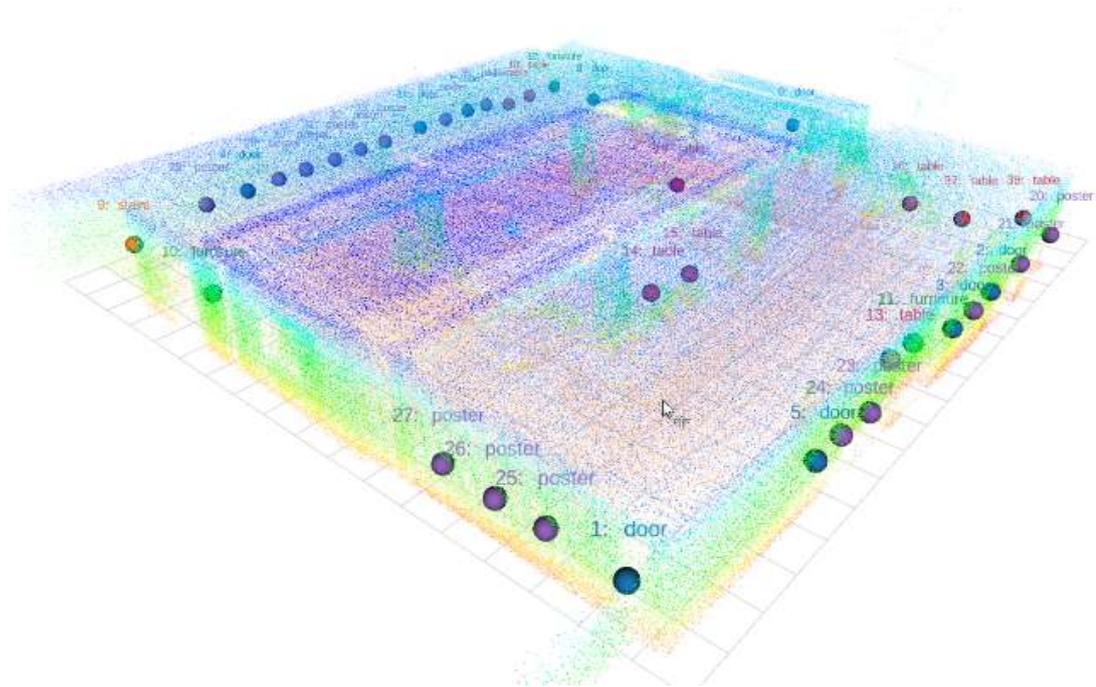


Рис. 4.25. Трёхмерный и семантический слой карты помещения, используемого для апробации комплексной технологии глобальной локализации на роботе RItRover

Посредством ассоциированных с положением объектов их фотографий с разных ракурсов были рассчитаны усреднённые CLIP-признаки объектов карты (2.24). В качестве средства непрерывной трехмерной локализации было выбрано HDL-Localization [Koide, Miura, Menegatti, 2019], находящееся в открытом доступе¹⁹. Это средство работает вкуче со средством HDL-Global-Localization²⁰, которое позволяет выполнить глобальную локализацию на основе облаков точек несколькими методами. В эксперименте с RItRover средство HDL-Global-

¹⁸ https://github.com/unitreerobotics/point_lio_unilidar

¹⁹ https://github.com/koide3/hdl_localization

²⁰ https://github.com/koide3/hdl_global_localization

Localization было заменено предлагаемой комплексной технологией, сохранив исходные интерфейсы (Рис. 4.26).

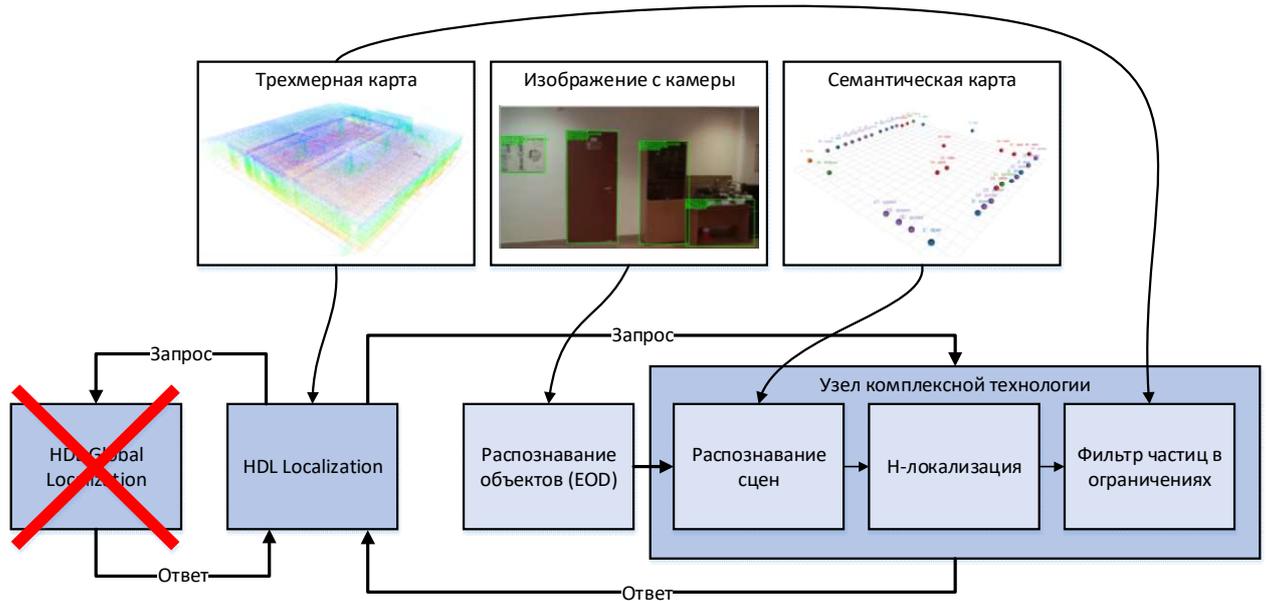


Рис. 4.26. Схема комплексной технологии глобальной локализации, примененной на роботе RItRover

Узел HDL-Localization получает запрос на глобальное определение своего положения, посылает `ros-service` запрос узлу глобальной локализации с указанием текущего скана (облако точек) и получает в ответ положение, которое потом дополнительно уточняется методом NDT.

Узел комплексной технологии получает от EOD набор распознанных и локализованных по стереокамере объектов. Далее для полученных объектов рассчитываются CLIP-признаки и формируется сцена. Сцена подается на вход реализации алгоритма `sig_soft` (Табл. 2.8), которая формирует набор гипотез о соответствии наблюдаемым объектам семантической карте $\{f\}$. Для каждой из лучших $k=3$ гипотез формируется задача локализации по ориентирам. Алгоритм `sdm_global_localization` (Табл. 2.10) решает каждую задачу локализации по ориентирам с использованием алгоритма исчерпывающего оценивания `sdm_esis` (Табл. 2.12). Для каждой из полученных областей, которые не были отброшены N-локализацией, производится определение положение с помощью ФЧ в ограничениях алгоритмом `pf_global_localization` (Табл. 2.16), использующим облака точек. Лучшее по метрике *fitness* из полученных положений, отправляется на узел HDL-Localization.

Некоторые примеры глобальной локализации в приведённых условиях приведены на Рис. 4.27 – Рис. 4.30. На них изображена карта с трехмерным и семантическим слоем и распознанные на изображении объекты. Объекты на семантической карте, отмеченные кубами, соответствуют объектам в одной из распознанных сцен. Фиолетовыми областями отмечены ограничения (дополнительно обведены овалами), выдаваемые Н-локализацией, желтыми стрелками – положение робота, рассчитанного ФЧ для каждой сцены, моделью робота (зеленой фигурой) – финальное положение, уточненное методом HDL-Localization.

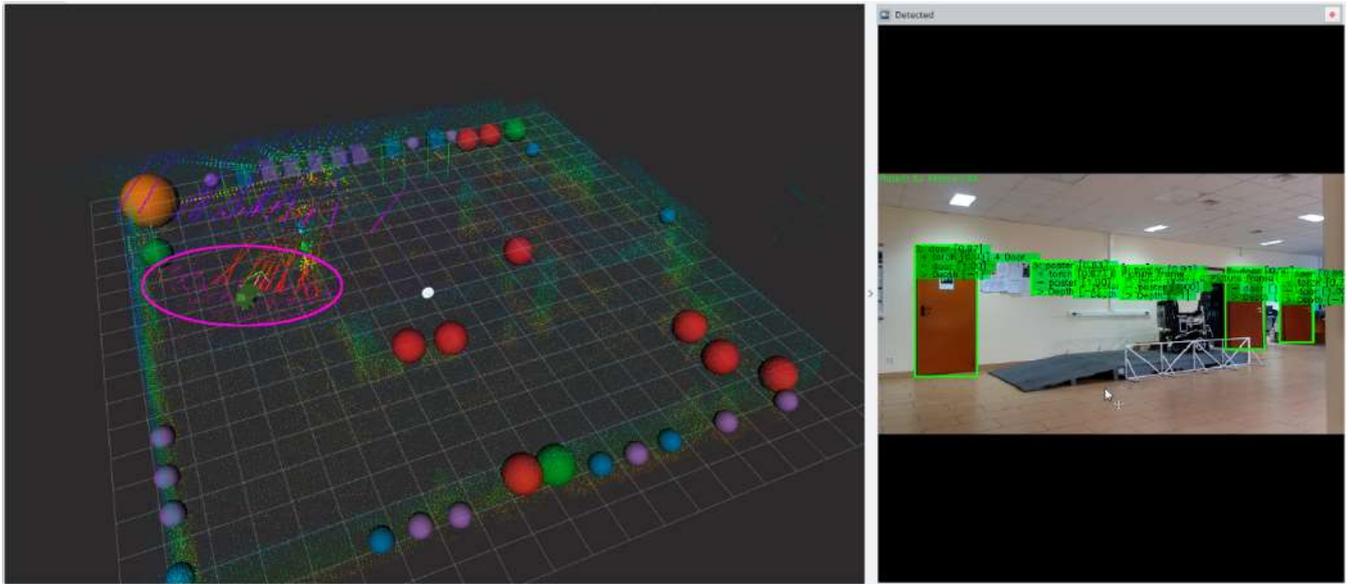


Рис. 4.27. Пример глобальной локализации робота RItRover по сцене из 7 объектов. Была оставлена лишь одна сцена, давшая правильное положение робота

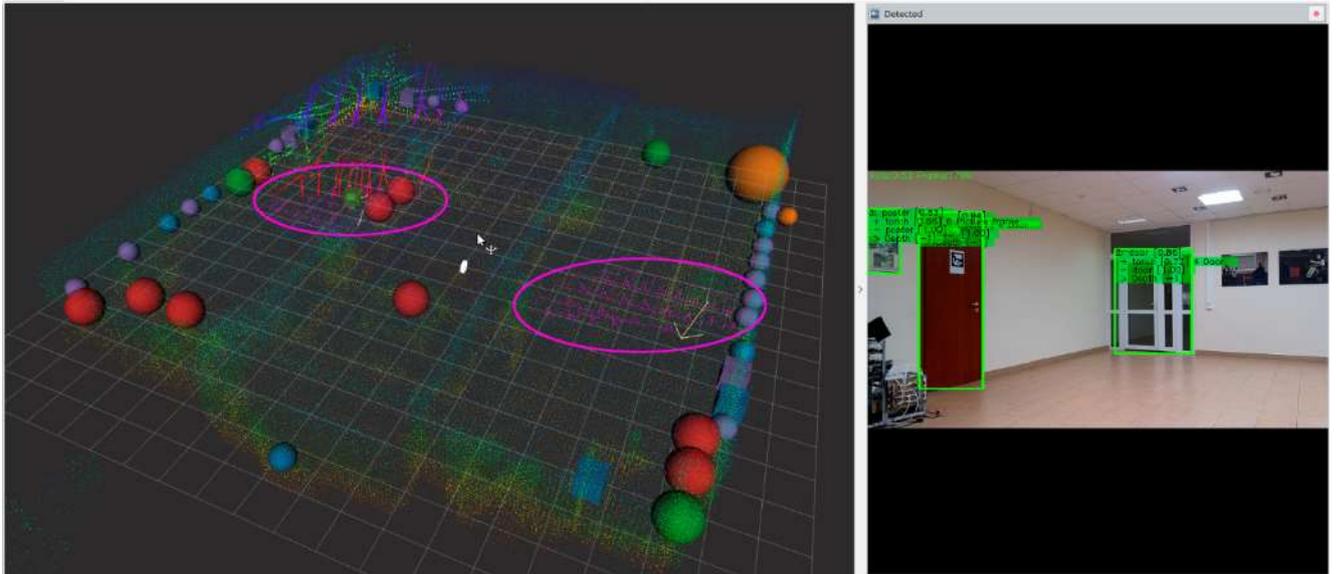


Рис. 4.28. Пример глобальной локализации робота RItRover по сцене из 3 объектов. Было построено две области для двух распознанных сцен, одна из которых была верной и прошла проверку ФЧ

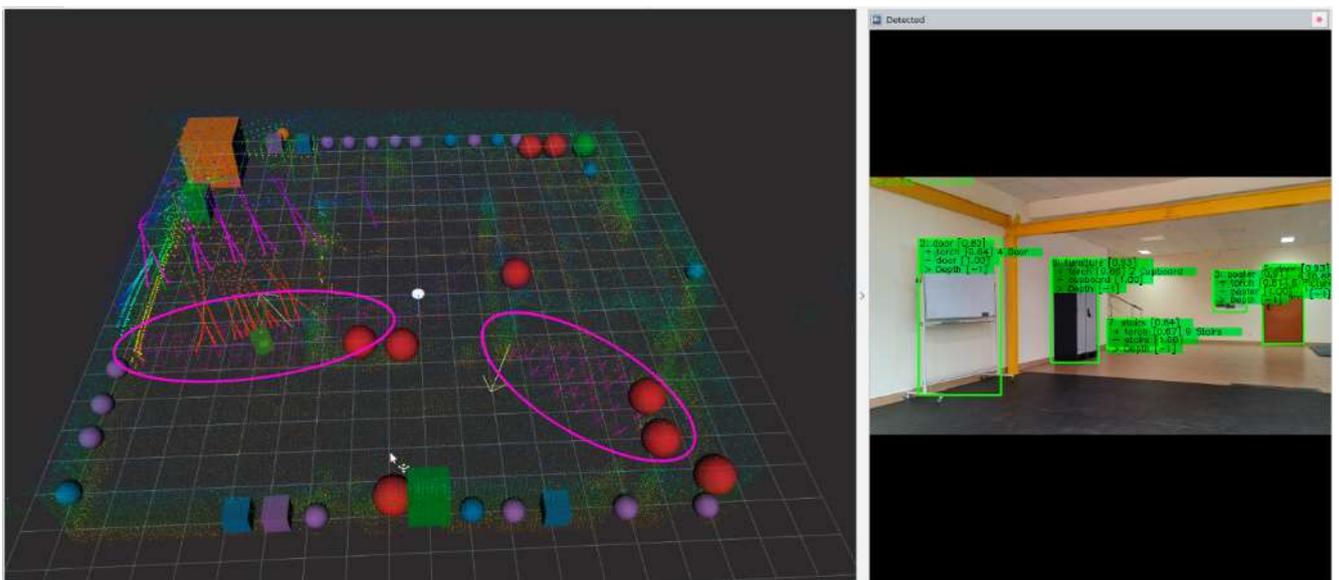


Рис. 4.29. Пример глобальной локализации робота RItRover по сцене из 5 объектов с одним выбросом. Было оставлено 2 сцены, в одной из которых выбросом посчитался объект «дверь» (верно), во втором случае объект «ступени» (не верно), однако на уровне облаков точек, был выбран корректный результат

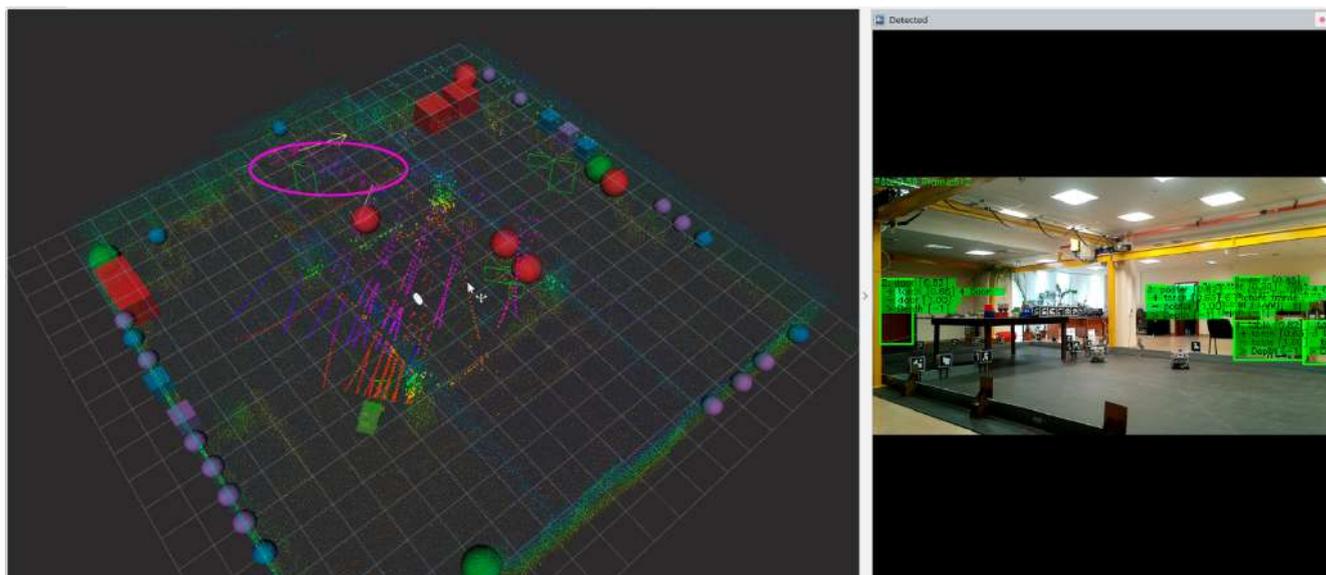


Рис. 4.30. Пример неудачной глобальной локализации робота RItRover по сцене из 6 объектов.

На Рис. 4.30 корректное положение робота (выставлено вручную) не соответствует рассчитанной области. Проблема в неправильном распознавании сцены, ввиду ошибок в локализации объектов (положение объектов отмечено зелеными рамками) из-за особенностей определения карты глубины камеры ОАК-D на больших расстояниях. Такая проблема может быть решена использованием более точного лидара для локализации объектов, либо путем увеличения оценки ошибок локализации и увеличения числа рассматриваемых гипотез.

Таким образом, предлагаемая комплексная технология была проверена на реальном роботе и показала свою практическую применимость в поставленных условиях в симметричном помещении.

4.4. Выводы к главе

В главе приведены результаты исследований предложенных методов распознавания сцен, а также метода N-локализации. Исследования проводились как в средах симуляции, так и открытом наборе данных (адаптированном под решаемую задачу), а апробация – на реальных роботах. В поставленных экспериментах рассматривались различные сценарии, проводилось сравнение с другими методами и алгоритмами. Из полученных результатов были сделаны выводы, что разработанные алгоритмы демонстрируют свою эффективность. Комплексная технология глобальной локализации выполняет поставленные

задачи по сужению области поиска для классических методов локализации по облакам точек.

Заключение

Основной научный результат работы заключается в решении актуальной научно-практической задачи глобальной локализации на больших пространствах, а также на пространствах, характеризующихся симметрией и периодичностью. Достигается это за счёт анализа окружения робота с точки зрения семантики и решения задачи распознавания сцен, а также применения подхода недоопределённых вычислений для опрделеления ограничений на положение робота по визуальным ориентирам. Полученные ограничения могут быть использованы набором классических методов локализации для сокращения пространства поиска.

В диссертационном исследовании решен ряд задач и получены следующие результаты:

1. Проведен анализ современных подходов к задаче распознавания сцен. Сделаны выводы, что подходы развиваются в сторону применения формальных подходов теории графов, позволяющих определять геометрические особенности взаиморасположения объектов на сцене, а также в сторону использования базисных моделей для определения визуального сходства.

2. Было предложено использование модели CLIP для определения визуального сходства наблюдаемых объектов, которое повысило качество решения задачи распознавания сцен по метрике *Degree* (2.85) до 10 раз по числу полностью верно найденных ответов (с 0.04 до 0.41) для метода на основе поиска максимальной клики. Данный подход был предложен одновременно с рядом других научных коллективов, занимающихся задачами распознавания сцен в применении к локализации.

3. Разработаны метод и алгоритмы распознавания сцен на основе алгоритмов поиска изоморфного подграфа, до этого не применявшихся в исследуемой области. Метод позволяет учитывать как геометрические особенности взаиморасположения объектов, так и визуальное сходство объектов. Мягкое расширение метода также позволяет работать с выбросами во входных данных. Метод позволяет получить ряд гипотез, снабженных численной оценкой. Полученный метод показал наибольшую точность (0.48) по метрике *Degree* (2.85) и наибольшую точность (0.57) по метрике *Precision* (2.79) среди исследовавшихся аналогов.

4. Разработан метод N-локализации посредством формализации N-модели к текущей области и разработки соответствующих алгоритмов, позволяющих

проводить локализацию робота по визуальным ориентирам и получать интервальные оценки положения робота (ограничения на положение), а также бороться с противоречиями во входных данных и учитывать погрешности в определении расстояний и угла на объекты. N-локализация может применяться совместно с вероятностными методами локализации, увеличивая как скорость их работы (до 10 раз для гистограммного фильтра) так и точность полученных решений (до 9 раз в случае фильтра частиц).

5. Реализован алгоритм формирования набора данных для тестирования задач распознавания сцен на основе открытого набора данных KITTI-360. Реализация алгоритма и полученные данные находятся в открытом доступе.

6. Разработан ряд метрик, позволяющих оценивать качество решения задачи распознавания сцен в виде множества гипотез.

7. Разработана система распознавания объектов для робототехнических применений на основе мультиатрибутивной парадигмы, позволяющей комбинировать в рамках одного объекта ряд различных методов технического зрения. Реализация данной системы выложена в открытый доступ.

8. Предложена архитектура комплексной технологии глобальной локализации по визуальным ориентирам и дальномерным данным с использованием методов распознавания сцен и подхода N-локализации. Проведённые эксперименты показали увеличение точности определения положения от 5 до 52% в сравнении с существующими методами по метрике *Recall@1* (2.83).

9. Разработанные алгоритмы были апробированы на ряде реальных роботов, разного назначения и обладающих различными сенсорными и вычислительными возможностями. Полученные научные результаты используются в ряде организаций, что подтверждается соответствующими актами об использовании, приведенными в приложении. Получен ряд свидетельств о регистрации программ для ЭВМ и патент на полезную модель. Таким образом, все задачи диссертационного исследования решены в полном объеме.

Перспективы развития. Несмотря на достижение всех поставленных целей, есть ряд направлений, способных развить предложенную технологию. Среди них можно отметить:

- Развитие методов формирования сцены на основе непродолжительного движения для увеличения информативности.
- Исследование процедуры «обрезки» сцены без существенной потери информативности, но с увеличением скорости распознавания.
- Внедрение карт с указанием размеров объектов.

- Внедрение описания объекта, имеющего набор векторов признаков для различных ракурсов.
- Адаптация метода N-локализации на трёхмерное пространство.
- Развитие методов локализации по облакам точек в ограничениях, среди которых:
 - Разработка более робастных способов оценки частиц;
 - Разработка методов класса RANSAC, способных фильтровать гипотезы на основе полученных ограничений, в т.ч. и угловых.

Список используемых сокращений и условных обозначений

Сокращение	Расшифровка	Пояснение
ГФ	Гистограммный фильтр	Вероятностный подход поиска решения в широком классе задач, в т.ч. локализации роботов
ФЧ	Фильтр частиц	Вероятностный подход поиска решения в широком классе задач, в т.ч. локализации роботов
AMCL	Augmented\Adaptive Monte Carlo Localization	Вероятностный метод локализации на основе ФЧ
CLIP	Contrastive Language-Image Pre-training	Архитектура нейронной сети, формирующая общие признаки для изображений и текстового описания
EOD	Extended Object Detection	Разработанное средство распознавания объектов (раздел 3.3.1)
ICP	Iterative Closest Point	Метод регистрации облаков точек
IoU	Intersection over union	Метрика сравнения двух брусков
NDT	Normal distributions transform	Метод регистрации облаков точек
ROS	Robotic Operating System	Фреймворк для разработки программного обеспечения для роботов
RANSAC	RANdom Sample Consensus	Метод регистрации облаков точек
SDM	Sub Definite Models	Н-модели
SDL	Sub Definite Localization	Метод локализации на основе Н-моделей (Н-локализация)
SLAM	Simultaneous Localization and Mapping	Процедуры одновременной локализации и построения карты
SVD	Singular Value Decomposition	Метод регистрации облаков точек

Список терминов

Брус – прямоугольник, со сторонами, параллельными осям координат. Используется как в N-локализации, так и в системах распознавания.

Выбросы (в применении к задаче распознавания сцен) – объекты, которые были обнаружены средствами распознавания, но отсутствующие на семантической карте.

Глобальная локализация – процедура определения своего местоположения, без каких-либо начальных знаний.

Замыкание (от англ. loop-closing) – процедура определения положения, в котором робот уже был, для исправления накопившейся ошибки (в рамках SLAM).

Комплексная технология (глобальной локализации) – предлагаемый в рамках работы процесс решения задачи глобальной локализации, основанный на использовании семантических карт и карт облаков точек. Состоит из процедуры распознавания и локализации объектов, методов распознавания сцен, метода N-локализации, методов локализации по облакам точек в ограничениях (раздел 1.4).

Недоопределённые вычисления (N-вычисления, N-модели) – вычислительная технология, предложенная А.С. Нариньяни.

N-локализация – предложенный в работе метод локализации по ориентирами, основанный на теории Недоопределённых вычислений.

Непрерывная локализация – процедура определения своего положения с некоторой периодичностью. Обычно использует данные о предыдущем положении и смещении робота.

Распознавание сцены – задача поиска соответствий между сценой и семантической картой. Постановка приведена в разделе 2.1.

Семантическая карта (семантический слой карты) – множество объектов, имеющих семантическую метку и положение в пространстве (2.3).

Сцена – наблюдаемый фрагмент окружающей среды, представленный как набор объектов, их семантических меток и их положений в координатах робота (2.2).

Трёхмерная карта (трёхмерный слой карты) – карта окружающей среды, представленная в виде облака точек.

Список литературы

1. Вентцель Е. С. Теория вероятностей. Москва: Высшая школа, 1999. Вып. 6. 576 с.
2. Tian Y. и др. Resilient and Distributed Multi-Robot Visual SLAM: Datasets, Experiments, and Lessons Learned // IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS). , 2023.
3. Московский А. Д. Распознавание сцен для задачи глобальной локализации робота // Труды 34-й Международной научно-технической конференции «Экстремальная робототехника». СПб.: , 2023. С. 267–274.
4. Московский А. Д. Распознавание сцен для задач локализации мобильного робота в условиях неопределённости // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов XII Международной научно_практической конференции (ИММВ-2024, Коломна, 14-17 мая 2024 г.). В 2-х томах. Коломна: Универсум, 2024а. С. 255–266.
5. Московский А. Д. Применение архитектуры CLIP в задаче распознавания сцен для локализации мобильного робота // Труды 35-й Международной научно-технической конференции «Экстремальная робототехника». , 2024b. С. 287–293.
6. Androsov A., Shary S. IntvalPy — библиотека интервальных вычислений на языке Python // 2022.
7. Ankenbauer J. и др. Global Localization in Unstructured Environments Using Semantic Object Maps Built from Various Viewpoints // 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). : IEEE, 2023. С. 1358–1365.
8. Arun K. S., Huang T. S., Blostein S. D. Least-Squares Fitting of Two 3-D Point Sets // IEEE Trans. Pattern Anal. Mach. Intell. 1987. Т. PAMI-9. № 5. С. 698–700.
9. Barfoot T. D. State Estimation for Robotics. : Cambridge University Press, 2017.
10. Behley J. и др. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences // Proc. of the IEEE/CVF International Conf.~on Computer Vision (ICCV). , 2019.
11. Cao Z. и др. SemanticTopoLoop: Semantic Loop Closure With 3D Topological Graph Based on Quadric-Level Object Map // IEEE Robot. Autom. Lett. 2024. Т. 9. № 5. С. 4257–4264.
12. Carletti V. и др. VF3-Light: A lightweight subgraph isomorphism algorithm and its experimental evaluation // Pattern Recognit. Lett. 2019. Т. 125. С. 591–596.

13. Chen R. и др. Deep Samplable Observation Model for Global Localization and Kidnapping // IEEE Robot. Autom. Lett. 2021. Т. 6. № 2. С. 2296–2303.
14. Chen R. и др. Robust 3D Object Detection Based on Point Feature Enhancement in Driving Scenes // 2024 IEEE Intelligent Vehicles Symposium (IV). : IEEE, 2024. С. 2791–2798.
15. Choy C., Dong W., Koltun V. Deep Global Registration // 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). : IEEE, 2020. С. 2511–2520.
16. Cohen C. J., Koss F. V. Comprehensive study of three-object triangulation / под ред. W. J. Wolfe, W. H. Chun. , 1993. С. 95–106.
17. Cordella L. P. и др. Performance evaluation of the VF graph matching algorithm // Proceedings 10th International Conference on Image Analysis and Processing. : IEEE Comput. Soc, 1999. С. 1172–1177.
18. Cramariuc A. и др. SemSegMap – 3D Segment-based Semantic Localization // 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). : IEEE, 2021. С. 1183–1190.
19. Dai A. и др. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes // Proc. Computer Vision and Pattern Recognition (CVPR), IEEE. , 2017.
20. Davison A. J. FutureMapping: The Computational Structure of Spatial AI Systems // 2018.
21. Dosovitskiy A. и др. CARLA: An Open Urban Driving Simulator // 2017.
22. Dubé R. и др. SegMap: Segment-based mapping and localization using data-driven descriptors // Int. J. Rob. Res. 2020. Т. 39. № 2–3. С. 339–355.
23. Fishler M., Bolles R. Random sample consensus: A paradigm for model fitting applications to image analysis and automated cartography // Proc. Image Underst. Work. 1980. С. 71–88.
24. Fletcher R. Practical methods of optimization; (2nd ed.). , 1987.
25. Font J. M., Batlle J. A. MOBILE ROBOT LOCALIZATION. REVISITING THE TRIANGULATION METHODS // IFAC Proc. Vol. 2006. Т. 39. № 15. С. 340–345.
26. Gawel A. и др. X-View: Graph-Based Semantic Multi-View Localization // IEEE Robot. Autom. Lett. 2018. Т. 3. № 3. С. 1687–1694.
27. Geiger A., Lenz P., Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite // 2012 IEEE Conference on Computer Vision and Pattern Recognition. : IEEE, 2012. С. 3354–3361.

28. Glocker B. и др. Real-time RGB-D camera relocalization // 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). : IEEE, 2013. С. 173–179.
29. Guo X. и др. Semantic Histogram Based Graph Matching for Real-Time Multi-Robot Global Localization in Large Scale Environment // IEEE Robot. Autom. Lett. 2021. Т. 6. № 4. С. 8349–8356.
30. Hewitt R. A. и др. The Katwijk beach planetary rover dataset // Int. J. Rob. Res. 2018. Т. 37. № 1. С. 3–12.
31. Hua B.-S. и др. SceneNN: A Scene Meshes Dataset with aNnotations // 2016 Fourth International Conference on 3D Vision (3DV). : IEEE, 2016. С. 92–101.
32. Hughes N., Chang Y., Carlone L. Hydra: A Real-time Spatial Perception System for 3D Scene Graph Construction and Optimization // 2022.
33. Jaulin L., Didrit O., Walter E. Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics. , 2001.
34. Jiang Y., Ehambram A., Wagner B. Interval-based Robot Localization with Uncertainty Evaluation // Proceedings of the 19th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO,. : SciTePress, 2022. С. 296–303.
35. Jüttner A., Madarasi P. VF2++—An improved subgraph isomorphism algorithm // Discret. Appl. Math. 2018. Т. 242. С. 69–81.
36. Koide K., Miura J., Menegatti E. A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement // Int. J. Adv. Robot. Syst. 2019. Т. 16. № 2.
37. Komorowski J. MinkLoc3D: Point Cloud Based Large-Scale Place Recognition // 2021 IEEE Winter Conference on Applications of Computer Vision (WACV). : IEEE, 2021. С. 1789–1798.
38. Kong X. и др. Semantic Graph Based Place Recognition for 3D Point Clouds // 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). : IEEE, 2020. С. 8216–8223.
39. Kuznetsova A. и др. The Open Images Dataset V4 // Int. J. Comput. Vis. 2020. Т. 128. № 7. С. 1956–1981.
40. Lafuente-Arroyo S. и др. Multisensory integration for topological indoor localization of mobile robots in complex symmetrical environments // Expert Syst. Appl. 2024. Т. 241. С. 122561.
41. Leordeanu M., Hebert M. A spectral technique for correspondence problems using pairwise constraints // Tenth IEEE International Conference on Computer Vision

(ICCV'05) Volume 1. : IEEE, 2005. C. 1482- 1489 Vol. 2.

42. Levenberg K. A METHOD FOR THE SOLUTION OF CERTAIN NON – LINEAR PROBLEMS IN LEAST SQUARES // Q. Appl. Math. 1944. T. 2. C. 164–168.

43. Liao Y., Xie J., Geiger A. KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D // IEEE Trans. Pattern Anal. Mach. Intell. 2023. T. 45. № 3. C. 3292–3310.

44. Liu Y. и др. Global Localization with Object-Level Semantics and Topology // 2019 International Conference on Robotics and Automation (ICRA). : IEEE, 2019a. C. 4909–4915.

45. Liu Y. и др. Global Localization with Object-Level Semantics and Topology // 2019 International Conference on Robotics and Automation (ICRA). : IEEE, 2019b. C. 4909–4915.

46. Lusk P. C., Fathian K., How J. P. CLIPPER: A Graph-Theoretic Framework for Robust Data Association // 2021 IEEE International Conference on Robotics and Automation (ICRA). : IEEE, 2021. C. 13828–13834.

47. Lusk P. C., How J. P. CLIPPER: Robust Data Association Without an Initial Guess // IEEE Robot. Autom. Lett. 2024. T. 9. № 4. C. 3092–3099.

48. Marquardt D. W. An Algorithm for Least-Squares Estimation of Nonlinear Parameters // J. Soc. Ind. Appl. Math. 1963. T. 11. № 2. C. 431–441.

49. Masoumian A. и др. Absolute Distance Prediction Based on Deep Learning Object Detection and Monocular Depth Estimation Models. , 2021.

50. Matsuzaki S. и др. Single-shot global localization via graph-theoretic correspondence matching // Adv. Robot. 2024a. T. 38. № 3. C. 168–181.

51. Matsuzaki S. и др. CLIP-Loc: Multi-modal Landmark Association for Global Localization in Object-based Maps // 2024 IEEE International Conference on Robotics and Automation (ICRA). : IEEE, 2024b. C. 13673–13679.

52. Matsuzaki S., Tanaka K., Shintani K. CLIP-Clique: Graph-Based Correspondence Matching Augmented by Vision Language Models for Object-Based Global Localization // IEEE Robot. Autom. Lett. 2024. T. 9. № 11. C. 10399–10406.

53. Melekhin A. и др. ITLP-Campus: A Dataset for Multimodal Semantic Place Recognition. , 2024. C. 185–195.

54. Mildenhall B. и др. NeRF: representing scenes as neural radiance fields for view synthesis // Commun. ACM. 2022. T. 65. № 1. C. 99–106.

55. Mirjalili R., Krawez M., Burgard W. FM-Loc: Using Foundation Models for

- Improved Vision-Based Localization // 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). : IEEE, 2023. С. 1381–1387.
56. Moscovsky A. Subdefinite Computations for Reducing the Search Space in Mobile Robot Localization Task // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) / под ред. S. M. Kovalev, S. O. Kuznetsov, A. I. Panov. Cham: Springer International Publishing, 2021. С. 180–196.
57. Moscovsky A. D. Extended Object Detection: Flexible Object Description System for Detection in Robotic Tasks // Smart Electromechanical Systems: Recognition, Identification and Modeling. , 2022a. С. 27–43.
58. Moscovsky A. D. Multi-Attribute and Graph-based Object Detection // LearnOpenCV [Электронный ресурс]. URL: <https://learnopencv.com/multi-attribute-and-graph-based-object-detection> (дата обращения: 01.02.2022b).
59. Moscovsky A. D. и др. Review of Methods for Autonomous Rough Terrain Traversal of Mobile Robots // Autom. Remote Control. 2025. Т. 86. № 2.
60. Munkres J. Algorithms for the Assignment and Transportation Problems // J. Soc. Ind. Appl. Math. 1957. Т. 5. № 1. С. 32–38.
61. Peterson M. B. и др. ROMAN: Open-Set Object Map Alignment for Robust View-Invariant Global Localization // 2024.
62. Pierlot V., Droogenbroeck M. Van. A New Three Object Triangulation Algorithm for Mobile Robot Positioning // IEEE Trans. Robot. 2014. Т. 30. № 3. С. 566–577.
63. Pramatarov G. и др. BoxGraph: Semantic Place Recognition and Pose Estimation from 3D LiDAR // 2022.
64. Qin C. и др. Semantic loop closure detection based on graph matching in multi-objects scenes // J. Vis. Commun. Image Represent. 2021. Т. 76. С. 103072.
65. Radford A. и др. Learning Transferable Visual Models From Natural Language Supervision // International Conference on Machine Learning. , 2021. С. 8748–8763.
66. Raguram R., Frahm J.-M., Pollefeys M. A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus. , 2008. С. 500–513.
67. Ros G. и др. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). : IEEE, 2016. С. 3234–3243.
68. Rosen D. M. и др. Advances in Inference and Representation for Simultaneous Localization and Mapping // Annu. Rev. Control. Robot. Auton. Syst. 2021. Т. 4. № 1.

C. 215–242.

69. Rosinol A. и др. Kimera: From SLAM to spatial perception with 3D dynamic scene graphs // *Int. J. Rob. Res.* 2021. Т. 40. № 12–14. С. 1510–1546.
70. Rovbo M. A., Sorokoumov P. S. Symbolic Control System for a Mobile Robotic Platform Based on Soar Cognitive Architecture // *Smart Electromechanical Systems.* , 2022. С. 259–275.
71. Rusinkiewicz S., Levoy M. Efficient variants of the ICP algorithm // *Proceedings Third International Conference on 3-D Digital Imaging and Modeling.* , 2001. С. 145–152.
72. Rusu R. B., Blodow N., Beetz M. Fast Point Feature Histograms (FPFH) for 3D registration // *2009 IEEE International Conference on Robotics and Automation.* : IEEE, 2009. С. 3212–3217.
73. Saarinen J. и др. Normal distributions transform Monte-Carlo localization (NDT-MCL) // *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems.* : IEEE, 2013. С. 382–389.
74. Saeedi S. и др. Characterizing Visual Localization and Mapping Datasets // *2019 International Conference on Robotics and Automation (ICRA).* : IEEE, 2019. С. 6699–6705.
75. Sahili A. R. и др. A Survey of Visual SLAM Methods // *IEEE Access.* 2023. Т. 11. С. 139643–139677.
76. Shah S. и др. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles // 2017.
77. Shaheer M. и др. Graph-Based Global Robot Localization Informing Situational Graphs with Architectural Graphs // *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* : IEEE, 2023. С. 9155–9162.
78. Sturm J. и др. A benchmark for the evaluation of RGB-D SLAM systems // *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems.* : IEEE, 2012. С. 573–580.
79. Telerman V., Ushakov D. Data types in subdefinite models // *Artificial Intelligence and Symbolic Mathematical Computation / под ред. J. Calmet, J. A. Campbell, J. Pfalzgraf.* Berlin, Heidelberg: Springer Berlin Heidelberg, 1996. С. 305–319.
80. Thrun S., Wolfram B., Fox D. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). , 2005.
81. Tinchev G., Nobili S., Fallon M. Seeing the Wood for the Trees: Reliable Localization in Urban and Natural Environments // *2018 IEEE/RSJ International*

- Conference on Intelligent Robots and Systems (IROS). : IEEE, 2018. С. 8239–8246.
82. Vizzo I. и др. KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way // IEEE Robot. Autom. Lett. 2023. Т. 8. № 2. С. 1029–1036.
83. Wang C.-Y., Bochkovskiy A., Liao H.-Y. M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors // 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). , 2023. С. 7464–7475.
84. Wang Y., Jiang C., Chen X. GOReloc: Graph-Based Object-Level Relocalization for Visual SLAM // IEEE Robot. Autom. Lett. 2024. Т. 9. № 10. С. 8234–8241.
85. Wu Y. и др. An Object SLAM Framework for Association, Mapping, and High-Level Tasks // 2023.
86. Xia L. и др. A survey of image semantics-based visual simultaneous localization and mapping: Application-oriented solutions to autonomous navigation of mobile robots // Int. J. Adv. Robot. Syst. 2020. Т. 17. № 3.
87. Yang H., Shi J., Carlone L. TEASER: Fast and Certifiable Point Cloud Registration // IEEE Trans. Robot. 2021. Т. 37. № 2. С. 314–333.
88. Yeshwanth C. и др. Scannet++: A high-fidelity dataset of 3d indoor scenes // Proceedings of the IEEE/CVF International Conference on Computer Vision. , 2023. С. 12–22.
89. Yin H. и др. A Survey on Global LiDAR Localization: Challenges, Advances and Open Problems // Int. J. Comput. Vis. 2024. Т. 132. № 8. С. 3139–3171.
90. Zemskova T. и др. Interactive Semantic Map Representation for Skill-Based Visual Object Navigation // IEEE Access. 2024. Т. 12. С. 44628–44639.
91. Zhu Y. и др. GOSMatch: Graph-of-Semantics Matching for Detecting Loop Closures in 3D LiDAR data // 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). : IEEE, 2020. С. 5151–5157.
92. Баженов А. Н. и др. Обработка и анализ интервальных данных. Москва, Ижевск: Институт компьютерных исследований, 2024. 356 с.
93. Бова Е. В., Леушина В. В. К вопросу об оценке эффективности модальностей человеко-машинного интерфейса при управлении мобильным робототехническим устройством // Сборник тезисов конференции «Экстремальная робототехника». , 2022. С. 60–62.
94. Бова Е. В., Леушина В. В. Использование биологической обратной связи второго рода в контуре управления робототехническим объектом //

Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов XII Международной научно_практической конференции (ИММВ-2024, Коломна, 14-17 мая 2024 г.). В 2-х томах. Смоленск: Универсум, 2024. С. 75–84.

95. Воробьев В. В. Кибермуравейник программно-аппаратный комплекс для апробации моделей социального поведения в групповой робототехнике // Труды 34-й Международной научно-технической конференции «Экстремальная робототехника». СПб.: , 2023. С. 84–91.

96. Жамбю М., Миркин Б. Г. Иерархический кластер - анализ и соответствия: Пер. с фр. : Финансы и статистика, 1988.

97. Загорюлько Ю. А., Загорюлько Г. Б. Недоопределенные модели Нариньяни: становление, применение, проблемы и перспективы // SORUCOM-20. , 2020. С. 126–132.

98. Захарова А. Н., Московский А. Д. Реализация алгоритмов многоагентной фуражировки на основе деревьев поведения // Материалы 65-й Всероссийской научной конференции МФТИ. Долгопрудный: МФТИ, 2023. С. 58–59.

99. Карпов В. В. и др. Мобильная платформа для проведения подспутниковых измерений. Патент на полезную модель 178972 от 24.04.2018 // 2018.

100. Карпов В. Э. О некоторых особенностях применения недоопределенных моделей в робототехнике // Международная научно-практическая конференция «Интегрированные модели и мягкие вычисления в искусственном интеллекте» (28-30 мая 2009) Сб. научных трудов.Т.1. М.: Физматлит. , 2009. С. 520–532.

101. Карпов В. Э., Ровбо М. А., Сорокоумов П. С. Сигнальная коммуникация для управления поведением коллаборативного робота // Мехатроника, автоматизация, управление. 2023. Т. 24. № 5. С. 260–268.

102. Малышев А. А., Московский А. Д. Антропоморфный робот для проведения экскурсий в лаборатории робототехники // Труды 34-й Международной научно-технической конференции «Экстремальная робототехника». СПб.: , 2023. С. 437–444.

103. Московский А. Д. Метод распознавания сцен на основе недоопределенных моделей // Шестнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2018 (24-27 сентября 2018 г., Москва, Россия). Труды конференции. В 2-х томах. М.: НИУ ВШЭ, 2018. С. 27–34.

104. Московский А. Д. Недоопределённые модели в задаче локализации мобильного робота // Пятый Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта»

- (БТС-ИИ-2019). Санкт-Петербург: Общероссийская общественная организация «Российская ассоциация искусственного интеллекта», 2019а. С. 9–18.
105. Московский А. Д. Система локализации мобильного робота на основе недоопределенных моделей / Программа для ЭВМ, №2019667433 от 11.12.2019 // Свидетельство о государственной регистрации программы для ЭВМ. 2019b.
106. Московский А. Д. Система автоматического формирования семантического слоя карты / Программа для ЭВМ, №2020619217 от 13.09.2020 // Свидетельство о государственной регистрации программы для ЭВМ. 2020.
107. Московский А. Д. Распознавание сцен в задаче глобальной локализации мобильного робота с использованием моделей векторных представлений изображений и графовых подходов // Управление большими системами. 2025а. № 114. С. 307–344.
108. Московский А. Д. Применение недоопределённых моделей в задаче глобальной локализации мобильного робота // Проблемы управления. 2025b. Т. 4. С. 64–78.
109. Московский А. Д. Комплексная технология глобальной локализации мобильного робота по данным семантических карт // Донецкий международный круглый стол «Искусственный интеллект: теоретические аспекты и практическое применение» (ДМКС-ИИ-2025), Донецк, 28 мая 2025 г. Донецк: , 2025с.
110. Нариньяни А. С. Недоопределенность в системах представления и обработки знаний // Известия АН СССР. Техн. кибернетика. 1986. Т. 5.
111. Нариньяни А. С. Введение в недоопределенность // Информационные технологии. 2007. № 4. С. 1–32.
112. Шарый С. П. Конечномерный интервальный анализ // Новосибирск ИВТ СО РАН. 2010.
113. Handbook of Constraint Programming / под ред. F. Rossi, P. van Beek, T. Walsh. : Elsevier, 2006.

Приложение 1. Акты о внедрении результатов диссертационной работы

Экз. №

УТВЕРЖДАЮ
 Директор Электростальского института
 (филиала) Московского
 политехнического университета
 И.З. Вольшонок

» 10 2019 г.



А К Т

**о реализации результатов диссертационных исследований,
 выполненных Московским Антоном Дмитриевичем**

Комиссия в составе: председателя комиссии – заместителя директора Электростальского института (филиала) Московского политехнического университета Филиповой О.Д., членов комиссии начальника учебно-методического отдела Дарымовой Н.С., начальника отдела АСУ Гурова Б.Д., исполняющего обязанности заведующего кафедрой «Прикладная математика и информатика» Ревина С.А, старшего преподавателя кафедры «Прикладная математика и информатика» Жучина А.В., составила настоящий акт о том, что научные результаты, полученные при проведении диссертационного исследования Московским А.Д., реализованы в Электростальском институте (филиале) Московского политехнического университета при проведении НИР «Разработка экспериментального образца аппаратно-программного комплекса сбора и обработки данных для контроля характеристик и калибровки целевой аппаратуры КА ДЗЗ в части создания макета подвижной платформы для проведения подспутниковых наземных радиометрических измерений» (шифр «Мониторинг-СГ-1.3.1.1.1-МПП»).

В ходе работы комиссии установлено, что Московский А.Д в период с 2014 года по 2017 год разработал и реализовал технические, алгоритмические и программные решения, которые применялись при создании экспериментального образца подвижной платформы для проведения подспутниковых измерений (далее ЭО ПП) в следующем виде:

1. Архитектура системы распознавания объектов с не полностью определенными признаками и метод распознавания сцен с использованием парадигмы недоопределённых моделей, используемые при создании системы технического зрения ЭО ПП.

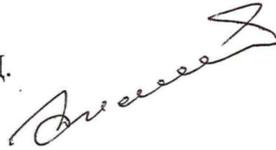
2. Метод локализации по различным источникам данным на основе парадигмы недоопределенных моделей, используемый при создании системы навигации ЭО ПП.

Также Московский А.Д. провел занятия по обучению сотрудников Научного центра Оперативного мониторинга Земли работе с ЭО ПП.

Разработанные Московским А.Д. методы и алгоритмы направлены на автоматизацию сбора данных ЭО ПП с наземных подспутниковых полигонов и были использованы при подготовке программной и конструкторской документации на изделия ЭПИФ.411711.001 (макет подвижной платформы) и ЭПИФ.411711.002 (экспериментальный образец подвижной платформы) (инв. 129156-129209 библиотеки Электростальского института (филиала) Московского политехнического университета).

Председатель комиссии:

Филиппова О.Д.



заместитель директора, кандидат филологических наук, доцент

Члены комиссии:

Дарымова Н.С.



начальник учебно-методического отдела

Гуров Б.Д.



начальник отдела АСУ

Ревин С.А.



исполняющий обязанности заведующего кафедрой «Прикладная математика и информатика», кандидат технических наук, доцент, профессор Академии военных наук

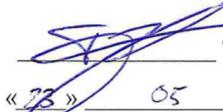
Жучин А.В.



старший преподаватель кафедры «Прикладная математика и информатика», кандидат технических наук, доцент

УТВЕРЖДАЮ

Директор
Физтех-школы природоподобных, плазменных и ядерных технологий
им. И.В. Курчатова Федерального государственного автономного
образовательного учреждения высшего образования
"Московский физико-технический институт
(национальный исследовательский университет)"


Т.Е. Григорьев
« 23 » 05 20 25 г.

АКТ

внедрения в учебный процесс кафедры Информатики и вычислительных сетей Физтех-школы природоподобных, плазменных и ядерных технологий им. И.В. Курчатова (КНТ) результатов диссертационной работы Московского Антона Дмитриевича «Методы и алгоритмы распознавания сцен для задач глобальной локализации мобильных роботов»

Настоящим актом подтверждается, что теоретические разработки и результаты экспериментальных исследований диссертационной работы Московского Антона Дмитриевича «Методы и алгоритмы распознавания сцен для задач глобальной локализации мобильных роботов», опубликованные в печати, внедрены в учебный процесс и используются в лекциях и практических занятиях на кафедре Информатики и вычислительных сетей КНТ.

Результаты диссертации использовались в лекционных курсах «Современные проблемы прикладной математики и информатики» (2024 г., в части методов решения задачи локализации) и «Технологии искусственного интеллекта» (2021-2022 гг., в части методов распознавания сложных объектов и недоопределенных моделей). Московским А.Д. прочитан соответствующий ряд лекций, которые получили положительную оценку и способствовали повышению знаний студентов КНТ.

Заместитель директора КНТ


Давыд Г.


УТВЕРЖДАЮ
 И.о. руководителя КК НБИКС-ПТ

 С.А. Мирошниченко
 «24» 06 2025 г.



АКТ
О внедрении результатов диссертационного исследования
Московского Антона Дмитриевича
В ФГБУ "«Национальный исследовательский центр «Курчатовский
институт»"

«24» 06 2025 года

г. Москва

Комиссия, в составе:

Председателя	Первый заместитель руководителя КК НБИКС-ПТ по научной работе	Демин В.А.
Членов комиссии	Заместитель руководителя КК НБИКС-ПТ – ученый секретарь	Тимаева О.И.
	Ведущий научный сотрудник	Рыбка Р.Б.
	Младший научный сотрудник	Чжао Д.Г.

настоящим актом подтверждает, что научные результаты, полученные при проведении диссертационного исследования Московским Антоном Дмитриевичем, были использованы при создании математического и

программного обеспечения следующих устройств: роботизированной многофункциональной коляски, в рамках выполнения работы по НИР «Исполнительные механизмы и системы электродвижения» по теме «Роботизированная многофункциональная коляска для людей с ограниченными возможностями» (приказы НИЦ «Курчатовский институт» (далее – Центр) от 25.06.2019 № 1361 и от 02.07.2020 № 1057)); группировки роботов серии YARP8, робота YARP13, робота-экскурсовода MENTOR-1 и робота RItRover в рамках выполнения работ по тематике 1п.1.2 подтемы 1П «Разработка робототехнических комплексов, в том числе энергетически автономных» (приказ Центра от 20.01.2023 № 86 «О проведении фундаментальной научно-исследовательской работы по комплексной теме Прикладные междисциплинарные исследования в области создания природоподобных технологий»).

Комиссией установлено, что Московский А.Д. за время проведения диссертационных исследований в период с 2015 по 2025 гг. реализовал следующие научно-технические и программные решения:

1. Средство мульти-атрибутного распознавания объектов по данным камеры.
2. Систему комплексной технологии глобальной локализации, включающей в себя средства распознавания сцен, Н-локализацию и модификацию классических методов, позволяющих работать в динамических ограничениях.
3. Программу автоматического семантического картирования.

Характеристики перечисленных решений соответствуют требованиям Технических заданий на выполнение указанных НИР. Эффективность внедрения научных результатов А.Д. Московского отмечена в научно-технических отчётах по указанным НИР за 2021 – 2024 гг.

Председатель комиссии



Демин В.А.

Члены комиссии:



Тимаева О.И.



Рыбка Р.Б.



Чжао Д.Г.

Приложение 2. Свидетельства о государственной регистрации РИД

РОССИЙСКАЯ ФЕДЕРАЦИЯ



ПАТЕНТ
НА ПОЛЕЗНУЮ МОДЕЛЬ

№ 178972

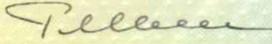
**МОБИЛЬНАЯ ПЛАТФОРМА ДЛЯ ПРОВЕДЕНИЯ
ПОДСПУТНИКОВЫХ ИЗМЕРЕНИЙ**

Патентообладатель: *Акционерное общество "Российская корпорация ракетно-космического приборостроения и информационных систем" (АО "Российские космические системы") (RU)*

Авторы: *Карнов Валерий Валерьевич (RU), Малышев Александр Александрович (RU), Московский Антон Дмитриевич (RU), Ревин Сергей Алексеевич (RU), Ровбо Максим Александрович (RU), Скрыльников Роман Александрович (RU), Тихонычев Виктор Викторович (RU)*

Заявка № 2017113950
Приоритет полезной модели **21 апреля 2017 г.**
Дата государственной регистрации в Государственном реестре полезных моделей Российской Федерации **24 апреля 2018 г.**
Срок действия исключительного права на полезную модель истекает **21 апреля 2027 г.**

Руководитель Федеральной службы по интеллектуальной собственности

 **Г.П. Изrael**



РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2020619217

«Система автоматического формирования семантического
слоя карты»

Правообладатель: *Федеральное государственное бюджетное
учреждение «Национальный исследовательский центр
«Курчатовский институт» (RU)*

Автор: *Московский Антон Дмитриевич (RU)*



Заявка № 2020617712

Дата поступления 22 июля 2020 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 13 августа 2020 г.

Руководитель Федеральной службы
по интеллектуальной собственности

Г.П. Ивлиев

РОССИЙСКАЯ ФЕДЕРАЦИЯ

**RU2019667433**

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ
ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства):
2019667433
Дата регистрации: 24.12.2019
Номер и дата поступления заявки:
2019666505 11.12.2019
Дата публикации и номер бюллетеня:
24.12.2019 Бюл. № 1
Контактные реквизиты:
тел.: (499) 1967100 (доб. 67-82),
morozova_ai@nrcki.ru

Автор(ы):
Московский Антон Дмитриевич (RU)
Правообладатель(и):
Федеральное государственное бюджетное
учреждение «Национальный исследовательский
центр «Курчатовский институт» (RU)

Название программы для ЭВМ:

«Система локализации мобильного робота на основе недоопределенных моделей»

Реферат:

Программа позволяет определить положение мобильного робота в рамках заданной карты по показаниям бортовых сенсоров. Программа позволяет работать со следующим набором сенсоров: одометрия, GNSS-навигация, IMU-модули, системы распознавания объектов-ориентиров. Программа реализована как узел фреймворка ROS и снабжена стандартными интерфейсами ввода-вывода данных через топики, взаимодействием с модулем tf, визуализацией rviz и средствами запуска через launch файлы. Тип ЭВМ: IBM PC-совмест. ПК на базе процессора Intel Core i7-3770; ОС: Ubuntu 16.04, Robotic Operation System Kinetic\Melodic.

Язык программирования: Python 2

Объем программы для ЭВМ: 0,063 Мб