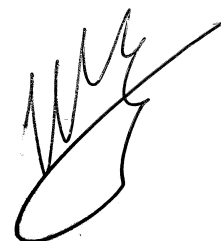


Федеральное государственное бюджетное учреждение науки
Институт проблем передачи информации им. А.А. Харкевича
Российской академии наук (ИППИ РАН)

На правах рукописи



Швец Евгений Александрович

**РАЗРАБОТКА МОДЕЛЕЙ КАРТИРОВАНИЯ И
ПАТРУЛИРОВАНИЯ КОЛЛЕКТИВОМ БЕСПИЛОТНЫХ
НАЗЕМНЫХ РОБОТОВ, ИСПОЛЬЗУЮЩИХ
ТЕХНИЧЕСКОЕ ЗРЕНИЕ И ЭХОЛОКАЦИЮ**

Специальность 05.13.18 —

Математическое моделирование, численные методы и комплексы программ

Диссертация на соискание учёной степени

кандидата технических наук

Научный руководитель:

к.ф.-м.н.

Николаев Дмитрий Петрович

Москва — 2016

Оглавление

	Стр.
Введение	5
Глава 1. Современные алгоритмы исследования местности	10
1.1. Используемые модели роботов и математическая модель получения цифровых данных	10
1.2. Алгоритмы картирования	14
1.2.1. Возникновение и развитие алгоритмов картирования	14
1.2.2. Актуальные проблемы одновременных локального и картирования	21
1.3. Алгоритмы патрулирования	24
1.3.1. Критерии оценки эффективности патрулирования	24
1.3.2. Обзор существующих методов патрулирования	25
1.4. Алгоритмы построения карт проходимости	32
1.4.1. Существующие методы построения карты проходимости	32
1.4.2. Прямая и обратная модели сонаров	34
1.5. Выводы к главе	39
Глава 2. Коллективное продолжительное картирование слабо-динамического окружения	41
2.1. Постановка задачи	41
2.2. Архитектура хранения данных	43
2.2.1. Используемые единицы данных	43
2.2.2. Структура хранилища данных	47
2.2.3. Деление на регионы	49
2.3. Функционирование системы, алгоритмы обработки данных	50
2.3.1. Обработка новой позы	50
2.3.2. Контроль размера ХЛД	56
2.3.3. Построение глобальной карты	59
2.3.4. Алгоритм движения роботов	60

2.4.	Процедура обмена данными	60
2.4.1.	Алгоритм для работы в нормальных условиях	61
2.4.2.	Алгоритм для работы в условиях недостаточной пропускной способности сети	62
2.5.	Выводы к главе	64
Глава 3. Стохастическое патрулирование коллективом роботов с использованием метода потенциалов		66
3.1.	Постановка задачи	66
3.2.	Система моделирования	69
3.2.1.	Варьируемые параметры моделирования	73
3.2.2.	Дополнительные возможности конфигурирования моделирования	74
3.3.	Патрулирование территории без препятствий голономными роботами	74
3.4.	Патрулирование территории с препятствиями голономными роботами	78
3.5.	Патрулирование территории с препятствиями неголономными роботами	83
3.6.	Улучшенные алгоритмы локального выбора направления движения	87
3.7.	Другие дополнения к алгоритму	91
3.8.	Дилатация “Тумана Войны”	94
3.9.	Работа в условиях плохой сети	95
3.10.	Выводы к главе	97
Глава 4. Построение карты проходимости на основе показаний сонаров		99
4.1.	Постановка задачи	99
4.2.	Прямая детерминистическая модель сонара	99
4.3.	Вероятностная прямая модель	104
4.4.	Стохастический метод восстановления карты	106
4.4.1.	Восстановление карты	107
4.4.2.	Численные эксперименты	109
4.5.	Метод градиентного спуска	111
4.5.1.	Восстановление карты	111

4.5.2. Численные эксперименты	113
4.6. Сравнение двух предложенных методов восстановления	117
4.7. Выводы к главе	118
Заключение	119
Список сокращений и условных обозначений	120
Словарь терминов	121
Список литературы	122
Приложение	138

Введение

Актуальность. В настоящее время автономные наземные и летательные аппараты (роботы) широко используются для различных задач, которые традиционно решались с помощью человеческих ресурсов: разведки, построения карты местности, патрулирования, видео- и фото- съемки, доставки или переноса грузов, в том числе в сценариях, где работа человека может представлять опасность (шахты, спасательные операции т.д.).

Для решения описанных выше задач роботы должны эффективно решать множество более простых подзадач: строить карту и определять собственное положение на территории, собирать и обрабатывать информацию с использованием технического зрения и эхолокации и выбирать маршруты, при движении по которым собирается необходимая в данный момент информация. Ранее разработка этих алгоритмов осуществлялась для случая одного робота.

В настоящее время становится популярным использование коллектива роботов для исследования территории, поскольку использование нескольких роботов позволяет повысить скорость исследования. Однако распределенное картирование, организация движения и другие задачи, возникающие при использовании нескольких роботов, исследованы заметно хуже, чем их аналоги для случая одного робота. Также для эффективного коллективного функционирования необходимо предложить алгоритмы, управляющие коллективным движением и регламентирующие обмен и обработку информации коллективом роботов.

Неотъемлемой частью задачи исследования территории является картирование территории. В современной робототехнике задачи картирования и локализации робота тесно связаны и обычно решаются вместе в рамках задачи одновременных локализации и картирования (англ. SLAM). Задача SLAM для коллектива роботов исследована заметно хуже, чем для случая одного робота.

Информация, которую собирает робот, зависит от маршрута его движения, поскольку сенсоры робота имеют ограниченную дальность. Для эффективного использования коллектива роботов необходимы алгоритмы движения, отличные от алгоритмов, используемых при применении одного робота.

Большинство существующих работ рассматривают задачу исследования статической территории – то есть территории, где ориентиры, используемые роботами для локализации, остаются актуальными в течение всего времени исследования. В реальном мире в большинстве сценариев территория является динамической – то есть изменяется так, что за время картирования накапливаются изменения, достаточные для того, чтобы алгоритмы перестали работать корректно. Достаточно легкой задачей считается фильтрация быстро движущихся объектов при построении карты; обнаружить медленно движущийся объект – более сложная задача, и очень мало работ посвящено этой проблеме.

Другой проблемой, слабо освещенной в литературе, является продолжительная работа в условиях ограниченности объема памяти роботов. В случае, если исследование длится достаточно долго, то объем первичных данных, собираемых роботом, превысит любой заданный объем памяти.

Таким образом, задача построения моделей для исследования меняющейся местности коллективом роботов с ограниченным объемом памяти является актуальной.

Степень разработанности темы. В литературе существуют решения проблем картирования коллективом роботов, длительного исследования местности и работы на меняющейся территории. Стоит отметить работы зарубежных ученых С.Труна, Г.Сухатме, Д.Вульфа, М.Фингсторна. Однако эти работы не решают все проблемы согласованно. Проблемы коллективного патрулирования и картирования меняющейся территории тесно связаны, но редко рассматриваются вместе; кроме того, внимание не уделяется обеспечению длительного функционирования системы при ограниченном объеме памяти роботов. Подавляющее большинство алгоритмов патрулирования предназначены для помещений, а не для открытой территории, и не рассматривают возможность изменения территории. Работ, рассматривающих все описанные проблемы в комплексе, найдено не было.

Целью данной работы является создание математических моделей и необходимой алгоритмической базы программного обеспечения для повышения эффективности сбора информации коллективом роботов с ограниченным объемом памяти.

Для достижения поставленной цели были решены следующие **задачи**:

1. Разработана архитектура системы группового исследования территории методами одновременных локализации и картирования (SLAM), пригодная для работы на меняющейся территории в течение неограниченного промежутка времени.
2. Разработана среда имитационного моделирования коллективного исследования территории для испытания алгоритмов патрулирования, оценки и сравнения их эффективности.
3. Построена математическая модель и разработаны алгоритмы нерегулярного патрулирования для коллектива роботов без единого центра управления с запретом на обмен информацией о планируемых действиях с использованием метода потенциалов.
4. Разработана система тестирования и визуализации алгоритмов построения карты проходимости с использованием имитационной модели показаний сонаров.
5. Разработан численный метод построения карты проходимости в реальном времени на основе показаний сонаров и проведено сравнение точности восстанавливаемых карт с традиционными методами построения карты реального времени.

Научная новизна. Разработана модель сжатия, обработки и обмена поступающей с сенсоров информации, которая позволяет использовать существующие методы локализации и картирования для исследования динамической территории коллективом роботов.

Разработанная модель и алгоритмы патрулирования, основанные на методе потенциалов, позволяют обеспечить автономное непредсказуемое движение роботов без обмена планами о будущем движении при решении задачи патрулирования открытой территории.

Предложен метод обработки информации для картографирования, который позволил использовать прямую модель сонара (forward sonar model) для работы в режиме реального времени.

Практическая значимость. Архитектура системы для хранения, обработки и обмена данными позволяет увеличить длительность исследования местности коллективом роботов и повысить эффективность сбора информации. Разработанная среда имитационного моделирования для визуализации, отладки и измерения

численных характеристик алгоритмов патрулирования может быть использована чтобы сэкономить время и средства при разработке и тестировании новых алгоритмов патрулирования.

Комплекс программ для имитационного моделирования использован в ИП-ПИ РАН в рамках работ по проекту РНФ.

Алгоритм построения карты проходимости успешно внедрен в проекте “Стая”. Алгоритм превосходит описанные в литературе аналоги по точности построения в классе алгоритмов реального времени для случая территории с препятствиями малого размера и узкими проходами. Алгоритм позволяет обнаруживать проходы, в 1.3 раза более узкие, чем традиционные алгоритмы, а также существенно увеличить точность определения ширины проходов.

Положения, выносимые на защиту:

1. Предложенная архитектура системы хранения и обмена данными в коллективе роботов обеспечивает поддержание актуальной карты меняющейся местности при ограниченной памяти роботов.
2. Разработанный комплекс программ для имитационного моделирования позволяет исследовать, сравнивать и визуализировать алгоритмы коллективного патрулирования с учетом возможных разрывов связи между роботами.
3. Разработанный алгоритм коллективного стохастического патрулирования пригоден для работы на открытой местности и обеспечивает непредсказуемое поведение роботов при отсутствии центра управления и не требует передачи по внешнему каналу команд и указаний о планируемых действиях роботов.
4. Численный метод построения карты проходимости на основе показаний сонаров способен работать в режиме реального времени, существенно увеличить точность определения ширины проходов и обнаруживать проходы в 1.3 раза более узкие, чем традиционные алгоритмы в ряде сценариев.

Степень достоверности результатов, полученных для задачи патрулирования, обеспечивается многократно повторенным имитационным моделированием. Архитектура обработки, обмена и хранения данных построена на основе анализе практик и обобщении опыта опубликованных работ других авторов. Алгоритмы

модели, отвечающие за ассоциирование данных, были имплементированы и протестированы на реальных данных. Алгоритмы построения карт проходимости на основе показаний сонаров имплементированы и протестированы на реальных и синтетических данных. Полученные результаты находятся в соответствии с результатами, полученными другими авторами.

Апробация работы. Основные результаты работы докладывались и обсуждались на двух международных конференциях: 29th European Conference on Modelling and Simulation (ECMS 2015, Albena, Bulgaria) и Eighth International Conference on Machine Vision (ICMV 2015, Barcelona) и были доложены на научном семинаре Лаборатории 11 Института Проблем Передачи Информации.

Личный вклад. Все результаты диссертации, вынесенные на защиту, получены автором самостоятельно. Автором также самостоятельно проведены вычислительные эксперименты на синтезированных и реальных данных. Постановка задач и обсуждение результатов проводилось совместно с научным руководителем, имплементация алгоритмов картирования на основе показаний сонаров осуществлена под руководством диссертанта стажером-исследователем лаборатории 11 ИППИ РАН Шепелевым Д.А.

Публикации. Основные результаты по теме диссертации изложены в 5 научных публикациях, 4 из которых изданы в рецензируемых журналах, 2 из которых рекомендованы ВАК [1,2], 2 — в трудах конференций, индексируемых Web of Science[3,4].

Объем и структура работы. Диссертация состоит из введения, 4 глав, заключения, библиографии и приложения, содержит 33 рисунка и 6 таблиц. Общий объем диссертации составляет 139 страниц. Библиография включает 143 наименования на 17 страницах.

Глава 1. Современные алгоритмы исследования местности

1.1. Используемые модели роботов и математическая модель получения цифровых данных

В процессе работы автономные роботы должны решать множество задач: определения собственного положения, сбора и интерпретации информации об окружающей среде, выбора направления движения и управление актуаторами для исполнения маневров. Для ориентации в окружающем мире и сбора информации роботы используют различные сенсоры. Данные сенсоров снимаются или преобразуются в цифровой вид, в котором роботы могут их обрабатывать с использованием различных алгоритмов. Наиболее часто наземные роботы используют следующие датчики:

- Цифровые камеры
- Сонары – ультразвуковые датчики измерения расстояния
- Датчики инерциальной навигационной системы: гироскоп, акселерометр, магнетометр и др.
- Датчики угла поворота колес.
- Лазерные датчики измерения расстояния (лидары).

Для тестирования алгоритмов, использованных в данной работе, использовался прототип наземной робототехнической платформы, разработанный в рамках коммерческого проекта “Стая”, целью которого являлась разработка программно-аппаратного комплекса, состоящего из коллектива взаимодействующих роботов, осуществляющих исследование и патрулирование территории.

На рисунке 1.1 приведена фотография используемого прототипа, а также на указано используемое оборудование. Прототип оснащен стереопарой на базе цифровых камер Imaging Source DFK 23U445 с широкоугольными объективами (140°), разрешением 1280x960 и максимальной частотой кадров 30 в секунду, 6-ю сонарами LV-MaxSonar-EZ1, инерциальной навигационной системой на основе чипа CMPS11. Для доступа к сети используется WiFi адаптер D-Link DWA-126.

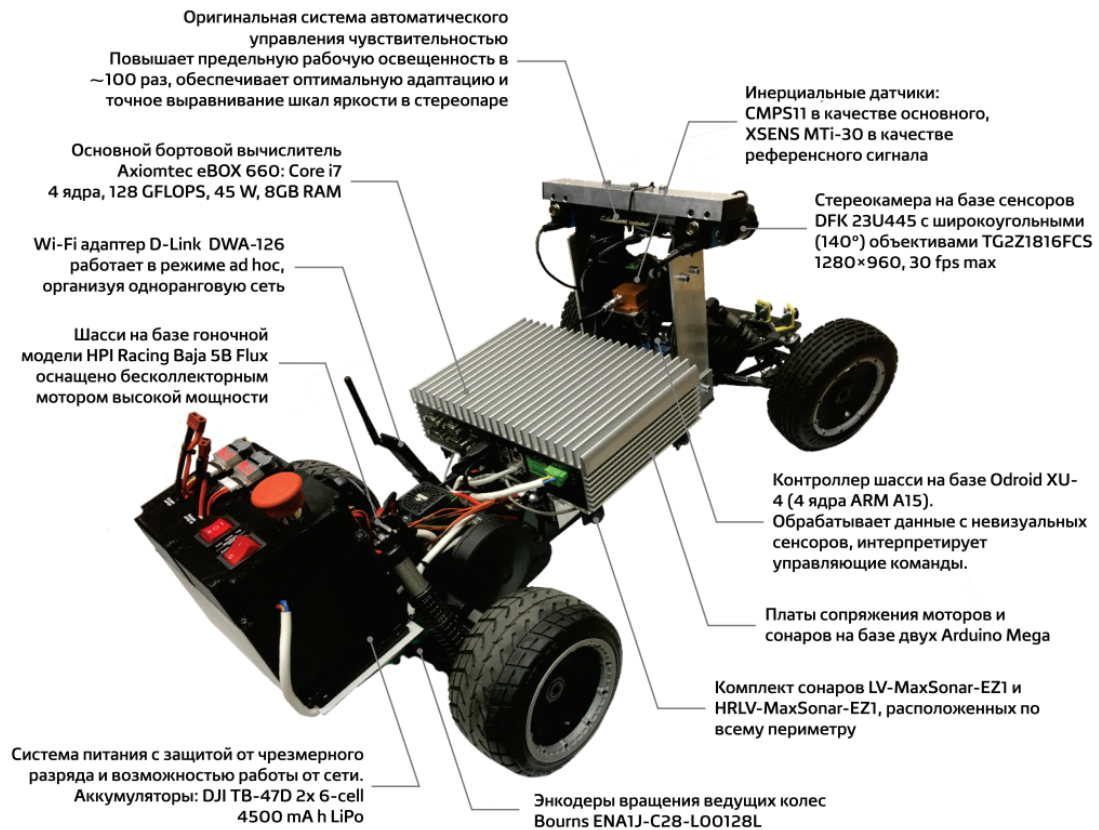


Рисунок 1.1. Оборудование используемого прототипа

Используется четырехъядерный бортовой вычислитель Axiomtec eBOX 660: Core i7.

На многих робототехнических платформах используется набор датчиков из сенсоров, стереопары и инерциальной навигационной системы. Такой набор позволяет получить достаточную для функционирования робота информацию без использования дорогостоящего лазерного дальномера. На рис. 1.2 приведена структурная схема функционирования одного автономного робота, оснащенного стереопарой (предоставляющей данные для алгоритмов технического зрения) и сенсорами (предоставляющими данные для эхолокации) и осуществляющего решение задачу исследования.

Рассмотрим вкратце модели получения цифровых данных с использованием этих датчиков. Схема работы камеры приведена на рисунке 1.3. Используется следующая модель получения цифрового изображения с использованием камеры:

1. Камера осуществляет центральную проекцию окружающего мира на плоскость фотоматрицы.
2. Элементы матрицы измеряют освещенность каждого из пикселей.



Рисунок 1.2. Структурная схема работы робота

3. Полученный на матрице сигнал дискретизируется по пространству и яркости проекции.

Значение освещенности в различное время суток может значительно варьироваться (от 17000 в солнечный день до 30 в сумерках). Кроме того, в процессе движения робот может посещать участки территории, освещенные разным образом. Поэтому необходима система, осуществляющая автоматический контроль экспозиции. Экспозиция равна произведению освещенности (регулируется диафрагмой) и выдержки (времени, в течение которого свет попадает на матрицу при снятии одного кадра). Регулирование диафрагмы является предпочтительным, поскольку позволяет избежать преждевременного выгорания пикселей матрицы камеры.

Принцип работы сонаров заключается в создании ультразвуковой волны, отражающейся от препятствий, располагающихся в луче сонара. Сонар измеряет время между излучением и приемом волны, и вычисляет расстояние до препятствия (считая известным скорость звука в воздухе). Затем происходит оцифровка полученной величины.

Электронные гироскопы, измеряющие ускорение, могут работать на разных принципах. Одним из основных является использование конденсатора с перемен-

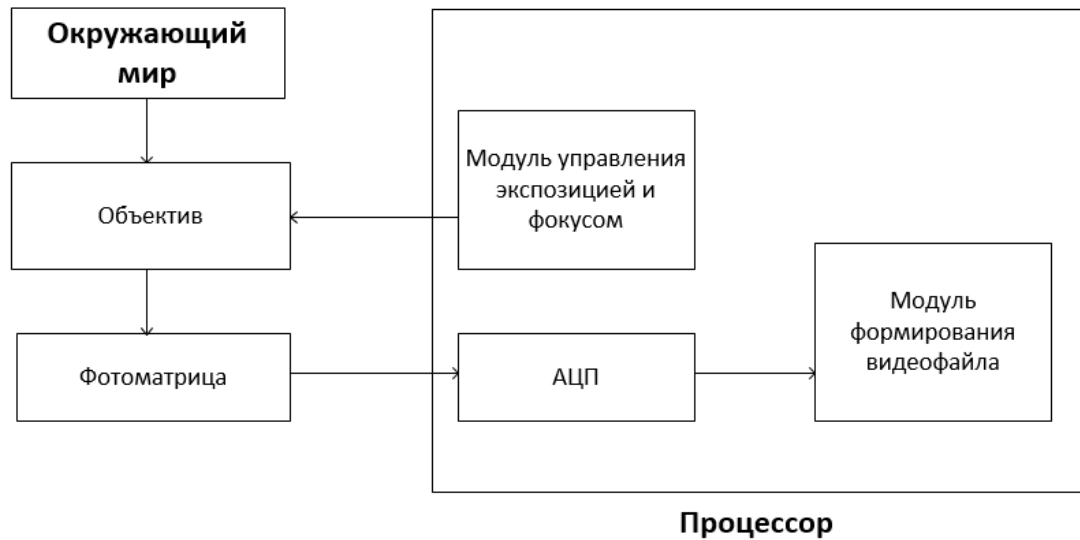


Рисунок 1.3. Схема работы цифровой камеры

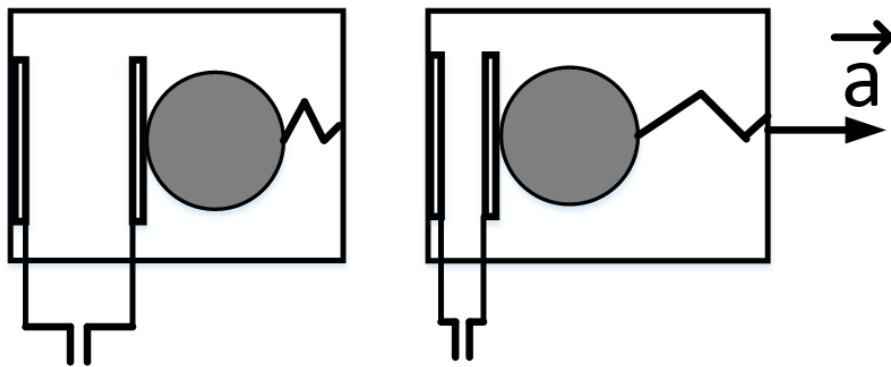


Рисунок 1.4. Схема работы электронного гироскопа

ной емкостью. На рисунке 1.4 приведен пример: при ускорении по оси ОХ груз сжимает обкладки конденсатора, увеличивая его емкость.

Датчики поворота колес считывают количество оборотов, которые совершили колеса робота вокруг оси вращения и позволяют рассчитать расстояние, которое робот проехал (не учитывая проскальзывание). Также они позволяют определять углы поворота колес в других плоскостях.

1.2. Алгоритмы картирования

1.2.1. Возникновение и развитие алгоритмов картирования

Задача картирования является неотъемлемой частью исследования местности. Задача картирования возникла в 1980-ых годах вследствие развития самоходных роботов и является одной из наиболее исследуемых и сложных задач робототехники и на сегодняшний день. Формально, задача картирования заключается в получении пространственной модели окружения с помощью роботов. Популярность задачи обусловлена тем, что успешное картирование обеспечивает работа адекватной моделью окружающего мира, пригодной для навигации и планирования движения. Картирование также является составным элементом для решения более сложных задач, где необходима автономность (например спасательные операции – поиск объектов в неизвестной местности, сбор разведывательной информации) и необходимым шагом на пути к созданию полностью автономных роботов. Полученные с использованием роботов карты могут использоваться людьми (например, картирование и исследование большого участка дикой местности).

Существует два подхода к представлению карт: топологическое и метрическое. Метрические представления описывают геометрические свойства окружения и задают точное положение объектов на территории, топологические – описывают соединенность различных мест территории. Одним из первых геометрических представлений стала сетка проходимости [1] – карта поделена на прямоугольные ячейки, каждая из которых содержит информацию о проходимости территории в соответствующем квадрате. На основе этого подхода было имплементировано много успешных робототехнических систем, например [2–4]. Другим примером геометрического представления карты является описание препятствий с помощью набора многоугольников [5].

Топологические карты представляют собой графы, где вершины соответствуют легко отличимым локациям на карте (например, комнатам в офисе), а ребра графа соединяют вершины между которыми возможно непосредственное передвижение робота [6; 7]. Различие между топологическим и геометрическим

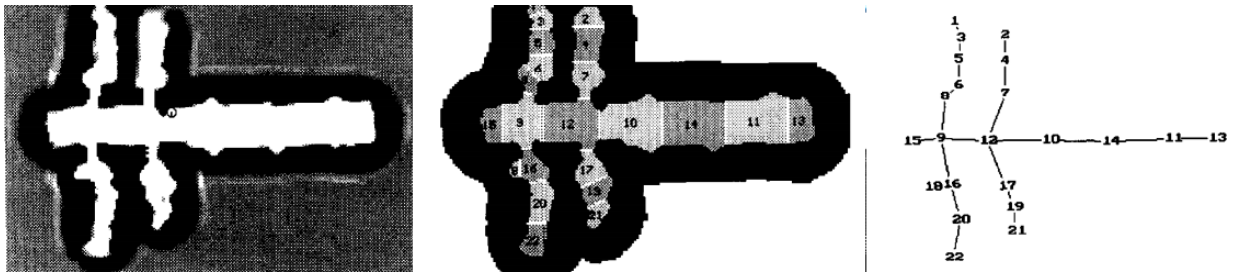


Рисунок 1.5. Пример построения топологической карты [10]

методами не является четко определенным, так как топологические карты зачастую получаются в результате обработки геометрической информации. Пример топологической карты приведен на рисунке 1.5. Топологические карты обладают преимуществом простоты, и использование топологических карт может облегчить решение задачи поиска маршрута. Однако методы, строящие топологические карты, очень плохо работают вне помещений, где территория не представляется в виде графа естественным образом. Кроме того, геометрические карты содержат значительно более точную и детальную информацию об окружении, и потому являются основным способом представления карт сегодня. Существуют методы, использующие смешанные представления [8; 9]. Так, в работе [8] карта строится путем решения задачи максимального правдоподобия. Топологическая карта используется для получения грубой оценки карты и для разрешения конфликтов в случаях, когда использование геометрического представления не позволяет отличить две различные, но похожие локации (например, два одинаковых коридора). Геометрическая информация используется для уточнения карты. В работе [9] предлагается вообще не строить глобальную геометрическую карту территории, и использовать топологическую карту для планирования долгосрочного движения и множество локальных геометрических карт для планирования локального.

Практически все подходы, представленные в литературе сегодня, конструируют карты, которые содержат информацию об окружении, представленную в глобальной системе координат, и объекты на карте вообще не содержат информации об измерениях сенсоров, которые привели к обнаружению этих объектов. Однако существует альтернативное представление окружения – в пространстве измерений [11]. При использовании подобного описания картирующий алгоритм не восстанавливает объекты окружения (стены, двери и другие препятствия) на

основе показаний датчиков, а документирует, какие показания дают сенсоры робота в различных местах территории. Вместо того, чтобы наносить на карту препятствия, существование и позиция которых выведены на основе измерений, на карту наносятся непосредственно измерения робота. Сравнивая текущие показания робота с показаниями, внесенными на карту, робот может оценивать свое положение. Однако подобный алгоритм имеет некоторые проблемы: во-первых, с использованием подобного алгоритма может быть сложно экстраполировать наблюдения в точке, где робот не проезжал, из существующих наблюдений. Во-вторых, подобный подход создает трудности при сравнении разных позиции, в которых показания сенсоров практически идентичны. Первый подход сегодня используется значительно чаще, однако существуют успешные имплементации второго подхода [12; 13].

Наиболее часто используемыми типами карт являются карты проходимости [1; 14], облака точек [15; 16] и карты ориентиров [17; 18]. Карты проходимости чаще всего представляют собой двумерную сетку, каждая ячейка которой описывает определенный участок территории и хранит информацию о проходимости. Существуют более сложные вариации: карты высот [19] хранят не просто информацию о проходимости в данной точке, но и высоту земной поверхности или препятствия в данной точке. Более сложные многоуровневые карты поверхностей [20] и трехмерные сетки проходимости [21] позволяют описывать объемные объекты (например, мосты и арки). Трехмерные облака точек описывают пространство в виде набора точек с известными координатами. Для успешного использования такая карта должна состоять из большого числа особых точек: при использовании лазерных сканеров один кадр может содержать десятки и сотни тысяч точек [22]. Карты ориентиров состоят из набора характерных объектов с известными координатами, хорошо различимых и позволяющих роботу проводить позиционирование относительно этих ориентиров. В качестве ориентиров могут быть использованы, например, найденные особые точки с характеризующими их дескрипторами, например SIFT и SURF [23; 24], иногда группирующиеся в объекты. В качестве ориентиров могут также выступать характерные объекты на карте глубины или объекты отличающиеся по цвету [25; 26]. Подобные карты особенно удобны для применения внутри помещений [16], где существует

большое количество ориентиров, однако могут быть использованы и вне помещений [25].

Способ представления карты зависит от типа сенсоров, которыми укомплектован робот. Поскольку различные сенсоры предоставляют данные различного типа, то и восстановленные карты имеют разную структуру. Наиболее используемыми сенсорами являются видеокамеры, сонары и лазерные дальномеры. Также, для отслеживания собственного передвижения (и, следовательно, более точного восстановления карты), роботы используют инерциальную систему и физические датчики (например, счетчики оборотов на колесных осях).

Видеокамера является достаточно дешевым и пассивным сенсором. С использованием видеокамер обычно строятся карты ориентиров (заметных объектов местности) [27]. С использованием двух (и более) камер и алгоритмов стереозрения возможно более легкое восстановление трехмерных позиций ориентиров [28], а также использование плотных алгоритмов стереосопоставления [29] для построения карт проходимости [30]. Недостатком видеокамер является зависимость от условий освещения. Лазерные дальномеры позволяют восстанавливать высоко детализированную трехмерную картину мира, т.е. плотные облака точек [16; 25], карты проходимости [31] и карты ориентиров. По сравнению с видеокамерами лазерные дальномеры дают значительно более точную информацию. Их значительным недостатком однако является дороговизна, а также достаточно крупные размеры. Кроме того, лазерный дальномер является активным сенсором, и его использование может быть нежелательно в случае, если необходимо скрыть присутствие робота (например, разведывательные операции). Сонары (ультразвуковые датчики измерения расстояния) являются компактными, и способны работать при любом освещении. Также они очень дешевы; однако их недостатком является сильная зашумленность измерений [1]. Из-за достаточно большой ширины луча сонара на основе их показаний проблематично точное восстановление маленьких объектов, размер которых сопоставим с шириной луча сонара. Сонары являются активными датчиками, однако радиус их действия мал и составляет несколько метров.

При локализации на основе показаний инерциальных и колесных датчиков возникает проблема постоянно накапливающейся ошибки: даже если каждое измерение содержит небольшую ошибку, при интегрировании показаний за долгий

промежуток времени происходит неограниченный рост ошибки. Чтобы бороться с этим явлением, обычно используются данные с видеокамер робота. С их помощью обнаруживаются характерные объекты на территории (ориентиры), вероятность обнаружить которые мала в другом месте территории. Совокупность ориентиров позволяет с большой вероятностью однозначно идентифицировать позицию на территории. Когда робот повторно посещает место, богатое ориентирами, он может определить, что находился в данной позиции ранее, и обновить текущую оценку своей позиции. Это позволяет избавиться от ошибки, накопленной интегрированием показаний сенсоров с момента последнего посещения этой позиции. Назовем подобный процесс “закрыванием циклов” (перевод с англ. “loop closing” [20; 32]), а процесс соотнесения данных сенсоров, полученных в разнесенные по времени моменты и определение, что на самом деле эти данные являются результатом наблюдения за одним и тем же объектом “ассоциированием данных” (перевод с англ. “data association” [33; 34]). Проблема ассоциирования данных является ключевой для успешной продолжительной локализации робота.

Обычно для осуществления картирования роботу необходимо осуществлять передвижение по территории, поскольку дальность действия сенсоров ограничена. Поскольку показания всех датчиков подвержены влиянию шума, то при оценке собственного положения робота возникает ошибка. С увеличением этой ошибки становится невозможным точное построение или поддержка карты: не зная собственного положения, робот не может корректно вносить изменения в карту; не зная карты, робот не имеет возможности точно определить свое положение. Кроме того, зашумленные показания сенсоров могут привести к внесению некорректной информации в карту. Поэтому для успешного картирования необходимо одновременно оценивать как карту, так и положение робота на строящейся карте. Кроме того, модель должна учитывать неизбежные шум сенсоров и неопределенность в оценке положения робота. Сегодня все успешные модели, используемые в робототехнике, являются вероятностными [35–39]. Необходимость одновременной оценки карты и положения была высказана в конце 1980-ых – начале 1990ых годов, и были предложены статистические методы, позволяющие оценивать положение робота на строящейся и расширяющейся карте [40; 41]. Сегодня задачи локализации робота по известной карте [42; 43], как и построение точной карты, когда известны точные значения положения робота в каждый мо-

мент времени [44] решены для большинства прикладных сценариев. Однако задача одновременного оценивания карты и собственного положения в ней до сих пор остается актуальной. В англоязычной литературе данная задача носит название SLAM (Simultaneous Localization and Mapping). Решению этой задачи посвящено огромное количество работ. В данной работе мы называем ее задачей ОЛИК (Одновременные Локализация И Картирование).

В работах [11; 18; 45] дается краткое описание современных подходов к решению задачи ОЛИК. Все подходы можно разделить на две категории: подходы, решающие задачу с помощью фильтрации и подходы, производящие глобальную задачу оптимизацию карты по всем полученным измерениям. Методы первой категории используют каждое новое полученное измерение для уточнения уже существующей оценки карты и собственного положения, при этом для более точной оценки положения робота используются различные источники данных, в том числе команды управления. Такие методы используют многочастичные фильтры Рао — Блэквелла [46], фильтры Калмана [47; 48] и другие вариации байесовских фильтров [49; 50]). Методы же второй категории после получения нового измерения производят переоценку карты и собственного положения на основании всех измерении. К таким методам относятся методы [51; 52] на основе ЕМ-алгоритма [53] и методы [54; 55], использующие графовую оптимизацию [56]. Методы второй категории могут давать более точный результат, поскольку зачастую новое измерение позволяет лучше интерпретировать данные, полученные ранее. Использование всех доступных данных на каждом шаге алгоритма позволяет, например, исправить ошибки в локализации, допущенные ранее. Однако такие методы потребляют значительно больше вычислительных ресурсов, и многие из них не способны работать в режиме реального времени, поскольку им необходимо делать несколько итераций по всем собранным данным на каждом шаге алгоритма [11]. Существуют и имплементации, способные работать в реальном времени [57].

Алгоритмы, решающие задачу ОЛИК, традиционно состоят из двух частей. Задачей первой части алгоритма является преобразование показаний сенсоров робота в математическую задачу. Будем называть эту часть “внешним алгоритмом” (от англ. front-end) — поскольку именно эта часть алгоритма является взаимодействует с информацией, поступающей извне от сенсоров. Задачей второй части алгоритма является решение математической задачи на основе данных, получаемых

от внутреннего алгоритма. Назовем эту часть системы “внутренним алгоритмом” (от англ. back-end).

На сегодняшний день одним из наиболее эффективных, популярных и исследуемых методов решения задачи ОЛИК является графовый подход [54;55;58]. Этот метод относится ко второй категории, то есть использует все измерения, полученные роботом, на каждом шаге алгоритма. Рассмотрим его подробнее. Основой графового подхода является объединение данных, получаемых роботом в “позы”. Каждая поза содержит информацию, которую робот собрал в течение некоторого промежутка времени, когда он двигался около какой-то точки территории. Фактически, каждая поза является локальной картой, построенной роботом. Поскольку на коротких промежутках времени при интегрировании показаний датчиков не накапливается большой ошибки, то информация, содержащаяся на таких локальных картах, является достаточно точной. Каждая поза соответствует вершине графа, ребрами же соединены те вершины (позы), для которых существует оценка их взаимного положения, записанная в виде вероятностного распределения с известным матожиданием и матрицей ковариаций. Оценка взаимного положения может быть получена как путем интегрирования показаний инерциальных датчиков, так и путем ассоциирования визуальных данных. Интерпретацией показаний датчиков и созданием такого графа занимается внешний алгоритм графового ОЛИК. Внутренний алгоритм проводит оптимизацию полученного графа для получения оптимальной конфигурации поз. После оптимизация графа известны оценки координат каждой из поз, и на основе множества локальных карт интегрируется единая глобальная карта. Существуют методы, не строящие глобальную карту, и оперирующие с множеством локальных карт [9].

Рассмотрим требования, предъявляемые к внешнему и внутреннему алгоритмам графового ОЛИК. Основой эффективного внешнего алгоритма является построение корректных ассоциирований данных, позволяющих избежать неограниченного нарастания ошибки в оценке положения робота. Проблемой является ошибочное ассоциирование данных, которое может внести серьезные ошибки в восстанавливаемую карту. Целый ряд работ направлен на обеспечение надежности ассоциирования данных [59–62]. В работе [60] предлагается использовать несколько источников информации совместно для нахождения корректных ассоциирований данных. Это снижает количество успешных ассоциирований, но

позволяет снизить процент некорректных. В работе [62] рассматривается случай коллектива роботов, и ассоциирования данных должны подтверждаться другими роботами. В работе [59] процесс ассоциирования данных не всегда осуществляется немедленно после получения измерений, а может быть отложен до получения более надежных измерений.

Даже эффективные внешние алгоритмы могут не обнаруживать некорректные ассоциирования данных, поэтому внутренний алгоритм ОЛИК также должен уметь распознать ложные ассоциирования данных. Преимуществом внутреннего алгоритма в борьбе с некорректными ассоциированиями является доступность информации о других ассоциированиях. Большинство методов позволяют определить некорректные ассоциирования данных анализируя ковариацию между различными ассоциированиями [63; 64], отбрасывая те, которые не согласуются с остальными. Важно отметить, что для задачи, когда окружение роботов может меняться (в работе рассматривается именно такой случай), методы, основанные на нахождении слабо согласованных данных могут работать плохо. Данный эффект описан в работе [65], и в этой же работе приведен способ решения подобной проблемы, основанный на выбрасывании целой группы измерений робота.

1.2.2. Актуальные проблемы одновременных локального и картирования

Существующие методы ОЛИК являются эффективными и способны справиться с вероятностной природой показаний сенсоров и неопределенностью позиции робота. Однако существуют слабо исследованные проблемы и задачи ОЛИК. Рассмотрим эти задачи и существующие решения. Одной из основных потребностей для успешной работы существующих алгоритмов ОЛИК является предположение о неизменности окружающей среды. Роботы могут достаточно легко распознать движущиеся объекты [66; 67] (например, людей в помещениях, машин и пешеходов на улицах) и не использовать их как ориентиры для патрулирования, вместо этого используя неподвижные объекты окружения. Значительно большую проблему представляют объекты, которые остаются непо-

движными долгое время, после чего меняют местоположение или исчезают (например, припаркованные автомобили), или объекты, которые движутся достаточно медленно, чтобы их было проблематично распознать как динамичных объектов. Многие существующие алгоритмы ОЛИК неспособны производить корректную карту территории при наличии медленно движущихся или исчезающих, меняющих положение объектов. Существует несколько подходов к данной проблеме. В статье [65] используется графовый подход к ОЛИК. Для нахождения устаревших ориентиров, в графе поз находятся группы ребер, скоррелированных между собой, но плохо согласующихся с остальным графом поз. Такие ребра часто соответствуют ассоциированиям данных, возникшим на основе движущегося ориентира, и подлежат удалению. Алгоритм обладает двумя недостатками: во-первых, он позволяет обнаружить некорректные ассоциирования данных, но не позволяет явно обнаруживать проводить классификацию между надежными и ненадежными ориентирами. Так, при длительном картировании динамический объект может повторно быть причиной некорректных ассоциирований. Во-вторых, метод не обобщается очевидным образом на картирование с использованием группы роботов на случай ассоциирования данных по ориентирам, полученным разными роботами (каждый робот может проводить подобную операцию независимо от других роботов на основе данных, полученным им собственноручно). В статье [52] используется ЕМ-алгоритм: на Е-шаге которого производится оценка того, какие из измерений относятся к динамическим объектам. Недостатком такого метода является большая вычислительная сложность и невозможность проводить оптимизацию распределенно, что не позволяет использовать его при картировании коллективом роботов. В статье [68] для подавления движущихся объектов показания сенсоров становятся неактуальными со временем, при этом карта состоит из нескольких уровней, скорость затухания для которых отличается.

Картирование коллективом роботов является значительно более эффективным по сравнению с патрулированием одним роботом. Однако использование коллектива роботов порождает целый ряд проблем. Во-первых, усложняется задача корректного ассоциирования данных (полученных разными роботами). Это связано с тем, что ассоциирование данных, полученных разными роботами, с большей вероятностью может быть ошибочным, поскольку роботы не обладают точными оценками движения других роботов. Для более надежного ассоцииро-

вания данных в [62] предлагается верифицировать полученные ассоциирования данных другими роботами.

Другой проблемой является “двойной учет” данных – ситуация, когда в процессе обмена информацией одно измерение передается какому-либо роботу более одного раза, вследствие чего робот вносит это измерение на карту несколько раз. Для многих существующих алгоритмов подобную проблему нельзя подавить простым включением уникального идентификатора для каждого измерения, поскольку роботы интегрируют получаемые данные в локальную карту, после чего обмениваются полученной картой, а не отдельными измерениями – в этом случае каждый робот может получать из сети информацию, которая является смесью измерений, которые он уже получил раньше и свежих измерений. Подходы к решению задачи “двойного учета” включают: использование набора фильтров, отслеживающих источник каждого измерения и обмен необработанными измерениями вместо обработанных карт [69], использование графа сети для расчета корреляции между измерениями [70], использование разреженной модели графа поз [71], строгое отделение карты, построенной роботом от карты, построенной на основе информации от всех других роботов [72].

Во-вторых, для эффективного картирования движение роботов не должно быть независимым – их движение должно быть согласовано, чтобы данные, получаемые роботами, были как можно менее скореллированы, но в карте не оставалось необследованных участков. Для координации движения роботов на неограниченной территории (размеры которой неизвестны) используются алгоритмы исследования территории [73], для поддержания существующей карты на ограниченной территории – алгоритмы патрулирования [74]. Подробный обзор алгоритмов патрулирования приведен в разделе 1.3.

Третья категория проблем связана с возможными помехами, разрывами в работе беспроводной сети, используемой роботами, и в ограничениях пропускного канала. Алгоритмы, используемые коллективом роботов для обмена данными, должны быть устойчивы к подобным проблемам связи. Эффективность распределенных методов ОЛИК зависит от успешности обмена информацией. Вопрос эффективного обмена данными (то есть выбора наиболее важного набора имеющихся данных для передачи) и зависимости эффективности ОЛИК от пропускной способности сети практически не получил освещения в литературе. В двух статьи,

где детально рассмотрен данный вопрос [72; 75], предлагают вести постоянное упрощение графа поз, а также сжимать передаваемые измерения. Предложенный подход является эффективным, но обладает следующим недостатком: параметры сжатия информации являются постоянными. В ситуации, когда пропускной канал является достаточным для передачи полной информации, сжатие данных может привести к снижению качества восстанавливаемой карты. Эффективнее будет метод, который позволяет динамически изменять степень сжатия передаваемой информации в зависимости от текущего состояния канала беспроводной связи.

Следующей редко рассматриваемой задачей является продолжительное картирование территории, то есть поддержание актуальной карты меняющейся территории в течение неограниченно долгого времени. Основным вопросом является хранение и обработка данных: количество измерений, снятых роботами, с течением времени неограниченно увеличивается; поэтому необходимо сжимать и удалять старые и неактуальные данные, а также данные, сильно скоррелированные с другими (содержащие дублированную информацию). Для статичных окружений существуют решения, основанные на максимизации хранимой информации [76]. Существует подход, непосредственно оценивающий качество восстановленной карты при выбрасывании различных групп измерений, и отбрасывающий группы измерений, отсутствие которых наименьшим образом повлияет на восстановленную карту [77]. Для меняющихся окружений доминирующей идеей, описанной в литературе, является отбрасывание наиболее скоррелированных измерений [78; 79].

1.3. Алгоритмы патрулирования

1.3.1. Критерии оценки эффективности патрулирования

Сформулируем критерий эффективности патрулирования. Назовем «туманом войны» $TB(X)$ в точке X территории время, прошедшее с момента последне-

го наблюдения за точкой X территории каким-либо из роботов. Часто для оценки эффективности патрулирования измеряют среднее значение $TB(X)$ по всем точкам карты за время патрулирования [80–82]. Дополнительно к этому критерию могут вводиться дополнительные требования, например, чтобы в каждый момент патрулирования любая точка карты должна быть достижима хотя бы одним из роботов за определенное время [80]. В данной работе предлагается использовать альтернативный, более редко используемый критерий для измерения эффективности патрулирования.

Пусть $MTB(t) = \max_{x \in T} TB(x)$ – максимальное значение тумана войны среди всех точек территории в момент времени t . Тогда в качестве критерия используется среднее значение $MTB(t)$ за время патрулирования. Подобный критерий рассмотрен в статьях [81; 83], причем в работе [81] показано, что в среднем для большинства методов патрулирования существует статистическая зависимость между средними значениями $MTB(t)$ и за время патрулирования: среднее значение $MTB(t)$ составляет от 2 до 3.

1.3.2. Обзор существующих методов патрулирования

Патрулирование является задачей организации движения роботов так, чтобы за каждой точкой территории происходило периодическое наблюдение [84; 85]. Эффективный алгоритм патрулирования может быть использован во многих приложениях робототехники: например, при картировании изменяющейся территории роботом необходимо постоянно посещать все места на карте, чтобы оперативно обнаруживать изменения; для задачи обнаружения объекта роботам необходимо равномерно распределиться по территории и посетить все точки, не оставляя необследованных участков.

Коллектив роботов может справляться с задачей патрулирования эффективнее единичного робота, однако для достижения действительно высокой эффективности движение роботов надо координировать (например, роботы не должны исследовать одну и ту же часть территории, а должны быть распределены по карте достаточно равномерно). Одним из решений является централизованное управле-

ние коллективом роботов [86]. Подобное управление может обеспечивать более эффективное движение и позволяет проводить расчет оптимальных направлений движения в едином центре, который обладает большими вычислительными мощностями, однако оно уязвимо: во-первых, в случае потери связи с центром управления робот не будет обладать алгоритмом действий, во-вторых, система с центральным координатором уязвима к его поломке или взлому.

Часто в литературе производится разделение методов патрулирования на методы, предназначенные для работы в помещениях и вне помещений [87; 88]. Формализуем это разделение. Будем называть территорию закрытой, если она естественным образом разбивается на граф такой, вершины которого соответствуют участкам карты, полностью обозримым одним роботом в случае его нахождения на этой территории, и если территория содержит множество узких проходов, которые робот использует для движения (подобные проходы соответствуют ребрам графа). Примером закрытой территории является, например, офисное помещение, для которого посещение любой точки комнаты достаточно для наблюдения всей комнаты. Комнаты помещения в таком графе соответствуют вершинам, дверные проходы и коридоры – ребрам. В противном случае (когда территория не разбивается естественным образом на граф) будем называть территорию открытой. Большинство помещений представляют собой закрытую территорию, а большинство территорий, расположенных вне помещений – открытую. Однако существуют и противоположные примеры: например, большой зал, размеры которого заметно больше дальности действия сенсоров робота, является примером открытой территории внутри помещения. Узкие улицы являются примером закрытой территории вне помещения (перекрестки являются вершинами графа, а улицы – ребрами).

Организация патрулирования на закрытой территории является значительно более простой задачей по сравнению с закрытой территорией. Причина заключается в том, что патрулирование закрытой территории фактически заключается в движении по графу (и посещая вершину графа, робот проводит наблюдение за всеми точками территории, соответствующими вершине), в то время как на открытой территории допустимые направления движения практически не ограничены. Однако основные идеи, используемые для организации патрулирования, являются схожими.

Одним из основных подходов к организации патрулирования на закрытой территории является нахождение такого маршрута на графе, который проходит через все вершины, нахождение соответствующего маршрута на территории и патрулирование роботами, равномерно распределенными по маршруту [84; 85; 89]. Проблемой данного метода является полная предсказуемость в поведении роботов. Наблюдая за роботами, злоумышленник может определить маршрут их движения и спланировать незамеченное проникновение. Кроме того, алгоритм является неустойчивым к выходу из строя одного (или более) роботов: во-первых, при этом нарушится равномерность распределения роботов по маршруту, во-вторых, если поломка произойдет в узком проходе, то другие роботы вообще не смогут продолжить патрулирование.

Вместо циклического движения роботов по всей территории в работах [90;91] предложен другой способ организовать патрулирование. Для этого строится аналогичный граф территории, после чего граф разбивается на подграфы (соответствующие локальным участкам территории). Роботы распределяются между участками территории, соответствующими подграфам, и каждый патрулирует отведенный ему участок. Подобный метод может быть дополнен и обобщен и на случай открытых областей: область патрулирования также делится на участки, и внутри каждого участка патрулирование производится одним роботом по некоторому алгоритму. Данный метод привлекателен тем, что позволяет полностью избежать ситуаций, когда несколько роботов осуществляют движение по одной и той же части территории (данный сценарий плох тем, что точки территории, патрулируемой двумя роботами, посещаются слишком часто), при его использовании роботы всегда достаточно равномерно распределены по карте. Необходимо также отметить, что согласно исследованию, проведенному в [81], данный алгоритм может быть очень эффективно масштабирован, то есть его эффективность слабо падает с увеличением территории патрулирования и числа патрулирующих роботов. Однако данный метод имеет и недостатки: во-первых, он также довольно предсказуем (роботы не покидают отведенных им участков). Во-вторых, хотя выход из строя одного из роботов не нарушает процесса патрулирования других, при поломке робота отведенная ему территория останется непатрулируемой. Аналогично, при добавлении нового робота ему не будет отведено территории. Поэтому подобный алгоритм должен быть дополнен алгоритмом, который ди-

намически перестраивает разбиение территории на участки при выходе из строя или добавлении нового робота. Подобного алгоритма в рассмотренных работах не предложено, а при отсутствии единого центра управления реализация такого алгоритма может быть достаточно сложной.

Описанные выше методы имеют одну общую черту: при использовании этих методов решение о направлении движения робота в каждый момент времени определено еще до начала патрулирования. Таким образом, роботы в принципе не имеют возможности “среагировать” на то, что происходит на территории патрулирования. Очевидно, что для алгоритм, способный реагировать на различные события (например, на поломку роботов или добавление новых, или на изменение территории патрулирования) должен динамически определять направления движения для каждого из роботов, исходя из текущей ситуации. Рассмотрим алгоритмы, использующие подобный метод.

Простейший алгоритмы, оценивающие обстановку и принимающие решения о направлении движения в реальном времени, используют только информацию о ближайшей видимой территории [80; 83]. Подобные алгоритмы просты в имплементации, и зачастую показывают эффективность патрулирования, сравнимую с более сложными методами [81]. В работе [92] используется машинное обучение с подкреплением [93] для организации патрулирования: в процессе патрулирования роботы изменяют свои алгоритмы поведения для лучшего патрулирования конкретной местности. В работе [94] описан метод, основанный на т.н. “аукционах”: участвующие в патрулировании роботы обмениваются сообщениями, определяют точки, которые необходимо посетить и, в зависимости от взаимного расположения, коллективно распределяют между собой эти точки. Данные подходы сложны и требуют постоянной работы радиосвязи. Кроме того, перехватывая сообщения между роботами, злоумышленник сможет заранее предсказать их намерения.

Существует принципиально иной класс методов управления движением роботов, лишенный перечисленных недостатков, основанный на методе потенциалов. Суть метода потенциалов заключается во введении искусственных сил, действующих на робота со стороны различных объектов на карте: других роботов, точек территории, препятствий, ориентиров, пунктов сбора и т.д. Подобные искусственные силы могут быть как силами притяжения, так и силами отталкивания.

Движение каждого из роботов определяется композицией сил, создаваемых объектами на карте (при этом возможны различные законы суммирования сил и выбора направления движения в зависимости от результирующей силы). Поскольку положение робота и объектов постоянно меняется, то силы, действующие на робота, также будут меняться; следовательно, поведение робота будет зависеть от ситуации на окружающей территории. Динамическое варьирование законов, по которым формируются силы, действующие на робота, позволяет добиваться различных типов поведения.

Метод потенциалов широко распространен в робототехнике, и организация патрулирования не является самой большой областью его приложения. Основное и наиболее изученное из применений метода потенциалов – управление очень большим числом вычислительно слабых роботов (например, самодвижущихся датчиков) [95–99]. Описанные работы предлагают методы, контролирующие взаимное положение роботов при движении и позволяющие создавать различные пространственные конфигурации роботов-сенсоров при развертывании сети. Также в работах рассматривается сценарий совместного движения роботов с объездом препятствий, при сохранении роботами взаимного положения в пространстве. Также метод потенциалов используется для нахождения путей на территории [100; 101] и для решения задачи избегания препятствий [102].

Существуют и работы, использующие метод потенциалов непосредственно для организации патрулирования. В работе [103] рассматриваются задачи окружения объекта по периметру и рассредоточения по территории для большого числа роботов. Важно отметить, что алгоритмы, разработанные в этой работе, применимы для открытых пространств. В работе [104] авторы предлагают эффективный способ патрулирования и “зачистки” закрытых помещений от захватчиков. Для этого они разбивают патрулируемую территорию на участки и представляют их в виде графа, после чего применяют метод потенциалов на графе. В работе [86] задача патрулирования также решается методом введения искусственных потенциалов на графе. Также в этой работе вводится ключевая идея притяжения роботов к точкам карты, которые дольше всего не были посещены. Работа [82] предлагает метод патрулирования, основанный на виртуальном феромоне, испаряющемся с позиций, посещенных роботами, и привлекающий роботов двигаться в сторону

наиболее долго не посещенных позиций; однако этот метод также опирается на графовое представление территории.

К сожалению, результаты рассмотренных работ не могут быть непосредственно обобщены на случай патрулирования открытой территории, поскольку она не может быть легко представлена в виде графа (по крайней мере, с малым числом связей). В работе [81] показано, что методы, использующие представление местности в виде графа, показывают низкую эффективность, если полученный граф обладает большим числом связей. Данная проблема может быть решена за счет использования большого числа роботов (десятки или сотни) и очень плотного покрытия территории [103], однако подобный метод непригоден для использования в сценариях, где количество роботов невелико. Таким образом, задача патрулирования открытой местности слабо освещена в литературе. Однако существует большое количество решенных задач на основе метода потенциалов. Использование и модификация этих решений может позволить достаточно легко разработать алгоритм патрулирования и использоваться для добавления алгоритмов поведения, необходимых в других сценариях.

Организация патрулирования на основе метода потенциалов в простейшем случае может быть описана как “каждый робот движется в направлении самой неисследованной местности”. Для корректной работы роботам необходимо знать только траектории движения других роботов: это позволяет рассчитать, когда посещалась каждая из точек территории. Даже в случае разрыва связи, роботы могут обмениваться собственной траекторией за некоторый отрезок времени. Еще более простой имплементацией является рассылка только текущих координат. При подобном подходе, однако, при разрывах связи, роботы могут обладать не полной информацией. Хотя подобная организация передачи данных и может снизить эффективность патрулирования, она также делает возможным патрулирование при значительной или даже полной потере связи. В любой имплементации, метод потенциалов значительно проще и менее затратен к сетевым ресурсам, по сравнению с методом обучения с подкреплением [93] и метода “аукционов” [94]. Также передача лишь собственных координат по сети делает систему устойчивой ко взлому: роботы не обмениваются планами дальнейшего движения, а положение роботов злоумышленник как правило может узнать и просто наблюдая за системой.

Важным достоинством метода потенциалов является его гибкость и возможность без больших затрат имплементировать другие режимы поведения, а также переключаться между ними. Например, в [103] предлагается использовать стороннего наблюдателя, который будет в ходе работы системы изменять правила, на основе которых идет подсчет сил, действующих на роботов. Подобное варьирование сил делает возможным реализацию самого различного поведения роботов. Так, резко создав силу притяжения между всеми роботами и посторонним объектом и незначительную силу отталкивания между роботами, можно добиться окружения “чужака”. Для равномерного распределения по территории после нейтрализации объекта достаточно ввести силы отталкивания между роботами. Для движения в форме шеренги или клина необходимо применить локальные потенциальные поля, описанные, например, в работе [96]. Большим преимуществом потенциальных полей также является возможность быстро реализовать необходимое поведение роботов, всего лишь настроив систему сил в системе согласно описанной в литературе.

Для осуществления патрулирования роботам необходимо постоянно двигаться по территории, избегать препятствий и столкновений с другими роботами. Часто задача нахождения пути разлагается на две задачи – выбора цели для долгосрочного движения и краткосрочное планирование движения [105; 106]. В данной работе нас в основном интересует долгосрочное движение робота, однако без алгоритма, планирующего локальное движение робота, работа системы невозможна: робот будет постоянно сталкиваться с различными объектами. В литературе для планирования локального движения часто используется подход нечеткой логики [102; 107]. Существуют алгоритмы, производящие поиск полного пути [108; 109] (а не выбирающие лишь ближайшие несколько шагов), однако разработка и имплементация подобных алгоритмов достаточно сложны, потому что многие роботы являются неголономными (то есть при их управлении контролируется меньшее число степеней свободы, чем он обладает – например, колесный робот с минимальным радиусом поворота), и обладают достаточно сложной моделью движения, поэтому не каждый алгоритм будет применим для различных моделей роботов. Для некоторых типов машин задача нахождения пути решена аналитически, например, для машины, способной ехать вперед и назад и обладающей минимальным радиусом поворота [110] и для машины с фиксированной

скорость [111]. Также существуют алгоритмы, осуществляющие поиск пути на основе метода потенциалов [102]. В данной работе мы рассматриваем методы нечеткой логики и метода потенциалов, при этом мы стремимся разработать такую систему потенциальных сил, которая одновременно решила бы задачи патрулирования и локального движения.

1.4. Алгоритмы построения карт проходимости

1.4.1. Существующие методы построения карты проходимости

Карта проходимости описывает позиции и форму препятствий и проходимость территории. Карты проходимости используются для навигации самим роботом, а также могут представлять интерес сами по себе (например, построение карты проходимости неизвестной территории роботами может помочь найти оптимальный маршрут для экспедиции). Карты проходимости могут строиться на основе показаний различных датчиков: стереопары (с использованием плотных алгоритмов стереосопоставления) [29;30], лазерного дальномера [112;113], радара [114] или сонаров [1; 14]. Сонары являются самыми дешевыми датчиками из перечисленных, а также способны работать в любых погодных условиях. В работе мы рассматриваем алгоритмы построения карты проходимости именно на основе данных сонаров.

Существует несколько типов представления карты проходимости. Самым распространенным и широко используемым является представление карты в виде сетки проходимости. При использовании этого метода каждая ячейка сетки соответствует участку территории. Каждая ячейка содержит информацию о проходимости соответствующего участка территории (зачастую выраженная в виде значения уверенности в оценке, заключенного от 0 до 1). Подобное представление карт проходимости было предложено в конце 1980-ых годов [1; 44] и популярно до сих пор благодаря своей простоте. Недостатком таких карт является строгая дискретизация пространства, вызывающая ошибки при отнесении точек к той или

иной клетке, а также отсутствие информации о нормалях поверхностей. Подобными недостатками не обладают методы, представляющие препятствия в виде геометрических фигур [115], в том числе на основе нахождения прямых границ между занятыми и незанятыми областями [116] с использованием преобразования Хафа [117]. Подобный подход имеет два существенных недостатка: он находит только границы между занятыми и незанятыми областями, и кроме того, он не способен дать оценку уверенности в полученной карте (т.е. вероятностное распределение в пространстве карт проходимости). В работе [118] предлагается подход, позволяющий объединить достоинства обоих подходов, однако подход является вычислительно сложным и не может быть реализован в реальном времени (эффективный алгоритм построения карты проходимости должен выполняться в режиме реального времени для осуществления помощи системе навигации робота). Такой метод однако может использоваться для пост-обработки для построение максимально точной карты территории.

Существует множество методов для построения карт проходимости на основе данных, полученных с ультразвуковых датчиков измерения расстояния (сонаров). Построение подобных карт не является простой задачей, поскольку измерения сонаров подвержены случайным помехам и выбросовому шуму; также большая ширина луча сонара затрудняет восстановление карты: одно и то же показание сонара может быть получено при различных конфигурациях окружающих препятствий (см. рис. 1.6), то есть показание сонара свидетельствует о наличии препятствия не в некотором определенном месте, а лишь о наличии препятствия в некоторой, достаточно большой, области. Кроме того, существуют поверхности, рассеивающие сонарные лучи, и поверхности, которые отражают лучи сонаров только под углами, близкими к прямому, что также затрудняет процесс построения карт. Для эффективного использования сонаров необходимы алгоритмы, способные компенсировать эти недостатки.

Введем необходимые обозначения. Пусть M - карта территории, которую необходимо оценить; $Z_n = \{z_1, z_2, \dots, z_n\}$ - последовательность измерений, полученная сонарами робота к некоторому моменту времени. Помимо показания сонара каждое измерение содержит его позицию в момент измерения (согласно предположению о решенной задаче определения положений робота). Тогда задачей построения карты проходимости является нахождение наиболее вероятной

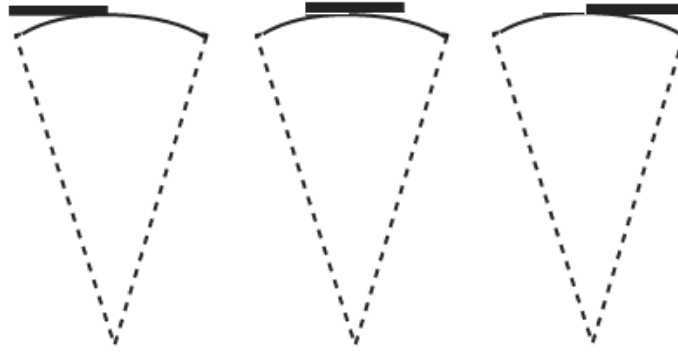


Рисунок 1.6. Пример различных конфигураций препятствий, приводящих к одинаковому показанию сонара

карты M и (опционально) вероятностного распределения $p(M|z_1, z_2, \dots, z_n)$ в пространстве карт проходимости. Обозначим $M_{x,y}$ реальную проходимость участка территории, соответствующего ячейке карты M с координатами x, y ($M_{x,y}$ может принимать два значения: 0, если ячейка карты свободна, и 1 в ином случае), а вероятность (или значение уверенности алгоритма в оценке), что эта ячейка занята обозначим $m_{x,y} = p(M_{x,y} = 1)$.

Для алгоритмов, которые производят итеративное оценивание карты после получения каждого измерения, будем обозначать M^t , m , $M_{x,y}^t$ и $m_{x,y}^t$ текущие оценки карты и значений проходимости ячеек после получения t измерений. Обратим внимание, что некоторые методы при построении карты хранят постоянно варьирующиеся от 0 до 1 значения $m_{x,y}$, и после завершения работы осуществляют бинаризацию полученной карты. Другие методы осуществляют поиск в пространстве бинаризованных карт. Для таких методов будем обозначать $m_{x,y}$ их оценку уверенности в значении проходимости клетки $M_{x,y}$.

1.4.2. Прямая и обратная модели сонаров

Большинство методов, строящих карту проходимости, используют предположение, что задача позиционирования робота уже решена, то есть считают все позиции сонаров, с которых произведены измерения, известными. Рассмотрим простейший подход к построению карты проходимости на основе показаний сонаров [1; 44]. Подобный подход опирается на предположение о независимости

значений вероятности ячеек. То есть распределение карт может быть найдено как произведение распределений отдельных ячеек:

$$p(M|z_1, z_2, \dots, z_n) = \prod_{x,y} p(M_{x,y}|z_1, z_2, \dots, z_n),$$

и задача нахождения карты M большой размерности переходит во множество простых оценок отдельных ячеек $M_{x,y}$.

Для удобства вычисления значения вероятностей клеток записываются в виде логарифмов вероятностей:

$$l_{x,y} = \frac{p(m_{x,y}|Z_n)}{1 - p(m_{x,y}|Z_n)}.$$

Хранение подобного логарифма позволяет однозначно восстановить значение $m_{x,y}$:

$$p(m_{x,y}|Z_n) = \frac{e^{l_{x,y}}}{1 + e^{l_{x,y}}}$$

Простейший итеративный способ построения карты (оценка после получения n -ого измерения при наличии оценки после $n - 1$ измерений) может быть записан в виде [44; 119; 120]:

$$l_{x,y}^t = \log \frac{p(m_{x,y}|z_t)}{1 - p(m_{x,y}|z_t)} + \log \frac{1 - p(m_{x,y})}{p(m_{x,y})} + l_{x,y}^{t-1},$$

где $p(m_{x,y})$ – априорная оценка ячейки карты проходимости, и $l_{x,y}^0 = \log \frac{p(m_{x,y})}{p(1-m_{x,y})}$.

Описанная выше модель опирается на распределение $p(r|z)$ реального расстояния r до препятствия в зависимости от показания сонара z . Такая модель называется обратной, поскольку она обратна процессу снятия показаний – расстояние до объекта в такой модели определяется показанием сонара, а на самом деле показание сонара определяется расстоянием до препятствия. Фактически, алгоритм картирования с использованием обратной модели можно описать следующим образом: “после получения каждого измерения уменьшить вероятность наличия препятствия в тех клетках, которые находятся в зоне видимости сонара и расстояние до которых меньше, чем показанная сонаром дальность. Затем увеличить вероятность наличия препятствия в клетках, которые находятся в зоне

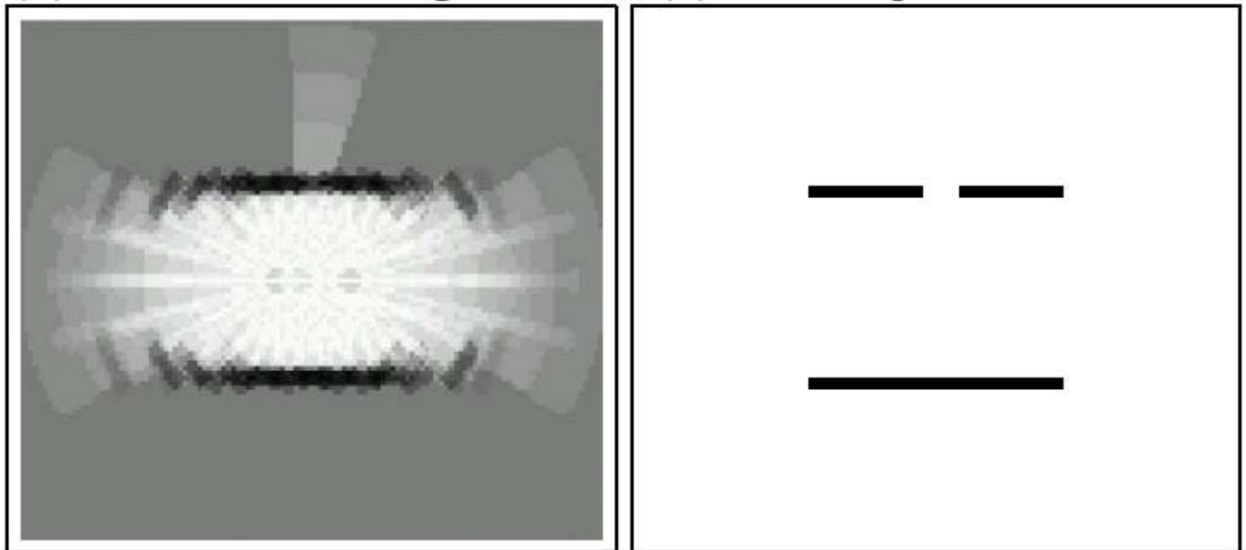


Рисунок 1.7. Пример восстановления карты традиционным методом

видимости сонара, и расстояние до которых примерно равно показанной сонаром дальности”.

Методы, использующие обратную модель сонара, не способны аккуратно моделировать ни вероятностную природу показаний сенсоров, ни возможность получения отражения когда препятствие закрывает луч сонара лишь частично. Предположение о независимости значений проходимости ячеек в описанной выше модели значительно упрощает решение задачи, однако оно может привести к построению сильно искаженных карт проходимости в некоторых сценариях. Обычно проблематичным является восстановление проходов, размер которых сравним с шириной луча сонара, например, дверных проемов. Маленькое препятствие, напротив, может быть увеличено в размере при использовании такого алгоритма восстановления карты. Примеры некорректного восстановления подробно рассматриваются в работе [119], один из них приведен на рисунке 1.7: дверной проем в результате работы традиционного алгоритма, основанного на предположении о независимости ячеек, оказывается полностью “закрашен”. Однако несмотря на указанные недостатки, рассмотренный неточный алгоритм построения карты проходимости остается актуальным и на сегодняшний день в силу простоты и легкости имплементации, и главное – возможности работать в режиме реального времени.

В работах [14; 121] рассматривается проблема переотражения лучей сонаров и способы борьбы с данным явлением. Подобные методы позволяют повы-

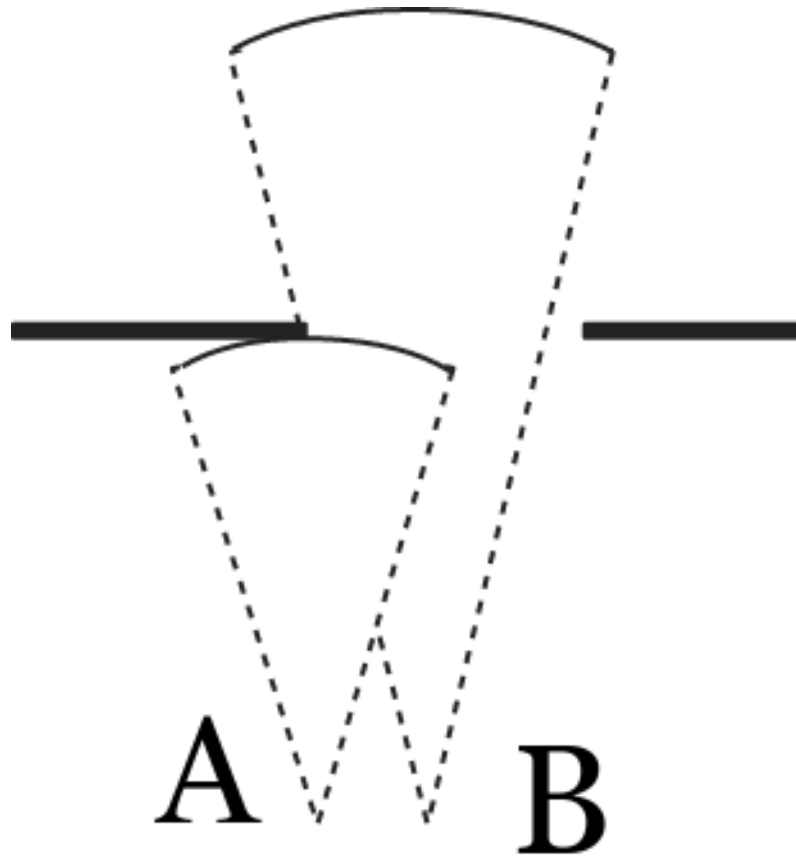


Рисунок 1.8. Пример сценария, где большое количество скоррелированных измерений приведет к неточной карте

ситель качество восстанавливаемой карты в некоторых случаях, однако не решают основные проблемы восстановления с использованием обратной модели.

Из описанных в литературе методов, основанных на обратной модели, стоит отметить метод [14]. В методе учитывается возможность переотражения лучей сонаров при снятии показаний, а также решается проблема большого количества скоррелированных измерений. Для иллюстрации этой проблемы рассмотрим рисунок 1.8. Пусть робот стоит в позиции А достаточно долго и регистрирует много (сотни) измерений; затем в процессе движения получает несколько измерений из позиции В. Поскольку измерений в позиции А значительно больше, то дверной проем на карте будет “закрашен”. Для снижения значимости множества одинаковых измерений в работе [14] предлагается снижать вес измерений, сильно скоррелированных с уже существующими (то есть снятых с той же позиции).

Значительно более точные карты можно восстанавливать с использованием прямой модели сонара. Прямая модель сонара записывается в виде распределения $p(r|M)$ – вероятности получить показание сонара r при условии окружающей карты M . Однако использование подобной модели не позволяет разделить задачу

оценки карты всей территории M на независимые оценки отдельных ячеек. Для оценки правдоподобности одного измерения сонара надо знать значения занятости во всех ячейках, которые находятся внутри луча сонара. Принцип восстановления карты с использованием прямой модели сонара может быть сформулирован следующим образом: “Перебор и поиск в пространстве карт проходимости такой карты, которая наиболее корректно описывает имеющиеся измерения сонаров”. При подобном подходе сохраняются все зависимости между ячейками, поскольку задача решается в начальном пространстве карт большой размерности. Например, в работе [1] обновление карты происходит итеративно, с пересчетом на каждом шаге, задаваемым выражением:

$$p(M_{x,y} = 1|Z_n) = \frac{p(z_n|M_{x,y}=1)p(M_{x,y}=1|Z_{n-1})}{p(z_n|M_{x,y}=1)p(M_{x,y}=1|Z_{n-1})+p(z_n|M_{x,y}=0)p(M_{x,y}=0|Z_{n-1})},$$

где $p(z_n|M_{x,y} = 1)$ и $p(z_n|M_{x,y} = 0)$ – вероятности получить измерение z_n при известном положении сонара, если ячейка карты $M_{x,y}$ является занятой или свободной соответственно. Эта оценка не может быть получена напрямую из модели сенсора, поскольку значения проходимости других клеток (например, расположенных между сонаром и ячейкой $M_{x,y}$) влияют на вероятность получения измерения z_n . Фактически, для оценки этой вероятности необходимо применение теоремы Колмогорова [122] и суммирование по всем возможным конфигурациям карты проходимости:

$$p(z_n|M_{x,y} = 1) = \sum_{\{G_{M_{x,y}=1}\}} p(z_n|M_{x,y} = 1, G_{M_{x,y}=1}) \cdot p(G_{M_{x,y}=1}|M_{x,y} = 1), \quad (1.1)$$

где $G_{M_{x,y}=1}$ – некоторая конкретная карта проходимости с занятой ячейкой $M_{x,y}$, а $\{G_{M_{x,y}=1}\}$ – набор всех таких карт. Вероятность $p(G_{M_{x,y}=1}|M_{x,y} = 1)$ может быть оценена на основе априорных вероятностей $m_{x,y}$ или априорной информации о структуре карты (например, об отсутствии препятствий большого размера); или же все карты $\{G_{M_{x,y}=1}\}$ могут считаться равновероятными.

Авторы [1] предлагают не перебирать все возможные конфигурации карт проходимости для расчета (1.1), а воспользоваться тем, что показание сонара зависит только от ячеек карты проходимости, которые попадают в луч сонара. Однако при таком подходе не ведется оптимизация карты на основе всех показаний сонаров: изменив часть карты так, чтобы она лучшим образом удовлетворяла показанию одного из сонаров, можно ухудшить карту с точки зрения другого

показания сонара (который частично пересекается с рассматриваемым). Поэтому для получения максимально точного решения надо вести оптимизацию по всей карте.

В работе [119] предлагается подобный метод, который перебирает бинаризованные карты (ячейки которых содержат только значения 0 и 1). Для этого используется прямая модель сонара описанная в разделе 4.3 – аналогичная модель используется в одном из предложенных в данной работе методов восстановления. В этой модели учитывается возможность прохождения/рассеяния луча сонара от препятствия: в случае наличия нескольких препятствий, каждое из них может отразить луч с некоторой вероятностью; также возможен случайный выброс, дающий случайное значение величины. Алгоритм использует метод ЕМ-оптимизации (от английского “Expectation - Maximization” – “оценка – максимизация”) [53]. На Е-шаге алгоритма строятся гипотезы о том, какому именно из препятствий, которые существуют на текущей итерации карты (или, возможно, выбросовому шуму) соответствует каждое из показаний сонара. На этапе максимизации строится новая карта на основе найденных сопоставлений. Данный алгоритм показывает высокую точность восстановления и способен восстанавливать карты на основе шумных измерений значительно точнее, чем традиционные методы. Кроме наиболее вероятной бинаризованной карты M , он дает оценку уверенности в значениях проходимости каждой из ячеек $m_{x,y}$, хотя данная уверенность считается для каждой клетки независимо. Важнейшим недостатком рассмотренного метода является неспособность работы в режиме реального времени.

1.5. Выводы к главе

В данной главе проведен обзор литературы по тематике диссертации: (i) описано развитие алгоритмов картирования местности с 1980-ых годов и современное состояние дел, описана ключевая задача одновременных локализации и картирования (англ. SLAM), (ii) описаны существующие алгоритмы патрулирования и область их применимости, приведены используемые критерии эффектив-

ности патрулирования, (iii) проведен обзор существующих методов построения карты проходимости на основе показаний сонаров, описаны достоинства и недостатки этих методов.

Глава 2. Коллективное продолжительное картирование слабо-динамического окружения

Основное содержание данной главы изложено в работе [123].

2.1. Постановка задачи

Задача продолжительного картирования меняющейся местности коллективом роботов включает в себя множество более простых задач, таких как слияние карт, полученных различными роботами, организация эффективного обмена данными между роботами, выбора наиболее релевантной информации среди собранной, выбора ориентиров, по которым роботы будут ориентироваться и вычислять взаимные положения, а также координация их движения. Некоторые из этих задач принято рассматривать вместе, например, одновременные локализация и картирование (ОЛИК). Однако полный набор задач редко рассматривается как единое целое, несмотря на то, что задачи связаны между собой, и решения для каждой из задач не всегда могут быть объединены в эффективно решение для полной проблемы. Мы считаем, что роботы не используют для локализации навигационное поле (GPS, ГЛОНАСС).

Например, для функционирования некоторых алгоритмов ОЛИК необходимо, чтобы роботы обменивались достаточно большим количеством информации [72], однако пропускная способность сети в некоторых случаях может не позволить передавать необходимое количество информации. Напротив, некоторые алгоритмы предполагают отсутствие эффективной сети [124], и минимизируют количество информации, передаваемой между роботами. Такой алгоритм будет уступать алгоритму, использующему весь доступный ресурс сети. Если заранее неизвестно, в каких условиях будет вестись картирование (или пропускная способность сети может меняться со временем), то оптимальным для использования будет такой алгоритм ОЛИК, который способен подстраиваться под текущие условия. Решения такой задачи в литературе найдено не было: построение algo-



Рисунок 2.1. Структурная схема функционирования робота для решения задачи коллективного исследования

ритма ОЛИК редко рассматривается совместно с задачей оптимизации передаваемых данных. Другая возникающая проблема заключается в том, что все роботы обладают конечным объемом памяти, и для организации длительного исследования необходимо удалять старые данные чтобы освободить место для новых. Это решение не может быть принято в отрыве от алгоритма, который используется для объединения данных между различными роботами (потому что на решение, стоит ли выкидывать данное измерение, влияет оценка того, насколько это измерение может полезно в будущем другим роботам).

В следующем разделе мы описываем, какие данные необходимы роботам для решения задачи, и как организовано их хранение. В разделе 2.3 мы описываем алгоритмы, используемые для обработки, сжатия и обмена этими данными.

На рис. 2.1 приведена структурная схема функционирования робота, использующего разработанную архитектуру хранения и обработки данных.

2.2. Архитектура хранения данных

При движении роботы собирают информацию из окружающего мира с помощью различных сенсоров. Используемые в данной работе сенсоры описаны в разделе 1.1. Рассмотрим, как эти данные структурируются и хранятся каждым из роботов.

2.2.1. Используемые единицы данных

Поза

Позы содержат информацию об окружении около определенной точки территории. Эта информация включает в себя локальные ориентиры и информацию о проходимости ближайшей местности, полученную методами технического зрения [125] и эхолокации (как описано в главе 4). Для описания положения ориентиров внутри позы вводится локальная система координат, точка отсчета которой называется центром позы. Обычно центр позы соответствует позиции робота в момент времени, в который были сняты измерения, объединенные в данную позу. Однако иногда происходит объединение двух поз (см. 2.3.2.), тогда у новой позы центр и позиции ориентиров пересчитываются. Поэтому в общем случае центр позы не соответствует позиции робота в какой-то момент времени.

Структура позы:

- Уникальный идентификатор (УИД) позы.
- Внутренняя система координат и оценка позиции позы в глобальной системе координат.
- Набор ориентиров с их координатами во внутренней системе координат.
- Карта проходимости ближней территории.
- Идентификатор региона (ИР) – оценка того, в какой части территории расположена поза, см. 2.2.3.

Ориентиры

Ориентиры – это характерные объекты или точки на территории. Каждая поза содержит информацию об ориентирах, которые располагаются вблизи этой позы. Чтобы оценить взаимное расположение поз роботов, сравниваются позиции ориентиров, которые содержатся в этих позах (то есть происходит ассоциирование данных). Трехмерные позиции ориентиров обычно восстанавливаются на основе алгоритмов стереосопоставления с данных двух видеокамер. В качестве ориентиров мы используем ключевые точки, найденные на каждом из изображений (снятых левой и правой камерами). Далее используются дескрипторы для сопоставления найденных точек на изображениях, и с помощью эппиплярной геометрии восстанавливается трехмерная позиция ориентира. При этом предполагается, что стереопара, использованная для работы, ректифицирована и линия, соединяющая камеры, расположена параллельно земле, поэтому при сопоставлении точек координаты точек на изображении y должны быть близки, а трехмерная позиция (X, Y, Z) точек может быть найдена с помощью:

$$\begin{aligned} Y &= \frac{d_x}{x_r - x_l} \\ X &= \frac{Y(x_l - c_x)}{f_x} = \frac{d_x(x_l - c_x)}{f_x(x_r - x_l)} \\ Z &= \frac{Y(c_y - y_l)}{f_y} = \frac{d_x(c_y - y_l)}{f_y(x_r - x_l)} \end{aligned} \quad (2.1)$$

Здесь $d_x = -f_x B$, где B – смещение правой камеры по оси X относительно левой (центр системы координат (X, Y, Z) располагается на левой камере, как показано на рисунке 2.2). Для простоты мы считаем, что значения фокусных расстояний (f_x, f_y) и оптических центров (c_x, c_y) обеих камер равны. Более подробно почитать про восстановление трехмерной позиции точек на основе изображений с нескольких камер можно в книге [126].

В качестве ключевых точек мы используем SIFT, однако могут быть использованы и другие: ORB, SURF, углы и прямые, см. [24]. К каждому ориентиру привязаны параметры “временная метка” t и “надежность” r , описанные ниже.

Структура ориентира:

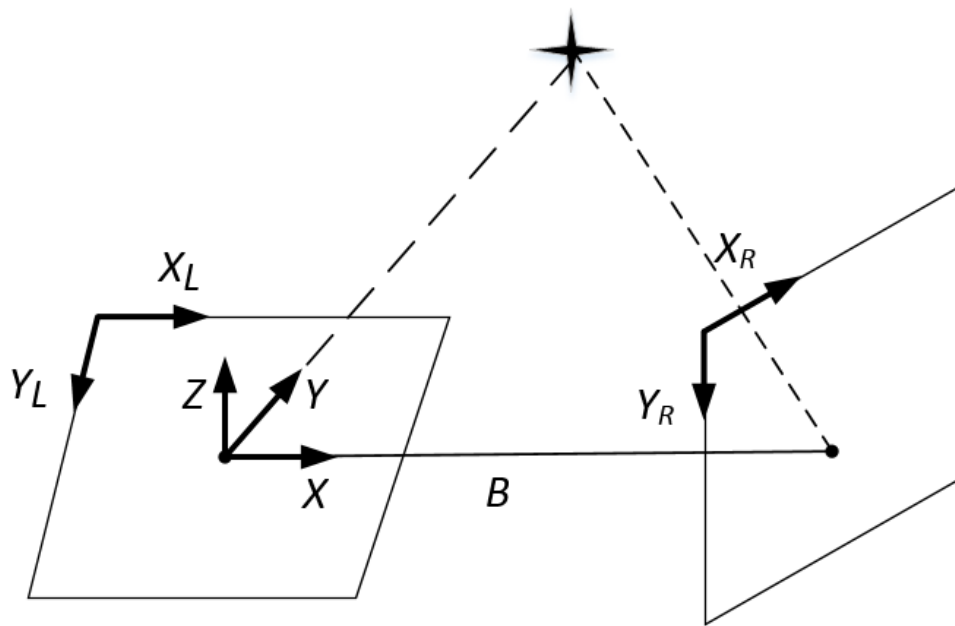


Рисунок 2.2. Иллюстрация работы стереопары

- Дескриптор – позволяет различать различные ориентиры. Как правило, ориентир получается как восстановленная трехмерная точка на основе двух найденных ключевых точек с кадров левой и правой камеры. Дескриптор ориентира считается как среднее между дескрипторами точек на правом и левом кадре.
- Надежность r – оценка того, насколько информация о трехмерной позиции ориентира и его дескриптора верна и актуальна. Надежность принимает значения от 0 до 1, а его начальное значение может быть как выставлено 0.5 по умолчанию, либо получено от алгоритма, сопоставляющего ключевые точки.
- Временная метка – оценка того, когда этот ориентир наблюдался каким-либо из роботов в последний раз. Надежность и временная метка могут быть изменены без изменения УИД позы, содержащей ориентир.

Замыкающие ребра

При помощи сопоставления ориентиров, возможно нахождение взаимного положения поз. Простейший случай ассоциирования данных следующий: робот

ездит по петле, причем по мере движения он накапливает ошибку в оценке собственного положения. Когда робот пересекает свою траекторию, он может по визуальным ориентирам обнаружить, что он уже находился в этой точке, рассчитать относительное смещение и уничтожить накопившуюся за время движения ошибку.

Для нахождения ассоциирований данных используется алгоритм RANSAC, описанный в разделе 2.3.1. Успешные ассоциирования данных хранятся в замыкающих (визуальных) ребрах.

Структура замыкающего ребра:

- УИД поз, соединенных ребром.
- Вероятностная оценка пространственного преобразования между позами (сдвиг и поворот)

Ребра невизуальной одометрии

Ребра одометрии содержат информацию о взаимном расположении двух поз. В отличие от визуального ребра, для ребер одометрии распределение расстояния между позами оценивается на основе информации от колесных датчиков, гироскопов, акселерометров и других датчиков, не используется ассоциирование данных. На длительных промежутках времени подобная информация не может быть просто проинтегрирована для получения позиции робота, потому что ошибка в измерениях постоянно накапливается и неограниченно растет. Именно для того, чтобы ограничить рост ошибки в оценке позиции, используются визуальные датчики. Однако на коротких промежутках времени невизуальная одометрия является достаточно точной и, что важно, в отличие от визуальной одометрии, имеет ограниченную ошибку (грубый пример: если робот с максимальной скоростью 5 метров в секунду достиг позиции B из позиции A за 5 секунд, то расстояние между позициями A и B не может превышать 25 метров). В данной работе ребра одометрии используются для (i) оценки расстояния между позами, когда нет релевантной визуальной одометрии, а также (ii) чтобы отбрасывать некорректные

визуальные ассоциирования данных, которые противоречат физической одометрии.

Структура ребра одометрии:

- УИД поз, соединенных ребром.
- Оценка пространственного распределения между позами (сдвиг и поворот).
- Надежные ограничения на преобразование между позами (такие ограничения, которые наверняка выполняются).

2.2.2. Структура хранилища данных

Хорошо известной проблемой коллективного картирования является проблема двойного учета, заключающаяся в том, что роботы могут учитывать одно и то же измерение несколько раз при создании карт. Рассмотрим сценарий, иллюстрирующий проблему. Пусть имеются три робота А, В и С. Робот А отправляет часть своих измерений роботу В. Робот В обновляет оценку карты и отправляет информацию роботу С, который также обновляет свою карту. При этом робот С использует измерения, изначально полученные роботом А. Затем робот С отправляет свою оценку роботу А. Эта оценка содержит измерения, которые изначально были сняты роботом А. Поэтому если робот А обновит оценку карты с использованием информации, полученной от робота С, то измерения, изначально снятые роботом А будут учтены два раза, что приведет к некорректному увеличению надежности соответствующих элементов карты.

Для борьбы с этой проблемой предлагается разделять информацию, полученную самим роботом, от информации, полученной другими роботами. Каждый робот хранит дату в хранилище локальных данных (ХЛД), хранилище полученных данных (ХПД) и хранилище замыкающих ребер (ХЗР). Каждый робот хранит полную карту окружения, которая собирается на основе всех имеющихся данных и постоянно обновляется в процессе картирования, а также априорную карту, которая остается неизменной в процессе картирования. Априорная карта одинакова для всех роботов.

Все измерения, снятые роботом самостоятельно, хранятся в ХЛД и представлены в виде графа: позы являются вершинами, а ребра одометрии – ребрами. Все измерения, снятые другими роботами и полученные по сети, хранятся в ХПД. ХПД разбито на слои, каждый слой содержит данные, которые были изначально получены одним из роботов (то есть измерения классифицируются по слоям в зависимости от того, в ХЛД какого робота они содержатся). Каждый слой ХПД имеет такую же графовую структуру, как и ХЛД. Разделение ХЛД и ХПД позволяет сохранять информацию об источнике каждого из измерений и исключить проблему двойного учета, при этом нет необходимости передавать необработанные измерения: робот-отправитель может изменить локальный граф, и при получении обновленного графа робот-получатель сможет, анализируя УИД поз в старом и новом графе корректно объединить данные. Подробно данная процедура описана в пункте 2.4.1.

ХЛД содержит невизуальные ребра одометрии, созданные роботом. Роботы могут обмениваться информацией из своих ХЛД и изменять ее. Роботы не обмениваются информацией из ХЗР. Роботы могут обмениваться информацией из своего ХПД, но не могут изменять ее, кроме временных методов ориентиров.

Полная карта окружения получается путем объединения данных из всех слоев ХПД, ХЛД и ХЗР. Полная карта выравнивается относительно априорной карты с использованием имеющейся у робота, что позволяет найти позицию робота в глобальной системе координат.

Мы предполагаем, что всем роботам предоставлена идентичная априорная карта территории. Это важно, поскольку априорная карта (i) предоставляет глобальную систему координат для каждого из роботов, (ii) позволяет роботам “общаться” об определенных частях территории, даже если их полные карты не выровнены, или не похожи. В данной работе речь идет о картировании изменяющегося окружения, поэтому априорная карта не может обладать высокой точностью: территория будет постоянно меняться, а априорная карта будет становиться неточной. Однако мы предполагаем, что карта является достаточно точной, а крупные характерные части территории остаются неизменными (например, горы или здания при патрулировании на улице (меняющимися частями окружения являются машины и люди) или стены при патрулировании внутри помещения). Это позволяет совместить априорную карту с полной картой, имеющейся у робота.

В случае, если априорная карта не предоставлена, она может быть получена любым из стандартных методов картирования в предположении, что территория картирования является неизменной. Карта может быть не точной, однако она должна быть одинаковой для всех роботов. Также, если роботы производят картирование с перерывами, например, для дозаправки, то в процессе патрулирования роботы могут обновлять априорную карту на основе полной карты, полученной на основе данных от всех роботов.

2.2.3. Деление на регионы

Как было описано выше, каждая поза содержит параметр “идентификатор региона” (ИР). Данный параметр дает примерную оценку положения позы на территории патрулирования.

Для введения этого параметра априорная карта делится на регионы – такие части, которые полностью покрывают карту, и не пересекаются друг с другом. Разбиение на регионы должно быть одинаково для всех роботов, потому оно проводится на априорной карте перед началом работы коллектива роботов.

ИР каждой позы определяется в момент ее создания (на основе текущей оценки роботом положения позы). ИР позы не меняется с течением времени, даже если оценка позиции этой позы изменяется. Поэтому возможен случай, когда некоторые позы принадлежат некорректному региону, однако при этом все регион все равно будет достаточно точно задавать оценку положения позы, что и требуется. Разбиение на регионы служит двум целям: во-первых, оно упрощает контроль за плотностью поз на карте; во-вторых, оно позволяет роботам запрашивать и отправлять информацию об определенных частях карты, даже если их глобальные карты в данный момент времени не выравнены.

Для дальнейшего описания необходимо ввести понятие “лоскута” – это один регион одного слоя ХЛД или ХПД. Источником лоскута назовем робота, который является источником данных, хранимых в этом слое ХПД (или ХЛД). Опишем, из чего состоит ХЛД и каждый слой ХПД:

- Граф поз, состоящий из списка поз и списка невизуальных ребер одометрии.
- Для каждого лоскута слоя:
 - Временная метка лоскута
 - Короткий список идентификаторов (короткий ИД список) – список, содержащий все УИД поз из данного лоскута.

2.3. Функционирование системы, алгоритмы обработки данных

В данном разделе мы рассматриваем, как работает распределенная система ОЛИК. Для осуществления долгосрочного картирования местности, каждый робот коллектива должен определять направление движения, проводить измерения и обрабатывать их, контролировать размер хранилища данных, обмениваться данными с другими роботами, а также составлять глобальную карту территории и локализовать себя на ней. В данном разделе описаны все эти алгоритмы.

2.3.1. Обработка новой позы

Каждый раз, когда роботом создается новая поза, она добавляется в ХЛД робота, и происходит следующее:

1. Ребра невизуальной одометрии рассчитываются на основе показаний датчиков и также помещаются в ХЛД.
2. Ориентиры позы используются для нахождения ассоциирований данных. Найденные ассоциирования помещаются в ХЗР.
3. Происходит процедура обновления временных меток (см. пункт 2.3.1) для всех ориентиров, которые были успешно использованы для нахождения ассоциирований данных. Такое обновление происходит как в ХЛД, так и во всех слоях ХПД.

4. Происходит процедура обновления надежности ориентиров в позах ХЛД.
5. Если размер позы превышает ограничение на максимальный размер позы M_p , происходит процедура сжатия позы (см. 2.3.2).

Каждый раз, когда робот получает позу от другого робота, она добавляется в ХПД и происходит следующее:

1. Соответствующие ребра невизуальной одометрии добавляются в ХПД.
2. Ориентиры позы используются для нахождения ассоциирований данных позы; найденные помещаются в ХЗР.
3. Происходит процедура обновления временных меток (см. пункт 2.3.1) для всех ориентиров, которые были успешно использованы для нахождения ассоциирований данных. Такое обновление происходит как в ХЛД, так и во всех слоях ХПД.

Ассоциирование визуальных данных

Визуальное ассоциирование данных – это процедура, которая сравнивает положения ориентиров двух поз для нахождения их взаимного положения. Каждый раз, когда новая поза помещается в ХЛД или ХПД, необходимо найти возможные ассоциирования данных между этой позой и всеми другими позами. Сопоставить перебором все позы в режиме реального времени невозможно, поэтому перед непосредственными попытками сопоставления откидывается большинство поз-кандидатов, которые содержат недостаточное количество ориентиров с дескрипторами, похожими на дескрипторы новой позы. Для этого используется иерархическая кластеризация [127; 128]. Затем, оставшиеся кандидаты дополнительно фильтруются на основе ребер невизуальной одометрии, и только оставшиеся после этого кандидаты подвергаются сопоставлению с новой позой.

Для нахождения взаимного положения двух поз используется алгоритм RANSAC [129]. Каждый ориентир из одной позы сопоставляется с несколькими кандидатами (максимальное число сопоставлений определяется настраиваемым параметром M_{max}) с ближайшими дескрипторами из второй позы. Рассто-

яние между дескрипторами ориентира и сопоставленного ему ориентира другой позы должно быть не больше $d_{threshold}$. Если для ориентира не нашлось ни одного соответствующего сопоставления, то он выбрасывается из процесса сопоставления.

Рассмотрим алгоритм, используемый для сопоставления точек. Для этого проводится много раз следующий цикл:

1. Из множества всех сопоставлений выбирается три случайных сопоставления таких, чтобы эти сопоставления не имели общих ориентиров.
2. На основе трех сопоставлений находится начальная оценка пространственного преобразования между двумя позами (так, чтобы сопоставить три пары ориентиров).
3. Позиция каждого ориентира преобразуется с помощью полученного преобразования и проверяется, совпал ли этот ориентир после преобразования с одним из соотнесенных ему ориентиров. Допускается ошибка e_0 на каждый ориентир, если ошибка меньше, то ориентир считается удовлетворяющим преобразованию, иначе – выбросовым, неудовлетворяющим.
4. Подсчитывается число точек, удовлетворяющих алгоритму. Если число точек больше некоторого порога T_{inl} , строится новое преобразование, только на основе удовлетворяющих старому преобразованию точек. Так получается новое преобразование, и вычисляются точки, которые удовлетворяют этому новому преобразованию.
5. Предыдущий шаг повторяется до тех пор, пока число удовлетворяющих новому преобразованию точек не перестанет увеличиваться.

После повторения цикла фиксированное число раз, выбирается лучшее из найденных преобразований (по числу удовлетворяющих найденному преобразованию точек и по средней ошибке. Возможны различные критерии оптимальности в зависимости от специфики задачи). Также возможна досрочная остановка алгоритма в случае, если на каком-то шаге найдено очень точное преобразование.

Рассмотрим одну итерацию алгоритма RANSAC и соответствующее пространственное преобразование между двумя позами. Пара (два сопоставленных друг другу ориентира) (l_i, l_j) считается инлайером, если пространственное рас-

стояние $d(l_i, l_j)$ между ними меньше порога d_{max} .

$$E = \frac{\sum_{L_{l_i, l_j}} \omega(r_i, r_j) d^2(l_i, l_j)}{N_{inl}}, \quad (2.2)$$

здесь $\omega(r_i, r_j)$ – вес, зависящий от надежностей r_i и r_j ориентиров l_i и l_j . Штраф за неточное сопоставление ниже для ориентиров с низкой надежностью. Вероятностное распределение пространственного преобразования между позами оценивается с использованием результатов всех итераций алгоритма RANSAC, для которых $N_{inl} \geq N_{min}$, причем веса рассчитываются с использованием выражения (2.2).

Иерархическая кластеризация (ИК)

Для нахождения всех возможных ассоциирований данных между текущей позой и хранимыми в памяти, необходим алгоритм для быстрой фильтрации поз и нахождения поз-кандидатов, которые наиболее вероятно смогут быть сопоставлены: в реальном времени возможно сопоставление лишь с конечным числом поз алгоритмом RANSAC.

Для этого используются аппроксимационные алгоритмы поиска ближайших соседей с использованием иерархической кластеризации [127], например, библиотека FLANN [128] (от англ. Fast Library for Approximate Nearest Neighbors – “быстрая библиотека для поиска примерных ближайших соседей”).

Данный алгоритм работает следующим образом: все дескрипторы ориентиров, содержащихся в позах, помещаются в несколько рандомизированных древовидных структур (то есть, среди всех точек выбираются несколько точек, служащих центрами, и другие точки разбиваются на кластеры на основе близости к этим центрам; затем, аналогичные процедуры повторяются внутри каждого кластера). Для поиска ближайших соседей ориентира, последовательно вычисляется его принадлежность к кластерам разного уровня и затем, в последнем кластере вычисляются ближайшие точки. Для более точной работы алгоритма используются несколько таких рандомизированных деревьев.

Рассмотрим теперь, как такая структура данных может быть использована для нахождения кандидатов на сопоставление с новой позой. Как было описано выше, дерево ИК хранит все ориентиры из хранимых в ХЛД и ХПД поз. Каждый помещенный в дерево ориентир также содержит УИД позы, которой он принадлежит. Для нахождения поз-кандидатов для новой позы, выбираются N ближайших соседей (по дескрипторам) для каждого из ориентиров новой позы (возможно нахождение менее, чем N соседей, если не нашлось достаточного количества точек с близкими дескрипторами). Если новая поза содержит N_l ориентиров, то в дереве ИК находится всего $N_l N$ соседних ориентиров. Каждый из найденных ориентиров “голосует” за позу, УИД которой он содержит. Позы с наибольшим числом голосов (и с количеством голосов, превышающих некоторый порог) являются кандидатами, остальные удаляются из процесса сопоставления.

Обновление временных меток

Рассмотрим процедуру обновления временных меток ориентиров. Временная метка ориентира описывает, когда этот ориентир последний раз наблюдался каким-либо из роботов.

Обновление временных меток происходит каждый раз, когда в хранилище данных поступает новая поза (неважно, сгенерированная самим роботом или полученная по сети от другого робота). После появления новой позы, как описано выше, происходит ассоциирование данных между новой позой и старыми позами. Временные метки всех ориентиров, которые успешно участвовали в создании этих ассоциирований данных, обновляются: обе временные метки заменяются на максимальную из них (при создании новой позы, временная метка ориентира из новой позы равна текущему моменту). Если ориентир использован при создании нескольких ассоциирований данных, все временные метки соответствующих ориентиров выставляются равными максимальному значению из них.

Обратим внимание, что подобное обновление временных меток не ведет к появлению проблемы двойного учета, потому что для обновления используется функция максимума (а не суммы).

Обновление значений надежности

Надежность ориентира показывает, насколько часто и успешно ориентир используется для сопоставления. Обновление значений надежности происходит только когда находится ассоциирование данных между двумя позами из ХЛД, и только информация в ХЛД изменяется в процессе этого обновления (для избежания проблемы двойного учета).

Данная процедура увеличивает надежности ориентиров, которые были использованы для построения ассоциирования данных, и уменьшает надежности ориентиров, которые могли (то есть находились в области, где располагалась часть сопоставленных ориентиров), но не использовались (для которых не нашлось парного ориентира с похожим дескриптором, или парный ориентир не прошел порог отсечения по расстоянию d_{max}).

Назовем новую позу, помещение которой в ХЛД вызвало процесс ассоциирования данных, “новой” позой, и сопоставляемую с ней позу из ХЛД – “старой”. Обновление надежностей ориентиров происходит после того, как найдено успешное преобразование между позами, следующим образом:

1. Надежность каждого ориентира X из старой позы, который не сопоставился ни с одним ориентиром из новой позы, делится на величину $1 + \beta d$, где d – плотность ориентиров в новой позе в позиции X (если у новой позы в положении этого ориентира располагается много собственных ориентиров, то факт, что ориентир старой позы не сопоставился ни с одним из новой, является более значимым; с другой стороны, если новая поза просто не содержит ориентиров в этой области, то уменьшать надежность ориентира старой позы не следует, просто потому что этот ориентир не попал в поле зрения робота, когда он конструировал новую позу). Численный коэффициент β выбирается в зависимости от среднего количества ориентиров, обнаруживаемых роботами и содержащимися в позах. При этом коэффициент β должен быть подобран таким образом, чтобы $L \geq 1 + \beta d \geq 1$ – надежность ориентира, который не сопоставился, не может ослабиться сильнее, чем в L раз или увеличиться. Варьирование параметра L позволяет контролировать, насколько быстро уменьшается

надежность ориентиров, когда они не используются для сопоставления поз.

2. Если ориентир успешно использовался для построения ассоциирования данных, то обе надежности – его и парного к нему ориентира становятся равными $r_{new} = 1 - (1 - r_1)(1 - r_2)$.

2.3.2. Контроль размера ХЛД

Чтобы сделать возможным длительное функционирование роботов необходимо обеспечить, чтобы размер ХЛД не увеличивался неограниченно с течением картирования. Для этого мы регулируем число поз в каждом из регионов так, чтобы объем ни одного из регионов не превышал M_r регионов и обеспечиваем, что объем каждой из поз не превышает M_{pose} . Для этого используются процедуры, описанные в пункте 2.3.2.

Контроль размера поз и их сжатие

Сжатие позы происходит когда новая или объединенная поза имеет размер больше порога M_{pose} . При сжимании позы мы удаляем из позы ориентиры с наименьшей ценностью U из позы, пока ее размер не станет допустимым (менее M_{pose}). Для оценки ценности U каждого из ориентиров используется выражение:

$$U = \frac{\phi(t, r)}{d}, \quad (2.3)$$

где t – это текущее время минус временная метка ориентира (оно равно 0, если ориентир принадлежит только что созданной позе), r – надежность ориентира, а d – плотность ориентиров в окрестности рассматриваемого ориентира (чем больше ориентиров находится в данной области, тем более они заменяемы, причем после работы алгоритма из каждого кластера ориентиров уберутся наименее на-

дежные и наиболее старые). Функция $\phi(t, r)$ показывает, насколько быстро ориентиры становятся устаревшими и насколько важен параметр надежности. Мы предлагаем использовать следующую функцию: $\phi(t, r) = \alpha e^{\frac{-\gamma t^2 (1-r)^2}{\sigma^2}}$. Для оценки плотности ориентиров используется ядро Гаусса [130].

Контроль размера регионов и их сжатие

Сжатие регионов необходимо для контроля плотности поз на территории патрулирования и для контроля объема информации, хранимой каждым из роботов. Для этого ставится ограничение M_r на объем данных, хранимых в каждом регионе. При этом необходимо сохранять только самую важную и нескоррелированную информацию.

Процедура сжатия регионов проводится каждый раз, когда память, занимаемая некоторым регионом, превышает M_r . Кроме того, процедура используется при передаче информации между роботами. Если необходимо послать несколько лоскутов другому роботу в условиях ограничения T на объем передаваемых данных (подробнее описано в разделе 2.4.2), то данные из этих лоскутов сжимаются аналогичным алгоритмом.

Обе данных операции заключаются в уменьшении размера набора лоскутов за счет выбора наиболее важных поз из набора и отбрасывания остальных. Когда робот удаляет позу, он освобождает часть памяти, которую занимала поза; когда робот объединяет две позы, он освобождает только часть памяти от обеих поз: часть памяти освобождается вследствие объединения сеток проходимости двух поз; дополнительно, освобождается часть памяти, если размер объединенной позы превышает M_p (вследствие чего вызывается процедура сжатия этой позы). Когда роботы необходимо просто уменьшить размер региона, он для этого может как объединять, так и удалять позы. Когда же робот подготавливает к отправке другому роботу набор поз, он может только выбрасывать позы, поскольку объединение поз приводит к созданию позы с новым УИД, который отсутствует у робота-приемника информации. В остальном, алгоритмы аналогичны.

Каждая поза может быть объединена только с позой, которая располагается близко к ней и с которой существует корректное и надежное ассоциирование данных (которые, как описано выше, строятся сразу после помещения новой позы в хранилище данных). Для каждой позы нам необходимо оценить, сколько памяти мы освобождаем и сколько информации мы теряем при (i) удалении этой позы и (ii) объединении этой позы с некоторой другой позой. Пусть M – объем памяти, освобождаемый в результате такого действия, а d – плотность поз в окрестности рассматриваемой до ее удаления (оцененная с помощью ядра Гаусса [130]). Пусть $\sum_{l \in L} U_l$ – это оценка суммарной полезности удаленных (в результате удаления содержащей их позы) ориентиров, U_l каждого ориентира считается с использованием выражения (2.3). Для каждой операции удаления или объединения находится эффективность:

$$E = \frac{M}{d \sum_{l \in L} U_l} \quad (2.4)$$

– фактически, сколько памяти мы освобождаем за единицу потерянной информации.

Обратим внимание, что чтобы удовлетворить требованиям на размер региона M_r , нам может понадобиться удалить более одной позы. Предлагаемое нами последовательное удаление (или объединение) наименее информативных поз (т.е. использование “жадного алгоритма”) может вести к неоптимальному решению. Однако нахождение оптимального решения слишком трудоемко – такая задача аналогична задаче про ранец [131].

Объединение поз

Объединение поз – процедура, проводимая для уменьшения объема данных, применяемая, когда размер региона, содержащего позы, превышает порог M_r . При процедуре две позы объединяются, для объединенной позы выбирается новая система координат и присваивается новый УИД. Карта проходимости поз объединяется (подробнее про карты проходимости написано в главе 4). Затем пересчитываются позиции ориентиров относительно новой системы координат

(и ориентиры, которые использовались для сопоставления поз, объединяются, и помещаются в дерево ИК).

Позы могут быть объединены только если существует точная оценка преобразования между их положениями: число ориентиров, по которым они сопоставлены должно быть не менее N_l , а распределение, задающее преобразование между позами, должно обладать малой дисперсией. В данной работе мы не приводим формальных критериев для пригодности поз для объединения – настройка этих алгоритмов зависит от качества работы алгоритма стереосопоставления и от количества обнаруживаемых роботом ключевых точек.

2.3.3. Построение глобальной карты

Для построения глобальной карты происходит объединение данных из всех хранилищ данных робота: ХЛД, ХПД и ХЗР. Для этого предлагается использовать алгоритм графовой оптимизации:

1. Конструируется граф, содержащий позы как из ХЛД, так и ХПД. Вершинами графа являются позы, а визуальные ассоциирования данных – ребрами. Граф оптимизируется и находится наиболее вероятная конфигурация взаимных позиций [132].
2. Используя информацию о локальных картах проходимости, происходит построение карты проходимости для всей территории.
3. После получения глобальной карты проходимости итоговая глобальная карта выравнивается с априорной картой (см. пункт 2.3.3). Такое выравнивание позволяет рассчитать позицию робота в глобальной системе координат и привязать позиции найденных ориентиров к координатам на априорной карте.
4. Выбросовые и некорректные ассоциирования данных обнаруживаются на данном этапе и удаляются из ХЗР. Точная процедура определения выбросовых ассоциирований зависит от используемого внутреннего алгоритма ОЛИК.

Поскольку оптимизация графа – довольно емкая вычислительная задача, предлагается не производить пересчет всего графа после получения каждого нового измерения, а проводить его после накопления достаточно большого объема свежих данных (или накопления большого количества новых ассоциированных данных).

Чтобы осуществить выравнивание полученной глобальной карты и априорной карты мы используем быстрое преобразование Фурье (БПФ) и L^2 метрику [133]. Мы проводим сравнение всех возможных найденных комбинаций поворота и сдвига и сохраняем N лучших результатов, а затем используем фильтр частиц для поддержания корректной гипотезы.

2.3.4. Алгоритм движения роботов

Чтобы карта, которую строят роботы, оставалась актуальной каждая часть территории должна регулярно наблюдаться тем или иным роботом; иначе говоря, роботы должны осуществлять задачу патрулирования. Алгоритмам патрулирования посвящена глава 3 данной работы. Обратим внимание, что при использовании алгоритмов патрулирования, которые позволяют роботу заранее предсказать направление его движения, позволяет заранее отправлять запросы на информацию об ориентирах той области, куда робот планирует двигаться.

2.4. Процедура обмена данными

В данном разделе описан алгоритм обмена данными по сети. Описано два способа обмена данными: первый, более простой алгоритм, используется в случае, когда пропускной канал позволяет передавать весь необходимый объем данных. Второй способ более сложен и предназначен для выбора и обмена наиболее важной информации для случаев, когда пропускная способность сети ограничена.

2.4.1. Алгоритм для работы в нормальных условиях

Даже в условиях, когда сеть обладает высокой пропускной способностью, роботы не обязаны обмениваться всей доступной информацией: это может замедлить ее обработку. Вместо этого, роботы могут запрашивать информацию об интересующих их лоскутах карты. Это позволяет роботам узнавать о любой конкретной части карты, или о данных от одного конкретного робота. Например, если робот хочет иметь всю информацию об определенном регионе, он может запросить других роботов послать ему $N_r - 1$ лоскутов, относящихся в этому региону. Таким образом, любой запрос может быть рассмотрен как комбинация запросов на один лоскут. Рассмотрим процедуру передачи одного лоскута X от робота B роботу A :

1. Робот A создает запрос, в котором описано, какой лоскут его интересует, и прикрепляет короткий ИД лист (список, содержащий все УИД поз из данного лоскута, см. пункт 2.2.3) для этого лоскута.
2. Если робот B – не источник данного лоскута, и временная метка для этого лоскута у робота B меньше либо равна временной метке робота A , ничего не происходит.
3. Если робот B – источник запрашиваемого лоскута, или его временная метка выше временной метки робота A , то:
 - (a) Робот B отправляет все позы из своего ХЛД (если он является источником) или соответствующего слоя ХПД, относящиеся к указанному региону, и которые не содержатся в коротком листе ИД робота A . Он также посылает свой короткий лист ИД для данного лоскута. Все ребра одометрии, прикрепленные к отосланным позам, также передаются роботу A .
 - (b) Робот A устанавливает временную метку лоскута на текущее время (если робот B является источником лоскута), или выставляет ее равной временной метке робота B (если робот B не является источником). Робот A удаляет все позы из лоскута, которые не включены в короткий лист ИД робота B и добавляет все полученные позы в соответствующий слой ХПД. Все

ребра одометрии, прикрепленные к этим позам, также передаются роботу A .

При ответе на запрос робота об определенном лоскуте, ему должен отвечать источник этого лоскута (если возможно), а иначе – робот, обладающий максимальной временной меткой на данном лоскуте.

Обратим внимание, что в случае полной связности и достаточного пропускного канала, роботы постоянно могут обмениваться информацией о всех лоскутах. В таких условиях каждый робот будет обладать полной копией ХЛД каждого другого робота, и тогда получаемые роботами глобальные карты также будут одинаковыми.

2.4.2. Алгоритм для работы в условиях недостаточной пропускной способности сети

Если роботы будут пытаться обмениваться полным объемом информации из ХЛД и ХПД в случае, когда пропускная способность сети для этого недостаточно, то произойдет “перегрузка” сети, и только часть информации будет действительно доставлена вовремя. Чтобы избежать такого сценария, роботам следует обмениваться только наиболее важными и необходимыми частями информации об интересующих частях территории. Простой алгоритм, описанные выше, позволяет обмениваться информацией об избранных регионах. Ниже описан алгоритм, позволяющий выбирать только наиболее важные части информации.

Обмен данными обычно инициируется запросом одного из роботов. Для сжатия информации из нескольких лоскутов, эти лоскуты должны быть смежными (то есть образовывать связное множество) и иметь один источник. Такой запрос не может быть (в отличие от запросов, рассмотренных выше) рассмотрен как комбинация запросов об одном лоскуте.

Для того, чтобы смоделировать ограничение, выдвигаемое пропускной способностью сети, ограничим допустимый размер ответа параметром T . Чтобы удовлетворить этому требованию, отвечающий робот должен выбрать поднабор из поз и ребер одометрии, которые он пошлет запрашиваемому роботу, причем так,

чтобы размер ответа не превысил T и содержал как можно больше полезной информации. Мы предполагаем, что роботам задано ограничение T на объем передаваемых данных.

Для имплементации данного алгоритма мы дополняем ХПД структурой, называемой “полным ИД листом”. Аналогично короткому ИД листу, каждый робот хранит полный ИД лист для каждого лоскута в ХПД. Полный ИД лист лоскута состоит из списка всех УИД поз, которые робот-источник содержал в момент времени, равный временной метке, проставленной в лоскуте. Иначе говоря, каждый раз, когда некоторый робот A запрашивает информацию о лоскуте у некоторого робота B (причем робот B является источником лоскута), робот B может вернуть только часть набора поз из лоскута (из-за ограничения T на размер ответа), но он обязан предоставить полный ИД лист, содержащий все УИД поз лоскута. После этого, каждый раз, когда роботы обмениваются информацией об этом лоскуте, они также передают полный ИД лист. Для ХЛД полный и короткий ИД листы совпадают.

Рассмотрим непосредственно алгоритм. Пусть робот A хочет получить информацию нескольких смежных лоскутах от робота B (который не обязательно является источником этих лоскутов, но лоскуты имеют один источник). В запросе робот A должен отправить короткий и полные ИД листы для лоскута. Ограничение T на размер ответа считается известным.

Перед тем, как формировать ответ, робот B рассматривает потенциальное множество поз, в которых заинтересован робот A . После этого робот B выбирает подмножество, которое удовлетворит требованию на размер ответа T ; если возможно, он передает все позы. Кроме выбранных поз, передаются также все ребра, соединяющие эти позы (поскольку размер ребра значительно меньше размера позы, то мы не учитываем размер ребер при подсчете объема передаваемых данных).

Для создания множества потенциально интересных поз, робот B последовательно рассматривает каждый из лоскутов запроса. Если робот B обладает меньшей, чем робот A , временной меткой для данной лоскута, он выбирает позы из своего короткого ИД листа, которые присутствуют в полном, но отсутствуют в коротком ИД листе A . Эти позы добавляются к множеству поз, потенциально ин-

интересных робота A . После этого робот B заменяет полный ИД набор для данного лоскута полным ИД набором, посланным роботом A .

Если робот B обладает большей временной меткой для данного лоскута, он передает полный ИД лист для данного лоскута робота A и свою временную метку (если робот B – источник, то он передает текущее время в качестве временной метки). Затем робот B добавляет все позы, которые присутствуют в его коротком ИД листе, но отсутствуют в коротком ИД листе робота A к множеству потенциально интересующих робота A поз.

Если роботы A и B обладают равными временными метками данного лоскута, это означает, что их полные ИД листы также идентичны. В этом случае робот B добавляет позы, которые присутствуют в его коротком ИД листе, но отсутствуют в коротком ИД листе робота A к множеству потенциально интересующих робота A поз.

После того, как множество потенциально интересных поз создано, выбирается набор поз с максимальной ценностью E (см. раздел 2.4), который удовлетворяет ограничению T на размер ответа. Этот набор поз, а также все прикрепленные к позам ребра отправляются роботу A .

После получения ответа робот A :

1. Обновляет полный ИД лист и значения временных меток для всех лоскутов, в которых робот B имел большие временные метки.
2. В каждом лоскуте робот A удаляет все позы, которые отсутствуют в полном ИД листе и удаляет ребра одометрии, прикрепленные к ним.
3. Добавляет все полученные позы и прикрепленные ребра в соответствующие регионы ХПД и обновляет свои короткие ИД листы.

2.5. Выводы к главе

В данной главе разработана архитектура хранения, обработки и обмена информации коллективом роботов, осуществляющих длительное картирование меняющейся местности. Предложенная архитектура позволяет

1. Избежать распространенной проблемы “двойного учета” измерений.

2. Снять ограничение длительности функционирования коллектива роботов, связанное с ограничением объема их памяти.
3. Эффективно использовать существующие алгоритмы локализации и картирования при использовании коллектива роботов.
4. Эффективно использовать существующие алгоритмы локализации и картирования в условиях, когда ориентиры, используемые роботами, меняются с течением времени.

Глава 3. Стохастическое патрулирование коллективом роботов с использованием метода потенциалов

Основное содержание данной главы опубликовано в работах [134; 135].

3.1. Постановка задачи

В данной главе описана разработка полностью распределенного, стохастического алгоритма патрулирования на основе метода потенциалов. Преследуется цель разработать алгоритм, который легко может быть модифицирован для имплементации других типов поведений. Для тестирования алгоритма разработана и написана система имитационного моделирования на языке программирования C++ [136]. Система позволяет тестировать работоспособность и эффективность различных алгоритмов. С использованием этой системы также исследуется вопрос о падении эффективности патрулирования при работе системы в различных условиях сети, таких как связь с ограниченным радиусом и прерывистая связь.

Карта территории известна заранее (построена одним из алгоритмов картирования, описанных в предыдущей главе). Предполагается, что возможны незначительные изменения в территории патрулирования с течением времени.

В качестве критерия оценки эффективности патрулирования используется среднее значение $MTB(t)$ за время патрулирования, как описано в разделе 1.3.1.

Для движения робота по территории патрулирования ему необходимо успешно объезжать препятствия. В данной работе преследуется цель разработать систему потенциальных полей, которая позволят правильно выбирать глобальные цели движения (двигаться к необследованным точкам, таким образом поддерживая низкое значение MTB), и одновременно успешно огибать локальные препятствия.

Для дальнейшего обсуждения введем некоторые обозначения. Пусть $P(R, T)$ – путь из текущего положения робота R до точки T территории. Дан-

ный путь не обязательно является прямолинейным, поскольку роботу необходимо объезжать непроходимые препятствия. Направлением пути P назовем направление луча, касательного к кривой пути в точке нахождения робота (по направлению движения). Длиной пути $L(R, T)$ назовем длину пути $P(R, T)$.

Принцип патрулирование на основе метода потенциалов может быть кратко сформулирован следующим образом: каждая необследованная точка T территории оказывает на робота R с виртуальную силу, которая зависит от значения MTB в точке и от расстояния от точки до робота. Эта сила “тянет” робота в направлении пути $P(R, T)$, которое ведет от робота к точке. Таким образом, при наличии, например, только одной необследованной точки в каждый момент времени, робот будет двигаться к этой точке. Когда же на робота воздействует множество точек, робот принимает решение о движении на основе их суперпозиции. Принцип выбора движения может быть как “движение в направлении векторной суммы сил”, так и “движение в сторону максимального притяжения”.

При использовании описанного подхода на каждом шаге алгоритма необходимо находить пути от каждого из роботов до каждой точки территории (чтобы рассчитать все силы, которые действуют на робота). Поэтому для работы алгоритма патрулирования в реальном времени необходим быстрый алгоритм нахождения маршрутов. Хотя при полностью статичной территории возможен предварительный подсчет путей и составление карты маршрутизации, при изменениях карты вероятны сценарии, когда все маршруты изменяются (например, если узкий проход между разными частями территории становится непроходимым), поэтому посчитанные заранее маршруты становятся некорректными.

Формально, алгоритм, управляющий движением робота, состоит из следующих частей:

1. Нахождение пути $P(R, T)$ от робота до каждой точки территории.
2. Расчет силы притяжения $F(R, T, MTB(T))$, с которой каждая из точек действует на робота.
3. Выбор желаемого направления движения робота на основе анализа всех сил, которые на него действуют.
4. Корректировка идеального желаемого направления движения для объезда локальных препятствий и избежания взаимных столкновений между роботами.

В работе мы рассматриваем отдельно каждую из этих задач: выбор алгоритма нахождения пути, выбор способа подсчета силы притяжения и выбор желаемого направления движения на основе анализа действующих сил.

Некоторые роботы обладают ограниченной маневренностью, и для таких роботов алгоритм движения должен учитывать эту особенность. Будем называть голономным роботом такого, который способен перемещаться мгновенно в любом направлении в пространстве его степеней свободы (то есть число управляемых степеней свободы равно полному числу степеней свободы). Неголономным роботом будем называть робота, для которого это условие не выполняется. В реальных сценарии большинство роботов не являются неголономными. В данной работе мы рассматриваем как голономных, так и неголономных роботов, обладающих минимальным радиусом поворота. Непосредственное использование метода потенциалов для контролирования движения неголономных роботов проблематично: выбирая предпочтительное направление движения, метод потенциалов не учитывает доступные для робота маневры. Например, метод потенциалов может требовать от робота резких смен направлений движения (когда некоторая точка с большим , притягивавшая робота, попадает в поле зрения другого робота). Поэтому, чтобы отделить проблему выбора оптимального направления движения от проблемы прокладывания маршрута, который неголономный робот может реализовать, в работе задача организации патрулирования решается пошагово:

1. Решение задачи патрулирования территории без препятствий голономными роботами.
2. Решение задачи патрулирования территории с препятствиями голономными роботами.
3. Решение задачи патрулирования территории с препятствиями неголономными роботами.

Каждый из этих алгоритмов имеет сценарии, когда он может быть использован. Например, роботы-пылесосы и сборщики мусора [137] могут быть голономными, то есть мгновенно менять направление движение. Задача оптимизации их движения для сбора мусора в открытом помещении попадает в категорию 1. Задача сбора мусора в помещении с большим числом препятствий - в категорию 2. Большие колесные роботы, как правило, обладают минимальным радиусом раз-

ворота, и осуществляют патрулирование вне помещений. Такая задача является основной целью работы и попадает в категорию 3.

3.2. Система моделирования

Для тестирования работоспособности и оценки эффективности разработанных алгоритмов патрулирования имплементирована система имитационного моделирования на языке C++. Моделирование ведется пошагово. На каждом шаге алгоритма роботы обмениваются своими текущими координатами. Каждый робот моделируется отдельно. В большинстве сценариев моделируется случай полностью связанной сети, когда роботы обладают всей информацией, доступной другим роботом. Однако симулятор позволяет моделировать ограниченную сеть, в этом случае каждый робот принимает решения на основе доступных только ему данных.

Голономные роботы представляются как круглые объекты радиусом 1 м, которые не могут пересекать границы карты или сталкиваться друг с другом (минимальное расстояние между центрами двух голономных роботов составляет 2 метра). В данной работе модуль скорости движущегося робота считается постоянным и равным 4.4 метра в секунду (или 15.8 км/ч). Робот также может быть неподвижным. Голономные роботы могут мгновенно менять направление движения.

Неголономные роботы отличаются от голономных наличием минимального радиуса поворота, равного 3.5 метрам. Неголономные роботы могут осуществлять движение задним ходом (с таким же ограничением на радиус поворота).

При моделировании радиус зрения роботов равен 10 метрам, кроме случаев, где указано иначе.

Симуляция ведется пошагово с шагом обсчета, соответствующим 0.5 секунды реального времени. Такая частота переоценки ситуации обусловлена необходимостью оперативно реагировать на изменяющуюся ситуацию, например, избегать столкновения с замеченным, но отсутствующим на карте, препятствием. Также высокая частота способствует увеличению плавности хода роботов.

На каждом шаге симуляции робот оценивает обстановку на территории патрулирования, рассчитывает искусственные силы, действующие на него и движется прямолинейно в выбранном направлении. Если алгоритм движения вынуждает несколько роботов двигаться на одном шаге симуляции так, что между ними возможно столкновение, то движение всех роботов отменяется на этом шаге симуляции. В сценарии патрулирования других ситуаций, когда роботы не осуществляют движение, не рассматривается.

Обратим внимание, что если движение двух роботов будет отменено согласно правилу, введенному в предыдущем абзаце, то на следующем шаге симуляции вероятно их повторное столкновение (поскольку их позиции не изменились). Подобные ситуации редки, но могут значительно повлиять на качество патрулирования, если “столкнувшиеся” роботы блокируют движения друг друга в течение длительного времени. Для разрешения проблемы при использовании голономных роботов, в случае блокировки на каждом шаге моделирования каждый робот с вероятностью 10% может выбрать случайное направление для движения. Подобное правило не оказывает значительного влияния на эффективность патрулирования в стационарном режиме, но позволяет избежать длительной взаимной блокировки, а также обеспечивает случайность и непредсказуемость поведения роботов. Подобное правило используется только для голономных роботов: для неголономных роботов разработан специальный алгоритм, позволяющий разрешить подобную ситуацию, описанный в разделе 3.5.

На рисунке 3.1 приведен скриншот симулятора в процессе работы. В правой верхней части окна располагается панель управления симуляцией. Для запуска моделирования необходимо сначала загрузить карту территории. Карта территории должна быть представлена в виде черно-белой картинки (которая описывает карту проходимости), причем один пиксель картинки описывает квадрат территории 10 на 10 сантиметров. Затем необходимо загрузить позиции роботов. Информация о позициях роботов хранится в текстовом файле в следующем формате:

1. первая строка - число роботов
2. следующие строки содержат разделенные пробелами поля:
 - (a) координата X в дециметрах
 - (b) координата Y в дециметрах

(с) угол ϕ в радианах, описывающий ориентацию робота в момент начала симуляции

Пример содержимого файла, задающего позиции роботов:

```
4
260 85 0
790 1070 0
1115 80 0
385 525 0
260 1105 0
```

Далее, следует загрузить семафоры (опционально) и запустить процесс моделирования. Моделирование можно проводить как пошагово, так и в непрерывном режиме. Конфигурации семафоров хранятся в текстовом файле в следующем формате:

1. Каждый семафор описывается множеством вершин многоугольника, ограничивающим его территорию. Они записываются парами координат вершин, разделенных пробелами.
2. Пары координат разделяются точкой с запятой без пробелов.
3. Каждый семафор описывается в отдельной строке.

Пример содержимого файла конфигурации семафоров:

```
953 615;753 551;787 315;1063 97;1181 100;1181 241
757 967;603 1003;577 839;743 531;987 609
```

Центральная часть окна визуализирует ситуацию на территории патрулирования. Роботы изображаются в виде кружков, а отрезок показывает направление, в котором робот развернут. Черным отображаются препятствия на территории. Фон варьируется от белого до серого в зависимости от значения MTB в точке: чем больше MTB , тем темнее точка. Как видно из рисунка, в радиусе обзора роботов точки отображаются белым, поскольку они наблюдаются в данный момент.

Результатом работы системы моделирования является скриншоты моментальных конфигураций роботов, записанные на каждом шаге моделирования, в формате png. Одновременно, численные метрики сохраняются в файле out.txt. Формат записи: время - МТВ - ДВТ.

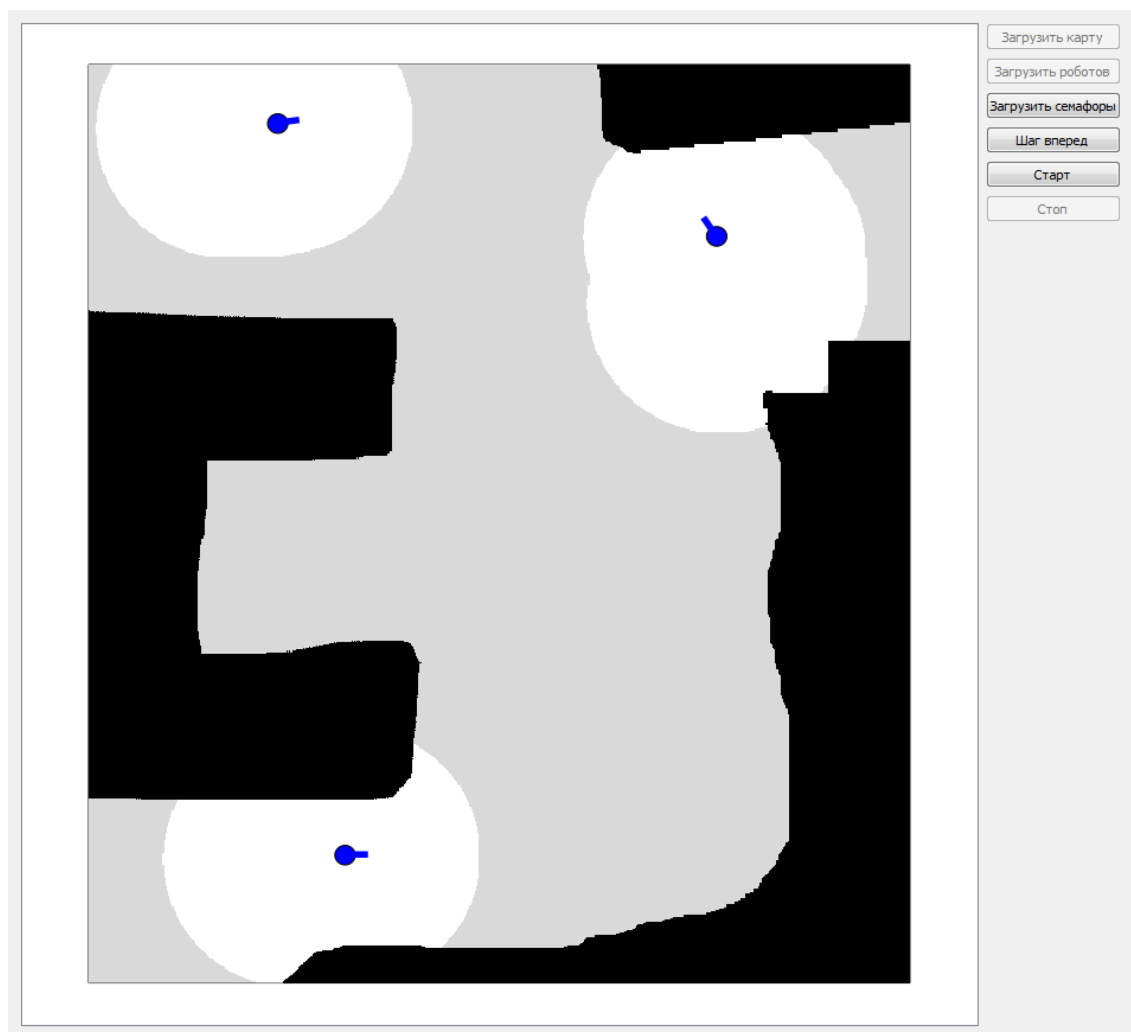


Рисунок 3.1. Скриншот симулятора патрулирования в действии

Симулятор также способен работать в режиме командной строки. При этом не создается скриншотов, что ускоряет проведение симуляции. В режиме командной строки возможно планирование и проведение серии экспериментов с варьирующимся набором параметров. При запуске в режиме командной строки параметры моделирования загружаются из файла “setup.json” (находится в папке model/testdata). Сам запуск осуществляется посредством запуска robot_cmd.exe.

В рамках работы с использованием данной системы моделирования было поставлено порядка тысячи численных экспериментов для различных сценариев.

3.2.1. Варьируемые параметры моделирования

- “robot_starting_position”: путь к файлу, в котором хранятся стартовые позиции роботов. Относительный путь должен указываться относительно расположения исполняемого файла.
- “map”: путь к файлу, в котором хранится карта территории патрулирования.
- “layout”: путь к файлу, содержащему информацию о семафорах. При отсутствии этого параметра загрузка семафоров не производится.
- “robot_number”: число роботов, ведущих патрулирование. Данное число имеет приоритет над числом, указанным в файле с позициями роботов.
- “worsening_window”: значение крыла дилатации
- “max_model_time”: число шагов моделирования
- “output_file”: файл, в который записываются результаты моделирования. Запись идет в формате шаг симуляции – МТВ – ДВТ
- “log_period_steps”: шаг, с которым ведется запись в файл.

3.2.2. Дополнительные возможности конфигурирования моделирования

Для изменения других параметров программы необходима повторная компиляция программы. Конфигурируемые параметры находятся в файле `model/prj.core/robot_simulator/src/model_parameters.h`

- “SCENARIO”: номер реализуемого сценария
- “PATROL_TIME”: время патрулирования до подачи команды на возвращение на исходные позиции
- “VISION_RADIUS”: радиус обзора роботов (в пикселях, 1 пиксель = 10см)
- “START_FOG”: туман войны в момент начала симуляции
- “TURN_RADIUS”: минимальный радиус поворота робота

3.3. Патрулирование территории без препятствий голономными роботами

Создание алгоритма, эффективно функционирующего на территории без препятствий, необходимо для разработки более сложных алгоритмов, поскольку патрулирование открытой местности – основной режим, в котором будут функционировать БНТС. В данном разделе рассматривается задача патрулирования голономными роботами местности без препятствий, и на примере решения этой задачи принимается решение о выборе функции $F(R, T)$, определяющей силу, которая действует на робота со стороны точек карты.

Об эффективности алгоритма, управляющего движением роботов, удобно судить именно по тому, как он функционирует на территории без препятствий: при движении на местности с большим количеством препятствий движения робота значительно ограничены: например, в узком тоннеле у робота есть всего лишь один вариант движения – вдоль стен. На открытой местности робот выбирает из большего числа направлений движения, и особенности алгоритма оказывают значительное влияние на эффективность патрулирования. Использование задачи патрулирования на местности без препятствий для выбора функции F удобно и

по другой причине. Как было описано в предыдущей секции, алгоритм движения робота состоит из трех частей: нахождение пути от робота до каждой точки карты; нахождение силы $F(R, T)$, с которой каждая точка карты действует на робота; и выбор направления движения в зависимости от действующих сил. В случае отсутствия препятствий результаты патрулирования зависят в большей степени именно от выбора функции F : путь $P(R, T)$ от робота до точки находится естественным образом – как отрезок, соединяющий точку T и центр робота R .

В работе мы рассматриваем два метода выбора направления движения в зависимости от действующих сил:

1. Робот осуществляет движение по направлению векторной суммы сил, действующих на него. Этот метод выбора направления движения далее называется “методом векторной суммы”.
2. Рассчитывается сумма сил, действующих на робота в каждом из направлений. Робот двигается по направлению максимума действующих сил (в сторону максимального притяжения). При использовании данного подхода направление для движения выбирается как мода угловой гистограммы направленности действующих сил. Этот метод выбора направления движения далее называется “методом моды”.

При оценке оптимальной функции $F(R, T)$ мы используем оба из предложенных алгоритмов.

При поиске функции F мы рассматриваем функции вида:

$$F(R, X) = f(FW(X)) \cdot g(L(R, X)), \quad (3.1)$$

$$FW_n(X) = \frac{FW(X)}{MFW(t)},$$

где функция $f(TB(T))$ зависит только от значения TB в точке T , а функция $g(L(R, T))$ – Только от длины пути от робота R до точки T .

Мы рассматриваем следующие функции f и g и все их возможные комбинации:

Разрабатываемый алгоритм должен быть применим для использования на территориях различных размеров и с различным числом роботов. Поэтому возможны различные средние значения тумана войны TB . Чтобы избежать проблем с возведением экспоненты в большую степень при вычислении TB , проводится нормировка тумана войны на максимальное значение $MTB(t)$ в каждый момент

$f_1 = FW_n(X)$	$g_1 = \frac{1}{L(R,X)}$
$f_2 = FW_n^2(X)$	$g_2 = \frac{1}{L(R,X)^2}$
$f_3 = e^{\alpha \cdot FW_n(X)}$	$g_3 = e^{-\frac{L(R,X)}{\beta}}$
$f_4 = e^{\alpha \cdot FW_n^2(X)}$	$g_4 = e^{-\frac{L^2(R,X)}{\beta^2}}$
$f_5 = e^{\alpha \cdot FW_n^4(X)}$	$g_5 = e^{-\frac{L^4(R,X)}{\beta^4}}$

Таблица 1. Рассматриваемые в работе функции f и g

времени согласно (3.1). Таким образом, значение нормированного TB_n в каждой точке территории принимает значение от 0 до 1. В этом и последующих разделах индекс опускается и обозначение TB используется для обозначения нормированного тумана войны.

Для выбора комбинации функций f и g производится моделирование патрулирования тремя роботами на площадке без препятствий с размерами 100 на 88 метров. Такой размер выбран, чтобы избежать возможных эффектов, которые могут возникнуть на картах строго квадратной формы. Центры трех роботов в момент начала моделирования задаются парами координат, отсчитываемых в дециметрах от левого верхнего угла карты: (480, 110), (660, 640), (205, 310).

В таблицах 2, 3 приведены средние значения MTB за время симуляции с использованием обоих методов выбора направлений движения.

	g_1	g_2	g_3	g_4	g_5
f_1	1476.0	1306.9	136.1	199.2	763.4
f_2	927.5	290.5	119.1	126.0	218.1
f_3	260.3	143.9	101.5	80.7	111.1
f_4	257.9	119.2	204.1	76.1	120.6
f_5	303.4	125.0	243.5	94.3	125.6

Таблица 2. Результаты моделирования для голономных роботов на карте без препятствий, среднее значение MTB , секунды (метод векторной суммы)

Как видно из результатов экспериментов, на карте без препятствий наиболее эффективного патрулирования можно добиться, используя метод векторной

	g_1	g_2	g_3	g_4	g_5
f_1	1211.3	545.6	132.7	144.3	394.2
f_2	490.1	204.7	127.9	113.8	167.1
f_3	167.0	133.0	98.5	90.0	125.8
f_4	167.9	140.8	118.2	92.9	116.5
f_5	168.5	142.3	192.8	106.7	128.3

Таблица 3. Результаты моделирования для голономных роботов на карте без препятствий, среднее значение MTB , секунды (метод моды)

суммы. Отметим, что вне зависимости от метода выбора направления движения наиболее эффективными комбинациями функций являются f_3g_4 и f_4g_4 . Как будет показано ниже, для реальных территорий с наличием препятствий несколько большую эффективность демонстрирует метод моды. Поэтому было принято решение использовать следующую комбинацию функций, которая показывает максимальную эффективность при использовании метода моды. Во всех последующих разделах для вычисления силы притяжения $F(R,T)$ используется указанная формула:

$$F(R,T) = f_3(T)g_4(R,T) = e^{\alpha \cdot TB(T)} \cdot e^{-\frac{L^2(R,T)}{\beta^2}} \quad (3.2)$$

Здесь α и β – нормировочные коэффициенты для эффективного использования представления числа с плавающей точкой, l – коэффициент, отражающий характерный линейный размер территории (в дециметрах), длина пути L также измеряется в дециметрах. Для проводимых в данной работе экспериментов выбраны следующие значения параметров:

$$\alpha = 40,$$

$$\beta = 65,$$

$$l = 800 \text{ дм.}$$

3.4. Патрулирование территории с препятствиями голономными роботами

Наличие препятствий на территории патрулирования создает две проблемы: во-первых, появляется необходимость в использовании алгоритма нахождения маршрута: отрезок, соединяющий робота R и точку T , не является корректным маршрутом, если он пересекает препятствия. Сила, с которой точка притягивает робота, также должна быть направлена не по отрезку, соединяющему робота и точку, а по пути, который огибает препятствия и ведет от робота к точке.

Алгоритм нахождения пути должен отрабатывать на каждом шаге алгоритма, находя пути от робота до каждой точки территории. Возможен предварительный подсчет всех кратчайших траекторий от каждой точки до каждой, но в случае больших карт это может потребовать большого количества памяти; кроме того, при изменениях на территории патрулирования старые маршруты могут становиться невалидными.

В данной работе рассматриваются два способа нахождения маршрута между точками территории и проведено сравнение эффективности патрулирования при их использовании.

1. Волна (алгоритм Ли): как было описано в разделе 1.1, карта территории разбита на квадратные участки 10×10 ; см, часть из которых является непроходимыми, т.е. содержащими препятствие. Для использования метода “волна” карта представляется в виде графа, где каждому проходимому квадратному участку соответствует вершина, и соседние проходимые участки соединены ребрами. Вес ребер между вершинами, соседствующими по горизонтали или вертикали, равен 1. Вес ребер между вершинами, соседствующими по диагонали, равен $\sqrt{2}$. Поиск маршрута из точки R до точки T производится на указанном графе методом Ли [138]. Обратим внимание, что при использовании подобного метода все силы, воздействующие на робота, будут направлены по одному из восьми направлений в зависимости от того, каким является первый шаг маршрута, соединяющий робота с точкой территории, до которой ищется путь.

Подобный алгоритм позволяет реагировать на изменения на территории патрулирования: при подобном изменении необходимо внести изменения в граф, разомкнув более не соединенные ячейки, убрав из графа проходимые ячейки, которые стали непроходимыми, или добавив те ячейки, которые стали проходимыми (и соединив их с соседними).

2. Алгоритм регионов. Для использования данного метода карту территории необходимо обработать до начала патрулирования и загрузить результаты в память роботов. Этот алгоритм значительно быстрее и представляет собой вариант “предварительного нахождения путей” в компактном представлении. Однако он не может быть использован, если на территории патрулирования происходят изменения.

Рассмотрим алгоритм регионов подробнее. Процесс предварительной обработки карты включает деление территории патрулирования на участки, внутри каждого из которых возможно прямолинейное движение робота между любыми двумя точками. Назовем подобный участок “регионом”. Для каждой двух соседних регионов выбирается точка, лежащая на границе двух регионов – “дверь”. Эта точка используется для движения между регионами, на границе которых она расположена. Нахождение пути между двумя точками происходит следующим образом. Если робот R и точка T находятся в одном регионе, то искомым путем является отрезок, соединяющий точки R и T . Если же точки находятся в разных регионах, то робот ищет путь до желаемой точки T как ломаную, проходящую через некоторую последовательность дверей. Такой маршрут от точки R до точки T может однозначно быть описан последовательностью дверей, через которые он проходит, а его длина найдена с помощью формулы:

$$L(R, T) = L(R, D_1) + L(D_1, D_2) + \dots + L(D_n, T),$$

где $D_1, \dots, D_n$ – последовательность дверей, через которые проходит ломаная.

Последовательность дверей, через которые проходит маршрут между двумя точками, зависит и от регионов, в которых располагаются точки, и от положения точек внутри регионов. Однако при разбиении территории на достаточно мелкие регионы кратчайшие пути для различных точек внутри одного региона практически не отличаются. При обработке карты территории для каждой пары регионов

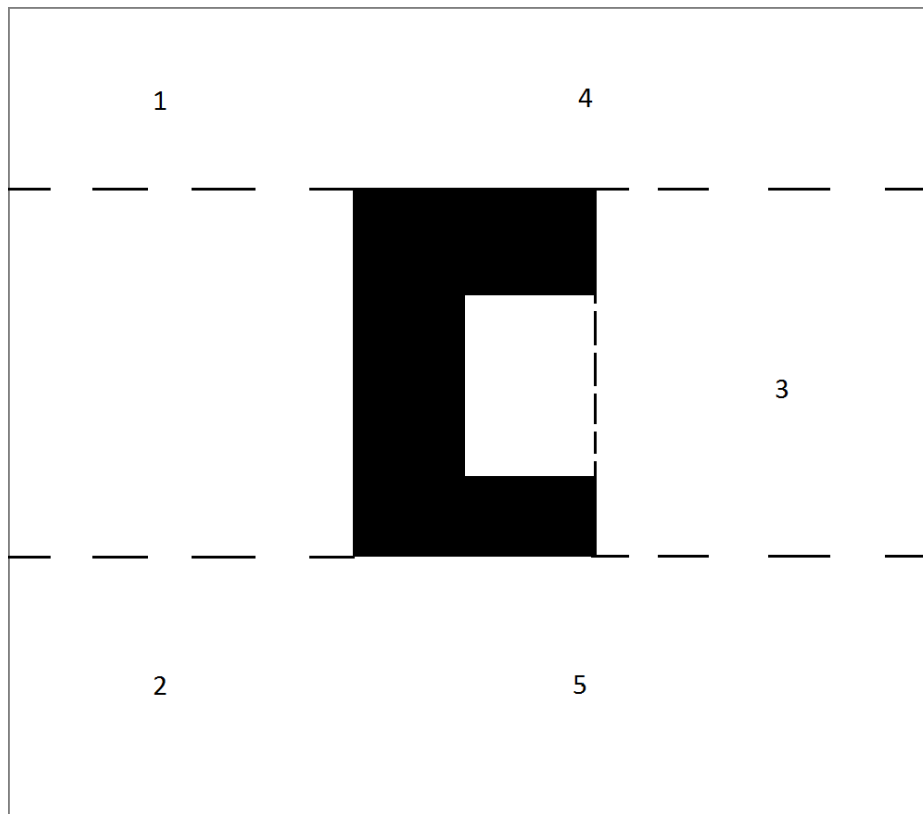


Рисунок 3.2. Карта территории для экспериментов с разбиением на регионы и начальными положения роботов

находится одна или более последовательность дверей, ведущая из первого региона во второй. В случае нескольких маршрутов между двумя регионами, для двух точек оценивается расстояние с помощью каждого из путей, и выбирается путь, который имеет минимальное расстояние.

После составления карты регионов и загрузки ее на роботов процесс поиска пути становится очень быстрым: по любым двум точкам R и T программа находит, в каких регионах они находятся, и выбирает необходимую последовательность дверей, составляющую кратчайший маршрут от R до T . Для сравнения различных методов поиска маршрута используется моделирование патрулирования на территории, приведенной на рис. 3.2. Размеры территории составляют 100 на 88 метров. Стартовые позиции роботов указаны на рисунке, и задаются координатами: (160,100), (160,730), (830,410), (620,100), (620,730).

Результаты моделирования приведены на рисунке 3.3. Как видно из графика, с точки зрения эффективности патрулирования на постоянной, не меняющейся территории метод регионов является предпочтительным, даже при достаточно грубом разбиении. Кроме того, метод регионов обладает требует меньше

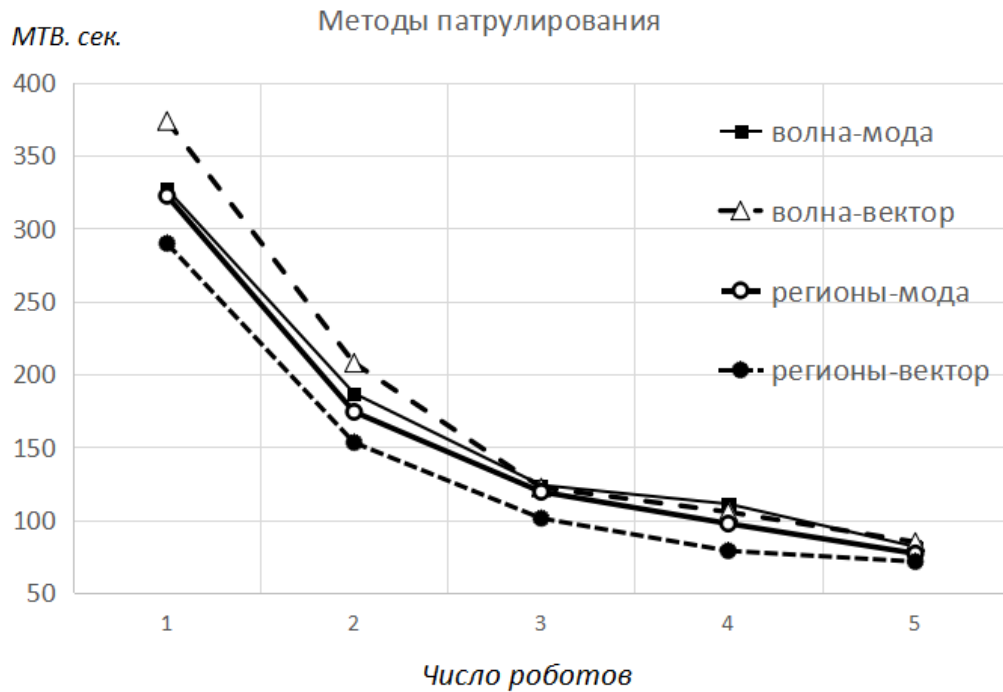


Рисунок 3.3. Сравнение различных методов нахождения пути и выбора направления движения для карты, представленной на рис. 3.2

вычислительных ресурсов. Однако разбиение территории на регионы и составление карты для более сложных территорий (например, см. рис. 3.4 – разбиение такой территории на выпуклые участки будет содержать более 15 регионов) является сложной и трудозатратной задачей. Далее в работе мы используем метод волны, поскольку он может быть применен на меняющихся территориях, а также не требует времязатратной предварительной обработки карты.

Второй эксперимент для сравнения методов выбора направления пути (методы векторной суммы и моды) смоделирован на территории, карта которой изображена на рис. 3.4. Размеры карты составляют 118 на 118 метров. Результаты моделирования приведены на рис. 3.3.

Результаты, приведенные в таблицах 2 и 3 и на рисунках 3.3 и 3.3, демонстрируют, что эффективность выбора направления движения методом векторной суммы падает с увеличением количества препятствий на территории патрулирования. Данный результат можно объяснить следующим образом: при выборе направления движения методом векторной суммы, возникают ситуации, когда две необследованные группы точек “тянут” робота в противоположные стороны, и уравнивают действие друг друга. Более того, векторная сумма может направлять движение робота в сторону препятствия. Рисунок 3.6 демонстрирует

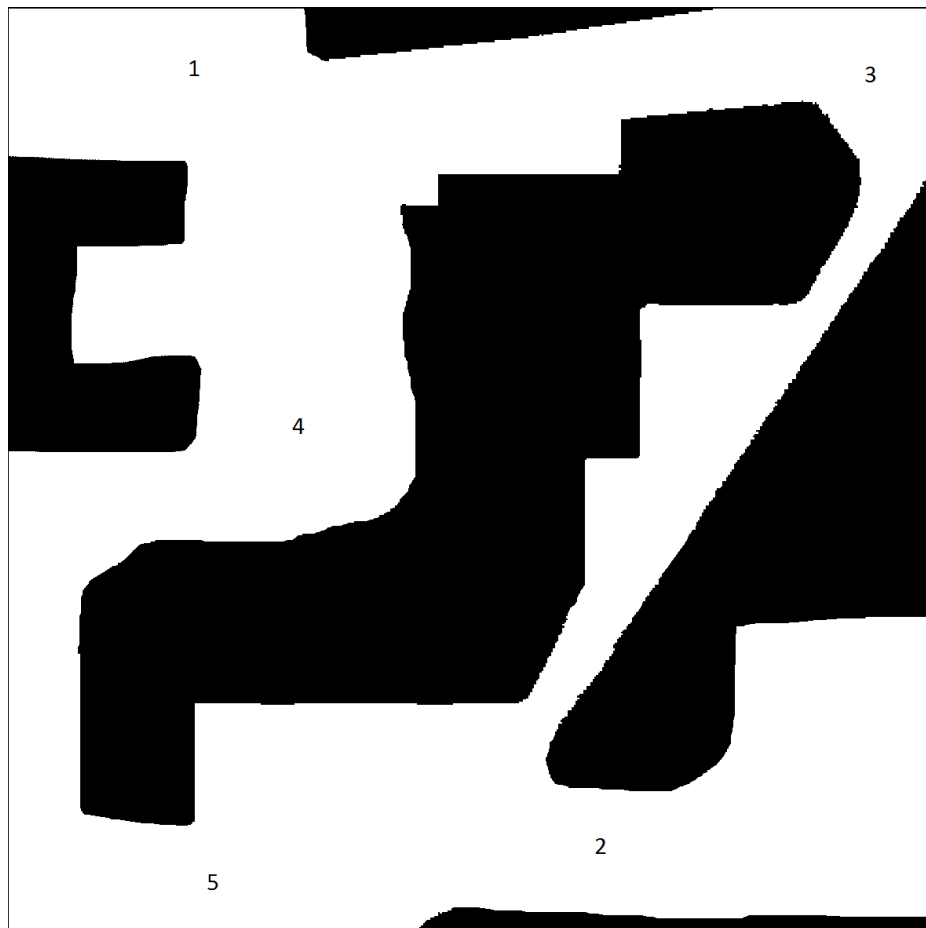


Рисунок 3.4. Карта территории для экспериментов с большим количеством препятствий и отмеченными начальными положениями роботов

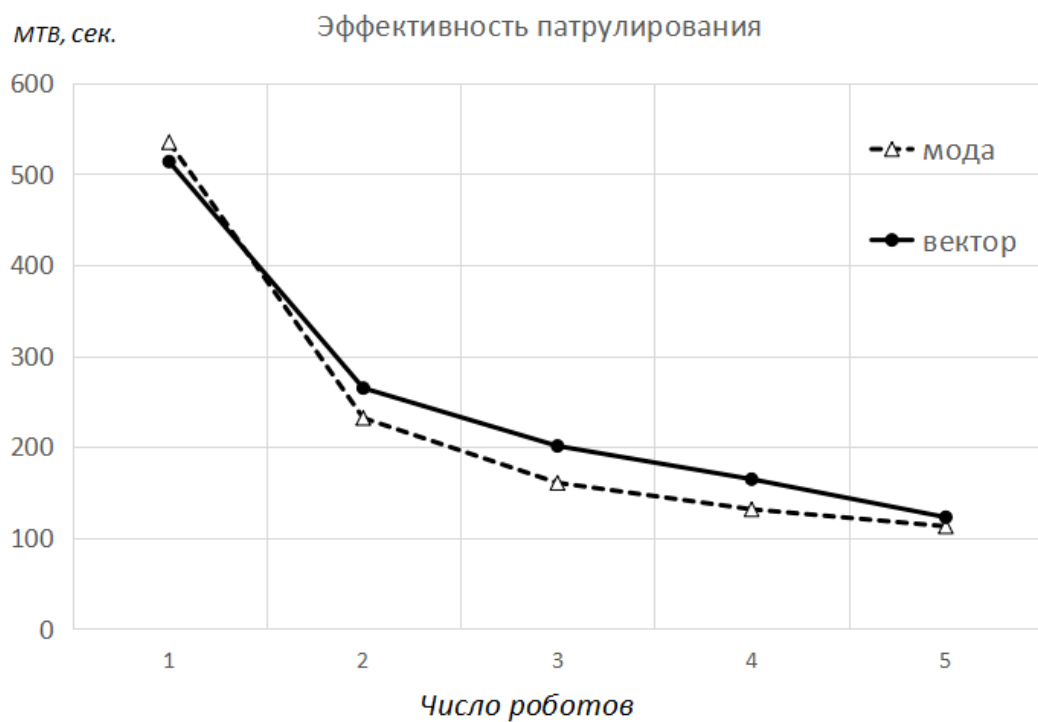


Рисунок 3.5. Сравнение различных методов нахождения пути и выбора направления движения для карты, представленной на рис. 3.4

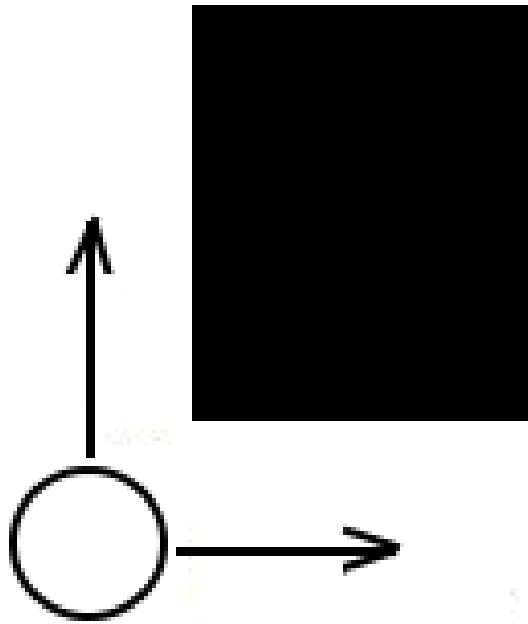


Рисунок 3.6. Иллюстрация неэффективности выбора направления методом векторной суммы при наличии препятствий

ситуацию, при которой две необследованные области территории располагаются сверху и справа от робота. Векторная сумма сил, действующих на робота, направит его в угол препятствия. Данное явление играет тем большую роль, чем больше располагается препятствий на территории патрулирования. Метод моды лишен подобного недостатка.

3.5. Патрулирование территории с препятствиями неголономными роботами

Неголономные роботы не могут менять скорость и направление движения мгновенно, и алгоритм, управляющий такими роботами, должен учитывать эти особенности. Мы предлагаем модифицировать разработанный ранее алгоритм следующим образом: каждый робот вместо того, чтобы двигаться в выбранном направлении, поворачивает колеса таким образом, чтобы как можно скорее ориентироваться по выбранному направлению.

При подобном подходе возникают ситуации, когда дальнейшее движение робота передним ходом невозможно. Будем называть подобные позиции роботов “тупиками”. Для того чтобы выбраться из тупика, роботу необходимо двигаться

задним ходом. При попадании в тупик первоочередной задачей робота является выход из тупика без столкновений с препятствиями и обеспечение возможности дальнейшего движения в требуемом направлении. Мы предлагаем следующий алгоритм выхода из тупика:

1. При входе в режим выхода из тупика робот выбирает направление, в котором планирует направиться после выхода из тупика. Данным направлением является направление, в котором на робота действует максимальное притяжение со стороны точек карты.
2. Робот осуществляет движение задним ходом, повернув колеса таким образом, чтобы угол между его направлением и желаемым направлением уменьшался.
3. Если в процессе движения задним ходом робот оказывается повернут необходимым образом, считается, что он успешно вышел из тупика.
4. Если двигаясь задним ходом робот достигает препятствия и не может продолжать движение, он начинает движение передним ходом, повернув колеса таким образом, чтобы угол между его направлением и желаемым направлением движения после выхода из тупика уменьшался.
5. Если в процессе движения передним ходом робот оказывается повернут необходимым образом, считается, что он успешно вышел из тупика.
6. Если движение передним ходом станет невозможным, он возвращается ко второму пункту алгоритма – движению задним ходом. Робот последовательно выполняет пункты 2–6, пока не окажется повернутым желаемым образом.

Попадание в тупик является крайне нежелательным событием по следующим причинам:

1. В режиме выхода из тупика робот движется на ограниченном участке территории и не участвует в патрулировании.
2. В процессе выхода из тупика робот многократно останавливается, что может быть нежелательно для реальных аппаратов, например, больших транспортных средств, которые испытывают проблемы, чтобы начать движение с места после остановки.

3. При попадании в тупик робот находится очень близко от непроходимого препятствия, и неточности и погрешности в управлении могут привести к столкновению с препятствием.

Поэтому следует минимизировать число ситуаций, когда робот попадает в тупик и вынужден останавливаться или менять направление движения. Мы предлагаем оценивать алгоритм патрулирования не только по эффективности патрулирования, но и по частоте, с которой робот попадает в тупики. Для этого вводится метрика “Доля времени, проведенная в тупике” - *ДВТ*, равная отношению времени, которое робот тратит в режиме выхода из тупика, к общему времени патрулирования.

Для того, чтобы лучше контролировать робота, мы предлагаем контролировать скорость неголономных роботов в зависимости от близости препятствия. Алгоритм выбора скорости должен решать две задачи: обеспечить, что робот движется с достаточно большой скоростью, когда нет опасности столкновения с препятствием; и заранее снижать скорость вблизи препятствий, чтобы успеть осуществить необходимый маневр для избежания столкновения. Если робот использует сонары для обнаружения ближних препятствий, то соблюдение низкой скорости рядом с потенциальными препятствиями (эту информацию робот получает на основе карты территории и оценке собственного положения) позволяет обнаружить препятствия на максимальном радиусе действия сонара и осуществить маневр по избежанию столкновения.

На основе характеристик робота были составлены следующие правила выбора скорости:

1. Если до ближайшего препятствия менее 2 метров, то робот снижает свою скорость до 3 км/ч;
2. Если до ближайшего препятствия от 2 до 5 метров, робот устанавливает скорость в 6 км/ч;
3. Если до ближайшего препятствия от 5 до 7.5 метров, робот устанавливает скорость в 9 км/ч;
4. Если до ближайшего препятствия более 7.5 метров, робот устанавливает скорость в 15 км/ч.

Расстояние до препятствия измеряется не только от передней части робота. В случае, когда робот едет рядом со стеной, скорость робота также необхо-

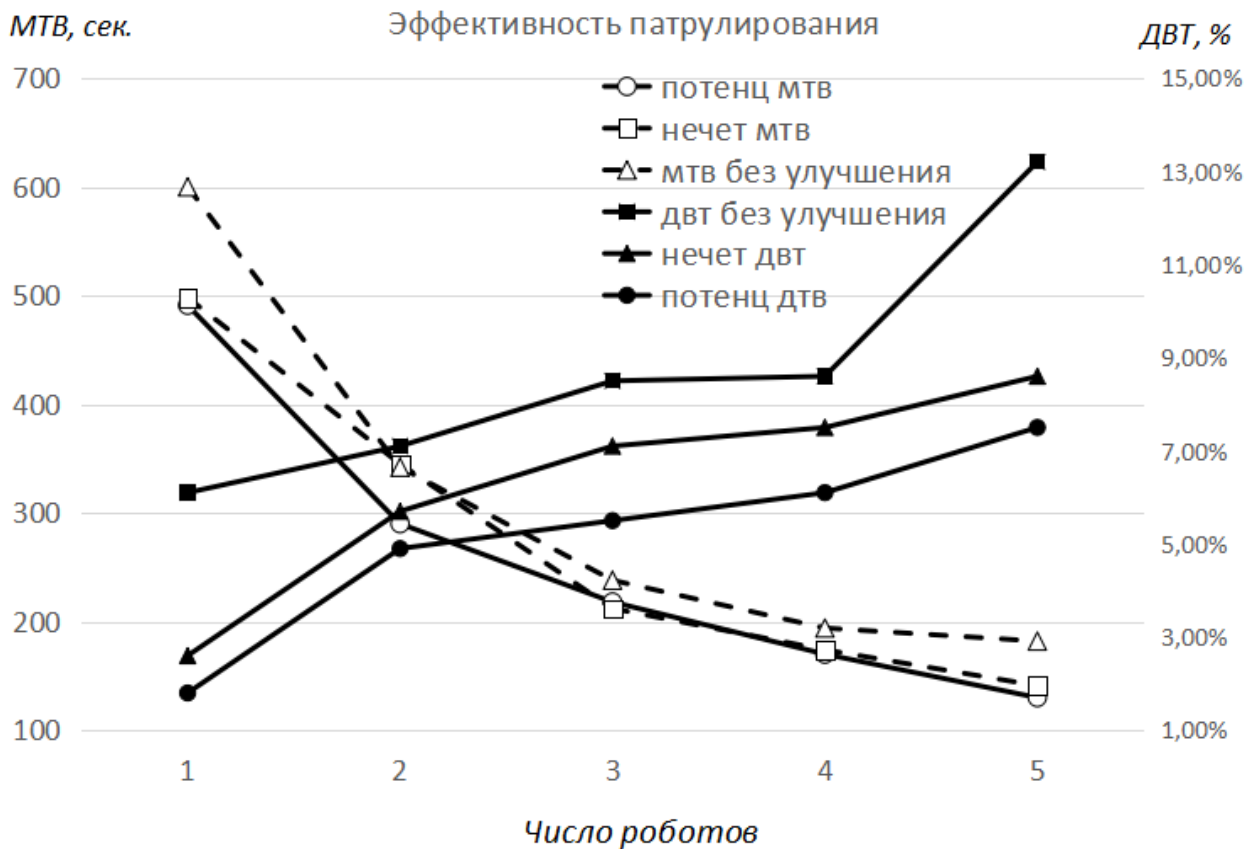


Рисунок 3.7. Сравнение эффективности различных методов избегания препятствий

можно снизить, поскольку потенциальная маневренность робота снижена при наличии рядом препятствия. Предложенные алгоритмы управления роботом (контроль скорости и алгоритм выхода из тупика) достаточны для успешного управления неголономным роботом (при движении с использованием данных алгоритмов робот не окажется в положении, когда его дальнейшее движение невозможно).

Поскольку основной проблемой при патрулировании неголономными роботами является движение вокруг препятствий без столкновений и попадания в тупики, алгоритм управления неголономными роботами исследовался на территории с большим числом препятствий. Как было показано в разделе 3.4, выбор направления движения методом моды показывает лучший результат на картах с большим числом препятствий, поэтому в данном пункте используется только данный метод.

Моделирование проводится на территории, карта которой приведена на рис. 3.4. Результаты моделирования приведены на рис. 3.7 (кривые “без улучшения”). Как видно из графиков, доля времени, проводимого каждым из роботов в режиме выхода из тупика, увеличивается с увеличением числа роботов. Добавле-

ние же новых роботов улучшает эффективность патрулирования тем слабее, чем больше роботов уже ведут патрулирование. Это объясняется тем, что увеличение числа роботов ведет к увеличению случаев, когда два или более роботов патрулируют один участок местности и/или мешают движению друг другу. Проблема падения эффективности патрулирования с увеличением числа роботов из-за их описана в работе [81], однако эффективного метода ее решения не предложено.

3.6. Улучшенные алгоритмы локального выбора направления движения

Несмотря на то, что описанный в предыдущем разделе алгоритм позволяет вести патрулирование неголономными роботами, он является неэффективным: роботы часто попадают в тупиковые ситуации, и им приходится тратить много времени в режиме “выход из тупика”, что снижает время, которое они проводят непосредственно патрулируя территорию.

Рассмотрим несколько алгоритмов, изменяющих локальное движение роботов. Данные алгоритмы призваны корректировать движение роботов таким образом, чтобы робот избегал столкновений и не попадал в тупики, однако не должны значительно изменять траектории движения робота, чтобы глобальная задача по патрулированию местности продолжала выполняться.

Первый механизм, предлагаемый в данном разделе, называется “метод уточнения направления”. На каждом шаге исполнения алгоритма движения роботов метод уточнения направления осуществляет следующие действия:

1. Алгоритм выбора направления движения, описанный в разделе 3.4. определяет направление, в котором должен двигаться робот. Назовем локальной целью движения точку G на расстоянии R_G от робота в выбранном направлении.
2. Робот рассматривает квадратный участок территории K со стороной r и центром в точке G . Локальная цель движения заменяется на точку внутри квадрата K , которая максимально удалено от близких препятствий на территории патрулирования.
3. Робот движется в направлении новой локальной цели.

Для того, чтобы определить, насколько каждая точка T внутри рассматриваемого квадрата K удалена от препятствий, рассматривается квадрат с центром в точке T и стороной $2l$. За каждую занятую точку в этом квадрате точке T начисляется штраф с весом, обратным квадрату расстояния между точкой T и точкой, содержащей препятствие. Точка внутри квадрата K с наименьшим значением штрафа является новой локальной целью. Координаты новой локальной цели задаются выражением:

$$(x, y)_{new} = \underset{(x, y) \in K}{\operatorname{argmin}} \sum_{\substack{\tilde{x} \in [x-l, x+l]; \\ \tilde{y} \in [y-l, y+l]}} \frac{M(\tilde{x}, \tilde{y})}{(x - \tilde{x})^2 + (y - \tilde{y})^2},$$

где $M(x, y)$ равно 1, если точка (x, y) занята или располагается вне территории, и 0 иначе. В случае, если несколько точек на территории обладают одинаковым штрафом, выбирается такая точка, направление на которую ближе к направлению, которое предлагает немодифицированный алгоритм движения.

Идея предложенного алгоритма может быть сформулирована следующим образом: робот будет двигаться в направлении, близком к предложенному методом потенциалов, но при этом будет выбирать немного корректировать траекторию движения, чтобы не приближаться к препятствиям. При проведения экспериментов мы фиксируем отношение $\frac{r}{R_G}$, таким образом, ограничивая максимальное возможное отклонение нового направления на цель от направления на цель до уточнения. На графике, представленном на рис. 3.8, приведены результаты численного моделирования для $r = 0.8R_G$, $l = 10$ метров и различных значений R_G . Моделирование проводилось на территории, карта которой приведена на рис. 3.4 с использованием трех роботов. Как видно из графиков, данный метод позволяет незначительно улучшить значение MTB и уменьшить долю времени, проводимую роботами в тупике, почти вдвое. Далее в данной работе при использовании метода уточнения направлений используются параметры $l = 10$ м, $R_G = 2.5$ м, $r = 2$.

Второй механизм называется “методом ослабления направлений”. Данный механизм выработан на основе методов нечеткой логики [102; 107], хотя сам не использует нечеткую логику в классическом понимании. В отличие от описанного выше метода, который принимает и корректирует полученную цель от “гло-

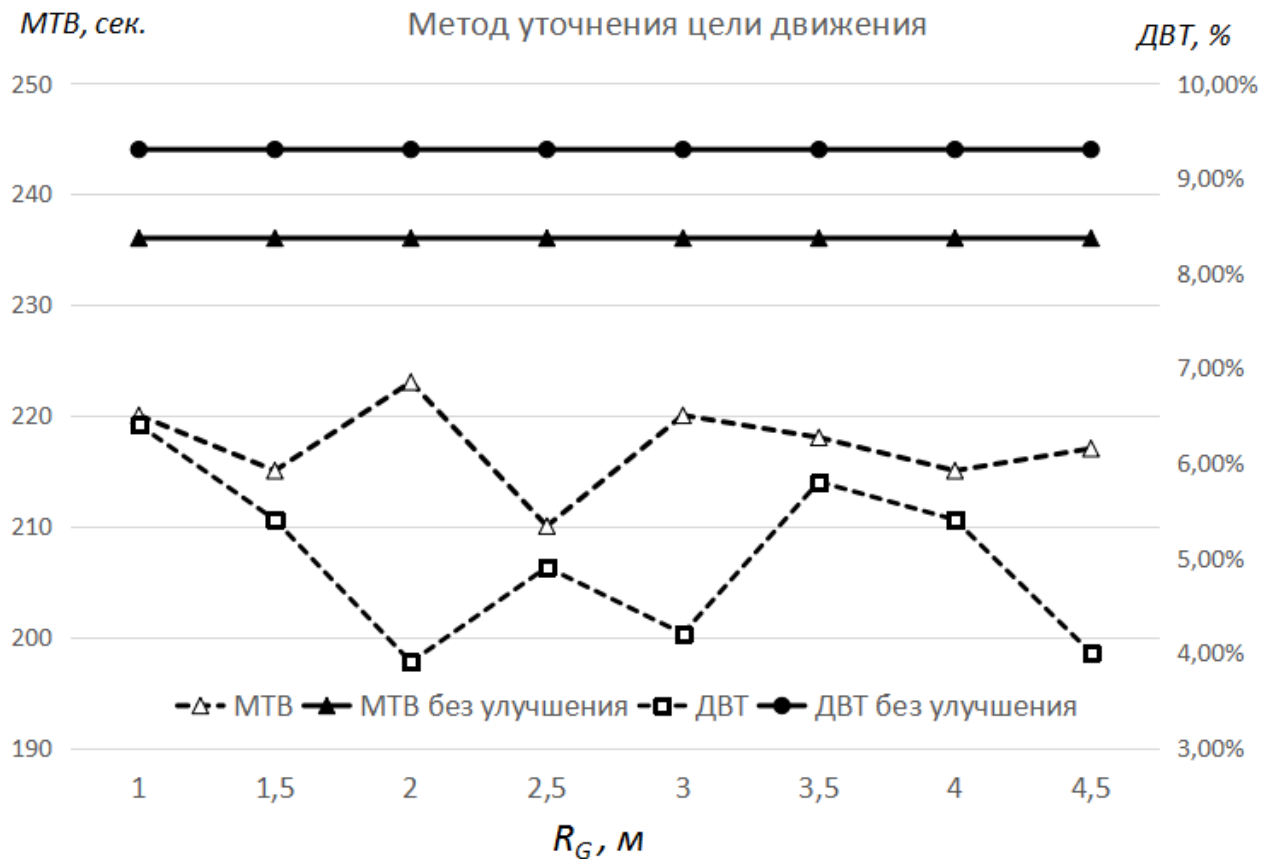


Рисунок 3.8. Работа метода уточнения направления движения

бального” алгоритма движения, метод ослабления направлений используется нами как модификация к алгоритму выбора направления методом моды. Фактически, метод ослабления направлений делает менее привлекательными такие направления движения, при движении в которых велика вероятность столкнуться с препятствием, причем степень ослабления зависит от количества препятствий.

Как описано в разделе 3.4, при выборе направления методом моды, все направления разбиваются на сектора. Силы, действующие на робота, суммируются внутри каждого сектора. Затем робот движется по направлению сектора, сумма сил в котором максимальна. Метод ослабления направлений корректирует значения сил, соответствующих секторам, в зависимости от близости препятствий, которые находятся в этом секторе. Для этого в каждом из секторов рассматривается ближайшее к роботу препятствие, и сумма сил, соответствующая сектору, умножается на коэффициент. Пусть r – расстояние от робота до ближайшего препятствия, находящегося в данном секторе. Тогда сумма сил в этом секторе умно-

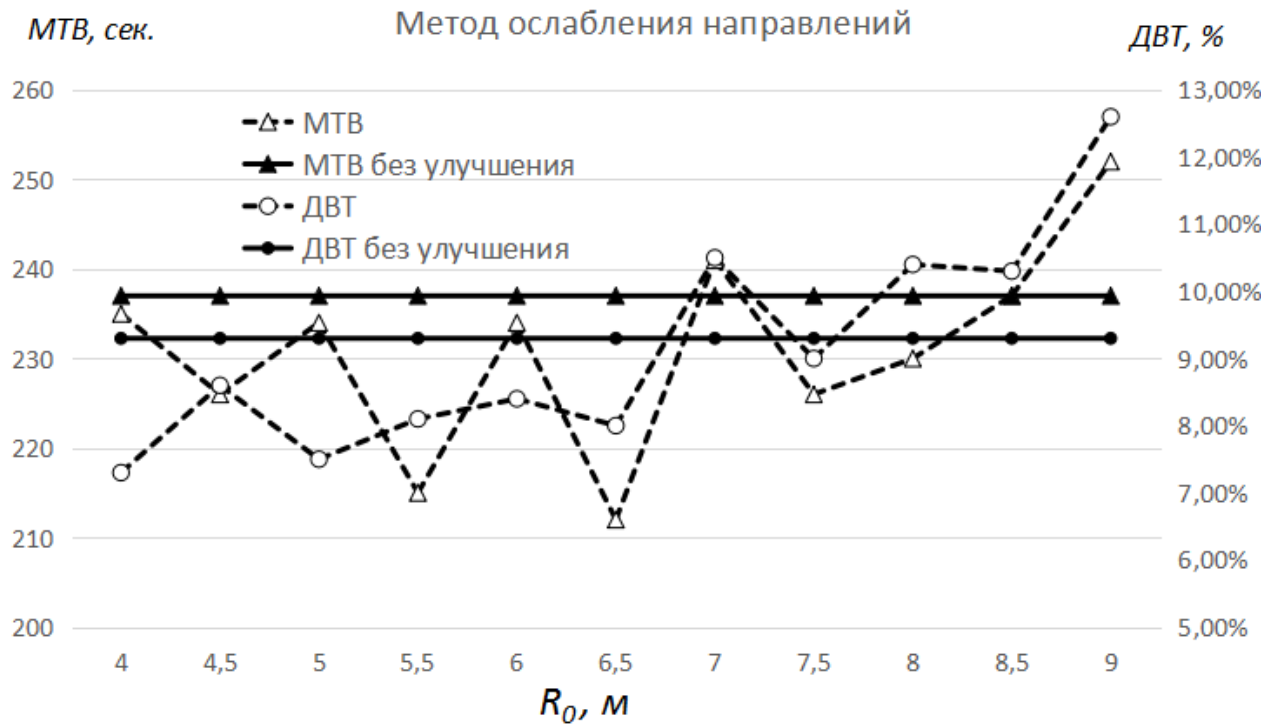


Рисунок 3.9. Работа метода ослабления направлений

жается на коэффициент x_w :

$$x_w = \begin{cases} 1, & r \geq R_0 \\ \frac{r}{R_0}, & r < R_0 \end{cases}$$

Здесь R_0 – подстраиваемый параметр, с увеличением которого увеличивается расстояние, на котором препятствия играют роль на направление движения. Моделирование показало слабую зависимость эффективности метода от параметра R_0 при $R_0 < 6.5$ м и значительное падение эффективности при $R_0 > 6.5$ м. На графике, показанном на рис. 3.9, приведены результаты моделирования на карте (приведенной на рис. 3.4). Моделирование проводилось 3 роботами. Далее при использовании метода нечеткой логики используется параметр $R_0 = 5,5$ метра.

Как следует из графиков, приведенных на рисунках на 3.7 и 3.9, оба предложенных метода (метод уточнения направления и метод ослабления направлений) повышают эффективность патрулирования, однако метод с уточнением цели движения является предпочтительным по обоим рассматриваемым метрикам (МТВ и ДВТ). Наибольший эффект при этом оказывается на метрику ДВТ, а метрика МТВ улучшается лишь незначительно. Это можно объяснить тем, что приведенные ме-

тоды корректируют лишь локальное поведение роботов, помогая избежать попаданий в тупики. Глобальное движение роботов на карте меняется незначительно. Также можно заметить, что часто для уменьшения ДВТ роботы строят менее эффективные маршруты, стараясь избежать столкновений и не выбирая короткие пути, содержащие большее количество препятствий. Таким образом, оценивать оптимальный набор параметров и методов необходимо исходя из параметров и маневренности робота.

3.7. Другие дополнения к алгоритму

Как описано выше, множество проблем, возникающих при патрулировании, связаны с низкой маневренностью роботов, а также со сценариями, когда роботы группируются на небольшой части территории. Проблема маневренности проявляется особенно остро при движении по узким и ограниченным территориям, например, туннелям и перешейкам. Проблема усугубляется, если в подобном туннеле одновременно оказываются два или более роботов: они мешают движению друг друга, уменьшая, таким образом, пространство для маневра. При разрешении подобных ситуаций роботы проводят длительное время в режиме выхода из тупика, снижая эффективность патрулирования.

В данной работе предлагается два способа решения данной проблемы. Первый способ называется “искусственные препятствия”. При использовании данного способа на карту каждого из роботов наносится искусственное препятствие в точке положения каждого из других патрулирующих роботов. Это виртуальное препятствие имеет форму круга с радиусом R_{obst} . При использовании подобного метода карта, которую роботы используют для нахождения путей, будет постоянно меняться. Пути, которые строит робот для движения по карте и вычисления действующих на него сил, будут проходить вокруг созданных искусственных препятствий. На открытой местности подобное нововведение почти не окажет эффекта на патрулирование; однако если какой-либо робот находится в узком перешейке, то созданное механизмом искусственное препятствие полностью перекроет проход для других роботов, и они будут строить свои маршруты таким

образом, чтобы не создавать дополнительных помех роботу, который там находится.

Второй метод называется “метод семафоров”. Для использования этого метода необходима предварительная обработка карты. На карте выделяются участки территории, на каждом из которых нежелательно присутствие более чем одного робота – например, узкие тоннели или маленькие комнаты, где маневренность роботов не позволяет им свободно двигаться. Назовем каждый такой участок “семафором”. Если на территории семафора не находится ни одного робота, но семафор является неактивным. При попадании на территорию семафора робота, он становится активным, и для всех роботов, кроме того, который первый попал на территорию семафора, территория семафора становится непроходимой на карте, используемой для нахождения путей (аналогично искусственным препятствиям, используемым в описанном выше методе). При попадании нескольких роботов на территорию семафора на одном шаге патрулирования, они оба могут продолжать там движение. Это может создать проблемы для проведения маневров, однако такая ситуация возникает редко.

Для исследования эффективности предложенных методов было проведено моделирование на территории, содержащей больше количество узких коридоров. Карта территории приведена на рисунке 3.10. Ее размеры составляют 240 на 159 метров, и карта составлена на основе реального объекта. При моделировании во всех сценариях использовался метод уточнения направления движения.

Результаты моделирования приведены в таблице 4.

		МТВ, сек.			ДВТ, %		
		Без модификаций	Семафоры	Искусств. препятствия	Без модификаций	Семафоры	Искусств. препятствия
Кол-во роботов	1	632.5	632.5	632.5	5.0	5.0	5.0
	2	354.3	340.8	317.7	4.9	6.0	5.9
	3	250.8	249.9	233.3	5.4	6.5	6.5
	4	194.8	196.4	172.7	5.9	6.7	6.5
	5	158.2	153.3	154.6	5.9	6.9	6.8

Таблица 4. Эффективность методов семафоров и искусственных препятствий

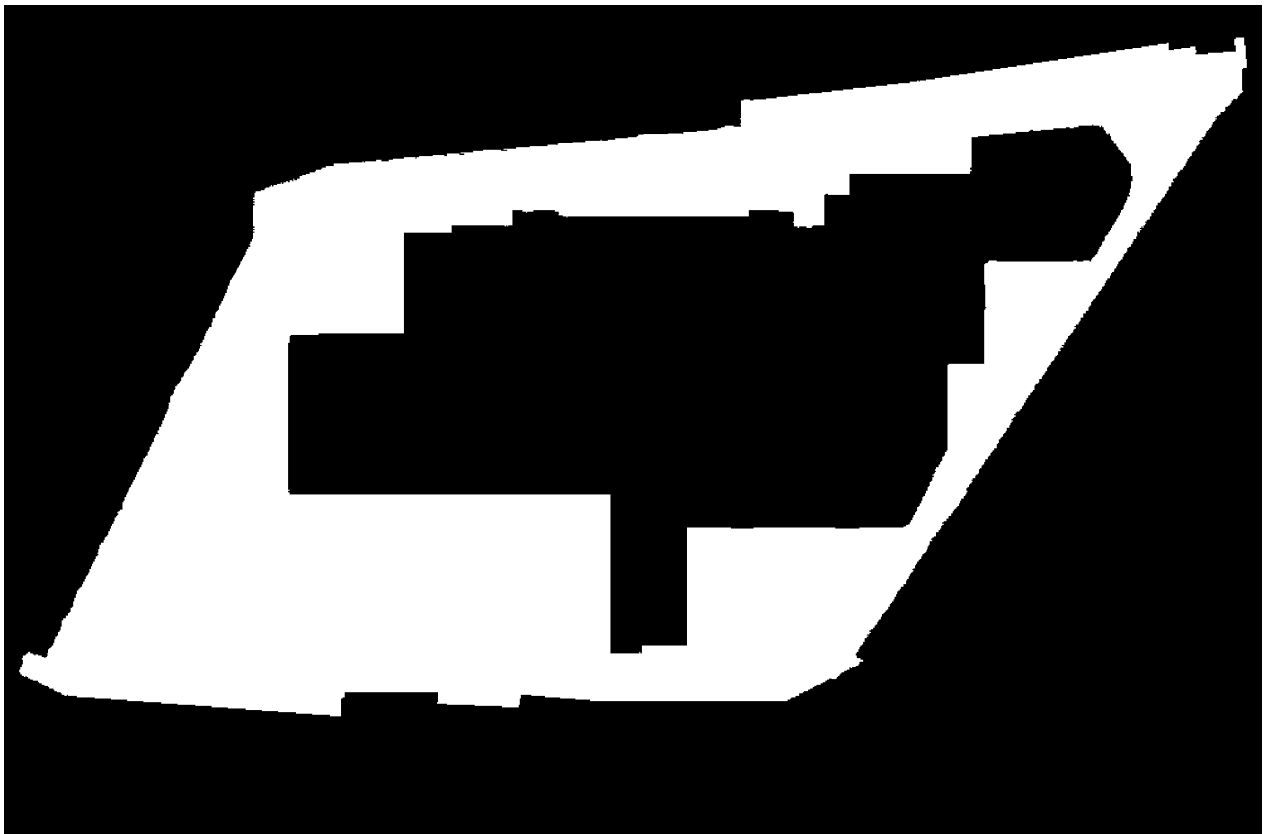


Рисунок 3.10. Карта территории с большим количеством узких коридоров

Как следует из результатов моделирования, метод семафоров практически не повысил качество патрулирования на рассматриваемой территории. Метод с введением искусственных препятствий незначительно ухудшил показатель $ДВТ$, но при этом позволил снизить среднее значение $МТВ$. Неэффективность метода семафоров можно объяснить отсутствием длинных узких участков на территории патрулирования. Из полученных результатов следует нецелесообразность применения метода семафоров, однако, вероятно, он будет полезен на карте, где присутствует большое количество длинных коридоров и будет показывать лучшие результаты на таких картах, чем метод искусственных препятствий: ведь искусственные препятствия, создаваемые роботами, не смогут перегородить всю длину длинного узкого участка. Однако в данной работе преследовалась цель рассмотреть патрулирование на открытых территориях, на территории же с большим числом узких участков возможно, более эффективным будет патрулирование на основе графовых методов.

Неожиданным результатом явилось повышение метрики $ДВТ$ при использовании метода искусственных препятствий. Это можно объяснить тем, что хотя роботы успешно избегают ситуаций, когда несколько роботов сосредоточены на

одном участке карты (что отражается пониженным значением MTB), подобный эффект требует дополнительного маневрирования, что ухудшает метрику $ДВТ$. Также подобное ухудшение может быть вызвано необходимостью резкого маневрирования, когда проход, через который робот планировал осуществить движение, оказывается закрыт другим роботом. Понижение метрики MTB , однако показывает эффективность такого метода.

Целесообразность использования данных методов следует выбирать в зависимости от того, какая территория патрулируется, и того, какая метрика (MTB или $ДВТ$) является более важным для конкретной задачи патрулирования.

3.8. Дилатация “Тумана Войны”

Другая проблема, замеченная при проведении экспериментов – пропускание роботами маленьких давно не посещенных областей. Так, робот может находиться между маленькой давно не посещенной зоной и большой зоной, которая посещалась сравнительно недавно. Из-за того, что не посещенная зона очень мала, робот может выбрать движение во втором направлении. Это является плохим решением, поскольку эффективность алгоритма определяется максимальным значением тумана войны, вне зависимости от площади, и небольшое время, которое робот потратил бы на наблюдение за ближней не посещенной областью, улучшило бы эффективность патрулирования. Простое увеличение значимости TB в формуле 3.1 не является решением, поскольку это может привести к тому, что все роботы будут стремиться посетить одну и ту же точку, что ведет к кучкованию.

Для решения этой проблемы мы предлагаем модифицировать значения TB в каждой точке. Ухудшим значение TB каждой точки, рядом с которой располагается точка с большим значением TB согласно формуле:

$$TB(x,y) = \max_{\substack{\tilde{x} \in [x-d, x+d] \\ \tilde{y} \in [y-d, y+d]}} TB(\tilde{x}, \tilde{y}),$$

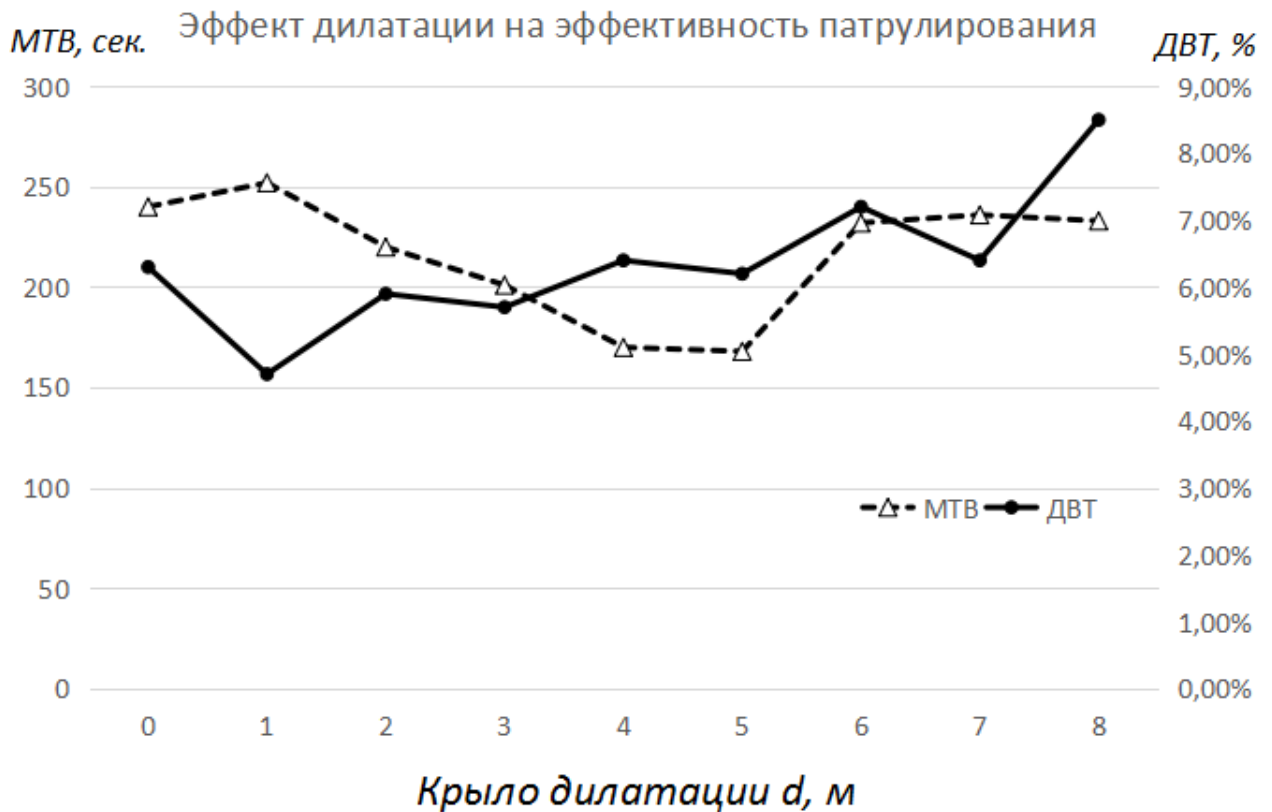


Рисунок 3.11. Влияние дилатации на эффективность патрулирования

Иначе говоря, мы “раздуваем” непосещенные участки территории. Назовем данную процедуру “дилатацией” (т.к. она соответствует морфологической операции дилатации [139]), а параметр d – крылом дилатации. Для вычисления оптимального параметра d было проведено моделирование патрулирования на карте, представленной на 3.10. Патрулирование проводилось с использованием 4 роботов. При моделировании использовался метод уточнения направления движения. Результаты моделирования представлены на рис. 3.11. Как следует из графиков, дилатация повышает эффективность патрулирования, незначительно повышая долю времени, которую роботы проводят в режиме выхода из тупика.

3.9. Работа в условиях плохой сети

В данном разделе исследуется, как разрывы и ограниченность дальности работы сети влияют на эффективность патрулирования. Мы предполагаем, что роботы могут ретранслировать сообщения, т.е. если даже если два робота не имеют

непосредственной связи друг с другом, но существует цепочка из роботов, которая их соединяет, то каждый из них обладает полной информацией о координатах другого.

Мы накладываем два ограничения на сеть: невозможность передавать сообщения дальше чем на расстояние R_{max} и невозможность передавать сообщения, если прямая линия, соединяющая роботов, пересекает препятствие. Также мы рассматриваем сценарий с полным отсутствием связи.

Моделирование проводится для двух территориях патрулирования, приведенных на рисунках 3.2 (пример территории с малым количеством препятствий) и 3.4 (пример территории с малым количеством препятствий). В патрулировании участвуют от 2 до 4 роботов. При моделировании используется метод волны для нахождения путей, метод уточнения направления и оба метода выбора направления движения – метод векторной суммы и метод моды. Результаты моделирования приведены в таблицах 5 и 6.

При моделировании ограниченной связи использовался параметр $R_{max} = 70$ м; при отсутствии связи – параметр $R_{max} = 0$ м. При моделировании полной связи роботы могут общаться даже если прямая, соединяющая их позиции, пересекает препятствие.

Количество роботов	2	3	4
Полная связь, мода	163.8	124	110
Полная связь, вектор	158.4	123.2	106.8
Ограниченная связь, мода	178	131.6	116
Ограниченная связь, вектор	176.6	136.7	110
Нет связи, мода	194.1	159.9	142.9
Нет связи, вектор	195.5	175.3	163.9

Таблица 5. Влияние работоспособности сети на эффективность патрулирования, МТВ, секунды, для территории на рис. 3.2

На основе полученных численных результатов можно сделать следующие выводы:

1. На территории с малым количество препятствий слабая связь или ее отсутствие меньше влияют на эффективность патрулирования.

Количество роботов	2	3	4
Полная связь, мода	268.7	225.6	168.2
Полная связь, вектор	264.3	212.2	174.1
Ограниченная связь, мода	313.9	240	217.5
Ограниченная связь, вектор	289	242.3	217.7
Нет связи, мода	389.1	342.7	339.8
Нет связи, вектор	358	339.7	333.6

Таблица 6. Влияние работоспособности сети на эффективность патрулирования, МТВ, секунды, для территории на рис. 3.4

2. Отсутствие связи сильнее снижает эффективность патрулирования при использовании большего числа роботов. Это объясняется тем, что при отсутствии связи роботы не реализуют коллективное поведение, а действуют индивидуально.
3. Полное отсутствие связи значительно сильнее ухудшает эффективность патрулирования по сравнению с ее ограничением, поэтому даже слабая сеть значительно лучше полного отсутствия связи.

3.10. Выводы к главе

В данной главе:

1. Разработан комплекс программ имитационного моделирования для исследования и сравнения алгоритмов коллективного патрулирования, основанных на методе потенциалов. Комплекс программ позволяет визуализировать, отлаживать и оценивать численные характеристики алгоритмов патрулирования.
2. Разработан и реализован алгоритм коллективного стохастического патрулирования, основанный на методе потенциалов. Алгоритм позволяет осуществлять патрулирование открытой территории коллективом роботов, обеспечивает нерегулярное движение роботов без единого центра управления и без передачи по сети планов о дальнейшем движении каж-

дого из роботов, в том числе с использованием передатчиков, способных работать только на коротких расстояниях. Проведено около 1000 численных экспериментов и вычислены оптимальные параметры алгоритма, приведены рекомендации о модификации алгоритма для повышения его эффективности в зависимости от типа патрулируемой территории.

Глава 4. Построение карты проходимости на основе показаний сонаров

Основное содержание данной главы опубликовано в работах [140; 141].

4.1. Постановка задачи

В данном разделе мы предлагаем два алгоритма, позволяющих восстанавливать карты проходимости. Данные алгоритмы основаны на прямых моделях сонара, и способны обеспечить более точное восстановление, чем стандартный метод, описанный в разделе 1.4. В работе предлагается программа, позволяющая генерировать синтетические показания сонаров на основе карты территории, описанной в виде черно-белого изображения, а также программа для восстановления карты проходимости с использованием различных методов и различных моделей сонара.

Прямая модель сонара описывает, насколько хорошо показание сонара объяснено при заданной карте.

4.2. Прямая детерминистическая модель сонара

Фактически, каждое показание сонара r говорит о следующих вещах: в некоторой области, расположенной на расстоянии примерно r от сонара, располагается один или несколько объектов, которых достаточно, чтобы сонар получил от них отражение на таком расстоянии. Кроме того, в области перед сонаром на расстояниях, меньших r , отсутствуют препятствия в таком количестве, которое бы вызвало показание сонара с меньшим r . Таким образом, чем лучше карта территории удовлетворяет описанным выше условиям, тем лучше она объясняет показание сонара. Введем функцию невязки, которая описывает, насколько показание сонара не согласуется с картой.

Для этого представим карту проходимости M в виде набора переменных x_i . Переменные x_i соответствуют ячейкам карты и принимают значения 0 или 1 – в зависимости от того, занята ли соответствующая ячейка карты или свободна. Искомый набор значений (карта) должен быть таким, чтобы как можно лучше объяснять имеющиеся показания сонаров.

Обозначим Ω_{free} множество переменных x_i , которые соответствуют ячейкам карты, находящимся в области луча сонара, где не должно быть препятствий в соответствии с его показанием; а Ω_{occ} – множество переменных x_i , соответствующих ячейкам карты, которые находятся в области луча сонара, где, согласно показанию, располагается препятствие. Поскольку восстановление карты ведется в пространстве сеток проходимости (то есть территория разбита на ячейки), будем считать, что каждая ячейка либо полностью лежит в области Ω_{free} или Ω_{occ} , либо лежит полностью вне ее – в зависимости от того, принадлежит ли центр ячейки соответствующей области.

Прямая модель, рассматриваемая в данной секции, опирается на следующее предположение: показание сонара зависит только от суммарной взвешенной площади препятствий в областях Ω_{free} и Ω_{occ} . Таким образом, в модели полностью игнорируется взаимное расположение занятых клеток внутри области. На самом деле, большое влияние на показание сонара имеет направление нормали к поверхности препятствия: легче всего обнаружить препятствие, поверхность которого перпендикулярна лучу сонара. Кроме того, сонары легче детектируют одно препятствие, расположенное в его поле зрения, чем несколько препятствий аналогичной формы и суммарной площади, которые рассредоточены в области луча. Описанные эффекты в данной работе не рассматриваются, однако благодаря тому, что предлагаемая модель сонара учитывает зависимость между ячейками карты, восстанавливаемая карта является более качественной по сравнению со стандартными методами.

Перейдем к определению невязки P^j , которая описывает, насколько хорошо карта M объясняет определенное показание сонара z_j . Эта невязка состоит из двух слагаемых: за отсутствие препятствий в области Ω_{free} j -ого сонара отвечает член P_{free}^j , за наличие препятствий в области Ω_{occ} – P_{occ}^j .

Поскольку целью работы является нахождение карты проходимости на основе показаний многих сонаров, то ниже мы будем оценивать, насколько хорошо

карта проходимости объясняет целый набор показаний сонаров. Поэтому мы считаем невязку аддитивной, а все показания сонаров одинаково достоверными и обладающими равным весом. Следовательно, значение невязки должно принимать одинаковые максимальные и минимальные значения для различных сонаров. Мы постулируем, что каждое слагаемое (P_{free} или P_{occ}) принимает значения от 0 до 1.

Рассмотрим, как формируется невязка для области Ω_{free} . Занятые ячейки внутри области Ω_{free} противоречат показанию сонара. Однако, если чувствительность сонара низкая, а площадь ячейки мала, то одна занятая ячейка может не являться достаточным основанием полагать, что показание сонара не объяснено картой: внутри Ω_{free} может содержаться несколько занятых ячеек и при имеющемся показании сонара. Кроме того, не все ячейки в области Ω_{free} являются равнозначными: ячейки, лежащие ближе к основанию луча сонара, играют большую роль, поскольку отражение луча от них более вероятно. Поэтому, чтобы получить оценку эффективного количества помех в области Ω_{free} , мы рассматриваем взвешенную сумму $X = \sum_{x_i \in \Omega_{free}} x_i \omega_i$, где ω_i – вес, определяющий, насколько важной является ячейка x_i . Если сумма X меньше некоторого порога X_0 (который тем больше, чем меньше чувствительность сонара), то показание сонара считается хорошо объясненным. Если же сумма X превышает порог X_0 , то начисляется штраф, увеличивающийся с возрастанием X . Будем считать, что слагаемое P_{free} может быть вычислено с помощью следующего выражения:

$$P_{free}(X) = \begin{cases} \alpha_1 X, & X < X_0 \\ \alpha_1 X_0 + \beta_1 (X - X_0), & X \geq X_0 \end{cases}, \quad (4.1)$$

при этом ставится ограничение: $P_{free}(1) = 1$. Отсюда следует $\beta_2 = \frac{1-\alpha_1 X_0}{1-X_0}$. Описанная модель ниже используется для нахождения карты методом градиентного спуска, поэтому параметр α_1 должен быть достаточно малым, но отличным от нуля: хотя в области $0 \leq X \leq X_0$ показание сонара хорошо объяснено, ненулевой штраф необходим, чтобы метод градиентного спуска мог сойтись.

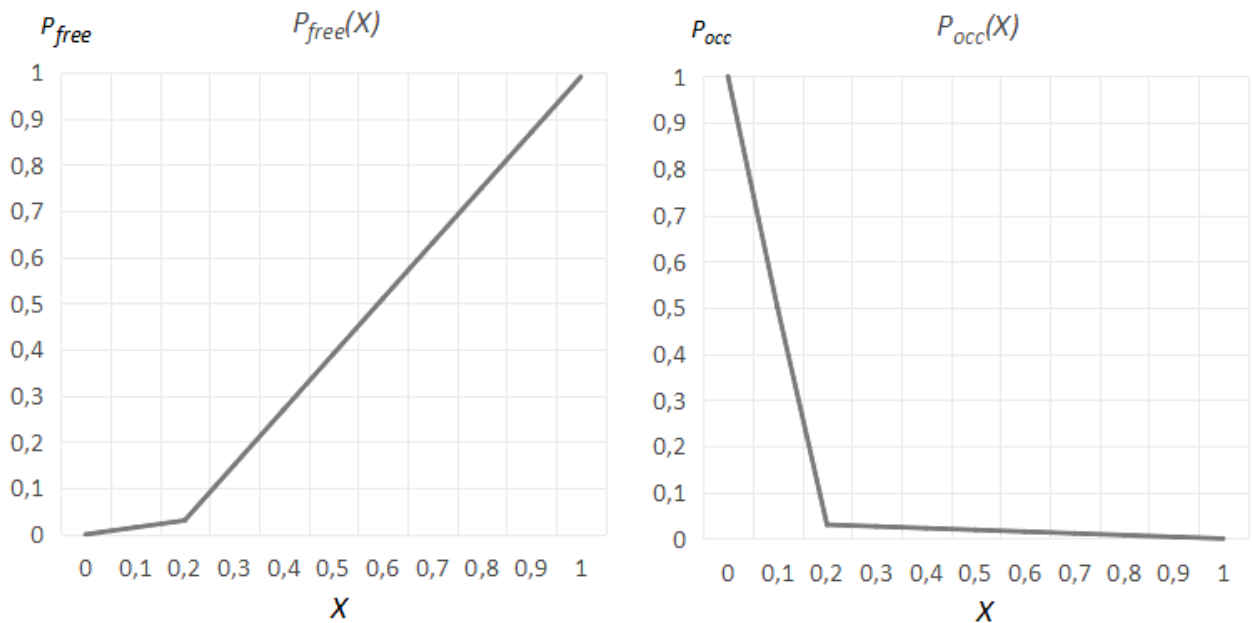


Рисунок 4.1. Примеры графиков функций $P_{occ}(X)$ (слева) и $P_{free}(X)$ (справа) для значения порога $X_0 = 0.2$

Слагаемое P_{occ} вычисляется аналогичным образом: сначала находятся веса ω_i и сумма X , затем невязка находится с использованием выражения:

$$P_{occ}(X) = \begin{cases} 1 - \beta_2 X, & X < X_0 \\ 1 - \beta_2 X_0 - \alpha_2 (X - X_0), & X \geq X_0 \end{cases}, \quad (4.2)$$

при этом ставится ограничение: $P_{occ}(1) = 0$. Показание сонара является хорошо объясненным картой при $X > X_0$, следовательно параметр α_2 должен быть мал, а $\beta_2 = \frac{1 - \alpha_2 + \alpha_2 X_0}{X_0}$.

Для примера, на рисунке 4.1 приведены графики функций $P_{occ}(X)$ (слева) и $P_{free}(X)$ (справа) для значения порога $X_0 = 0.2$.

Для расчета порога X_0 и весов ω_i необходимо знать чувствительность сонара. Обычно информация о чувствительности сонара предоставляется в спецификации от производителя, которая строится следующим образом: используется набор столбов различного диаметра, и каждый из столбов помещается всевозможные точки луча сонара. Таким образом определяется геометрическое место точек для каждого из столбов, в которых он обнаруживается сонаром. Пример полной спецификации, использованный в данной работе, может быть найден по ссылке [142], а пример построенной диаграммы для столбов различного диаметра

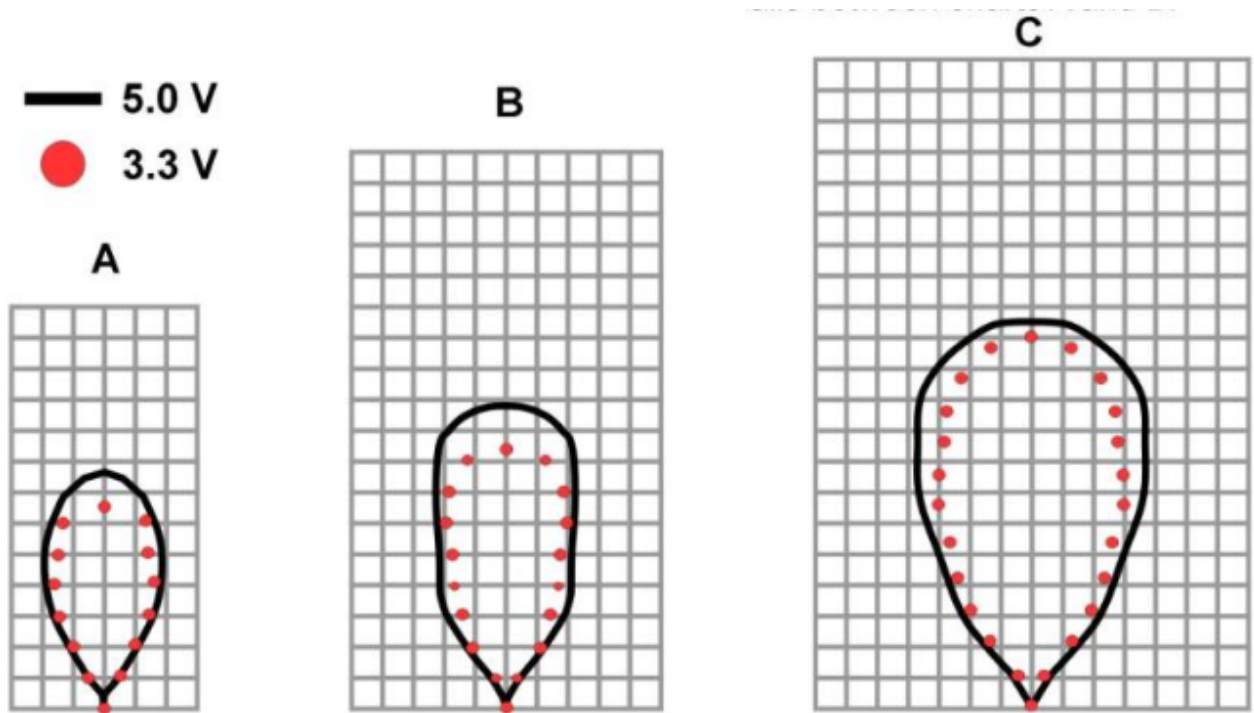


Рисунок 4.2. Пример спецификации реального сонара, сторона клетки 30 см, для разных напряжений. Диаметры используемых столбов: А - 6.1 мм, В - 2.54 см, С - 8.89 см

представлен на рисунке 4.2. В работе мы обобщаем такое описание чувствительности, и предполагаем ее заданной в виде функции $d(x,y)$, которая определяется следующим образом: для каждой точки x,y (координаты которой заданы относительно сонара), функция задает минимальный диаметр столба d , который обнаруживается сонаром в этой точке. Для реального сонара подобная диаграмма чувствительности такая спецификация может быть найдена эмпирически, либо построена приближенно на основе спецификаций.

Необходимо учитывать, что восстановление карты проходимости происходит на дискретной сетке. Пусть площадь одной ячейки сетки равна s . Тогда алгоритм выбора весов ω_i и порога X_0 следующий:

1. Сначала происходит расчет предварительных весов: для каждой ячейки x_i выбирается вес $\omega_i^0 = \min(\frac{s}{d(x,y)}, 1)$. То есть вес равен отношению площади клетки к площади, которая должна располагаться в этой точке, чтобы сонар смог обнаружить препятствие, но если площадь ячейки больше, чем площадь, необходимая для обнаружения, вес все равно берется равным 1 (поскольку минимальный размер препятствия на дискретизованной карте не может быть меньше размера ячейки).

2. При таких весах, сонар обнаружит препятствие, если сумма $X = \sum_{x_i \in \Omega_{free}} x_i \omega_i^0$ будет не меньше 1. В самом деле, ω_i^0 для каждой ячейки показывает, насколько эта ячейка заполняет “квоту”, необходимую для обнаружения препятствия.
3. При таких весах однако, невязка, вычисленная с помощью формул (4.1, 4.2), будет иметь различный диапазон значений для различных показаний сонаров. Как было описано выше, мы хотим, чтобы значения функции P_{free} и P_{occ} были заключены между 0 и 1, а следовательно, согласно (4.1, 4.2), и сумма X должна принимать значения от 0 до 1. Для этого мы проводим нормировку: $\omega_i = \frac{\omega_i^0}{\sum \omega_i^0}$. Порог X_0 тогда уменьшается в пропорционально число раз:

$$X_0 = \frac{1}{\sum_i \omega_i^0}$$

4.3. Вероятностная прямая модель

Рассмотрим прямую модель сонара $p(r|M)$, способную моделировать случайное отсутствие отражения от препятствий небольшого размера, случайные выбросы в показания сонара, а также неточность показаний. Модель предназначена для работы с сетками проходимости и была впервые использована в работе [119].

Фактически, модель опирается на следующую логику работы сонара: с вероятностью p_{error} (обычно достаточно малой) показание сонара является случайным, принимающим случайное значение от минимального R_{min} до максимального R_{max} показаний сонара. В большинстве случаев сонар получает не случайное показание – это происходит с вероятностью $1 - p_{rand}$. В этом случае рассмотрим ближайшее к сонару препятствие, которое располагается в его луче. С вероятностью p_{hit} , сонар отразился именно от этого препятствия, и показание сонара задается нормальным распределением с центром в p_{hit} и некоторой дисперсией σ . В противном случае, с вероятностью $1 - p_{hit}$, луч сонара дошел до следующего препятствия, от которого он также отразился с вероятностью p_{hit} , и так далее. Если луч сонара не отразился ни от одного препятствия, то выдается максимальное показание R_{max} .

При использовании модели каждая ячейка карты считается препятствием. С использованием такой модели препятствие, покрывающее несколько ячеек карты проходимости, имеет больший шанс отразить луч сонара, чем препятствие, состоящее из одной ячейки. Пример построенной характеристики $p(z|M)$ на основе окружающих препятствий с использованием этой модели приведен на рисунке 4.3.

Рассмотрим, как в явном виде записывается распределение $p(z|M)$ показания сонара при заданной карте M . Пусть в области луча сонара располагается K препятствий. Обозначим расстояния до них в возрастающем порядке: $D = \{d_1, d_2, \dots, d_K\}$. Идеальный сонар, который не подвержен случайному шуму ($p_{rand} = 0$), и который надежно обнаруживает ближайшее препятствие ($p_{hit} = 1$), всегда будет давать показание z_1 .

Для описания модели, введем переменные, описывающие, от какого препятствия получено отражение ультразвука, вызвавшее показание сонара: $C = \{c_*, c_0, c_1, \dots, c_K\}$. Каждая из этих переменных принимает значения 0 или 1, причем только одна переменная из набора C равна 1. Если $c_i = 1, 1 < i \leq K$, то считается, что показание сонара вызвано препятствием с d_i . Если $c_* = 1$, то показание вызвано случайным шумом. Если $c_0 = 1$, то сонар не обнаружил ни одного препятствия, то есть получил показание $r = R_{max}$.

Можно записать условные распределения $p(z|M, C)$, если известно, каким препятствием вызвано показание сонара:

$$\begin{cases} p(z|M, c_i = 1) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \frac{(z-d_i)^2}{\sigma^2}}, 1 < i \leq K \\ p(z|M, c_0 = 1) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \frac{(z-R_{max})^2}{\sigma^2}} \\ p(z|M, c_* = 1) = \frac{1}{z_{max}} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \ln \frac{R_{max}^2}{2\pi\sigma^2}} \end{cases}$$

Если свести приведенные выше равенства в одно выражение, то получим:

$$p(z|M) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} [c_0 \cdot \frac{(z-R_{max})^2}{\sigma^2} + \sum_{i=1}^K c_i \cdot \frac{(z-d_i)^2}{\sigma^2} + c_* \cdot \ln \frac{R_{max}^2}{2\pi\sigma^2}]} \quad (4.3)$$

С помощью выражения (4.3) можно получить вероятностное распределение показаний сонара, при условии, что известно соответствие c , то есть известно, какая из величин $\{c_*, c_0, c_1, \dots, c_K\}$ принимает значение 1. В реальных задачах та-

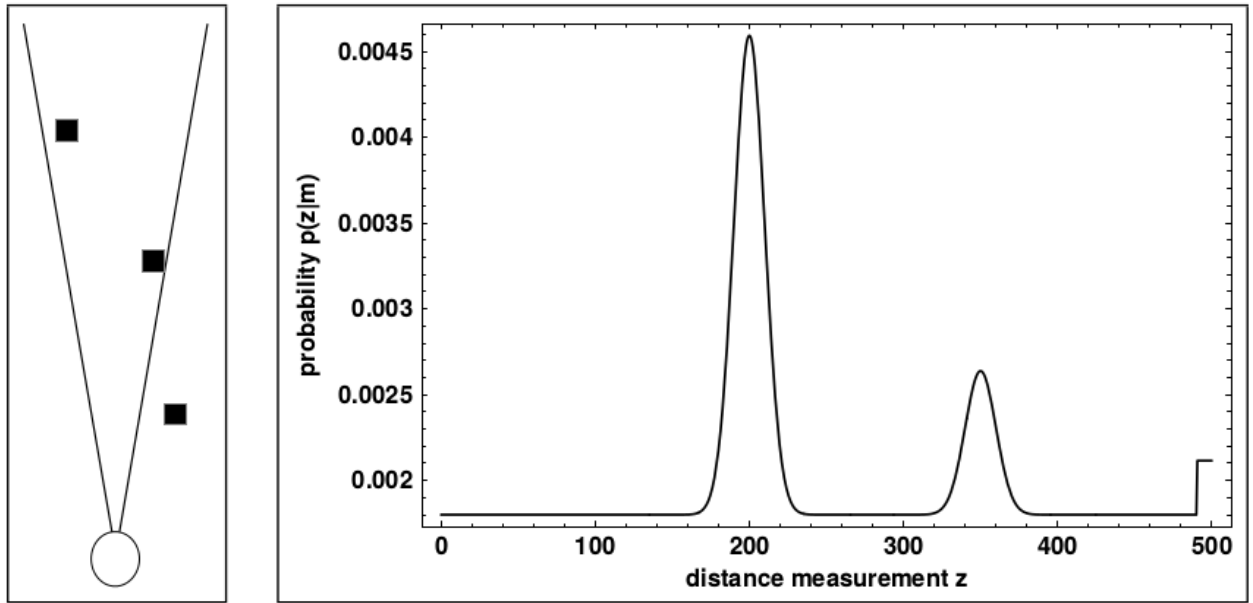


Рисунок 4.3. Пример прямой вероятностной модели сонара. Слева изображена карта с тремя препятствиями, два из которых находятся в луче сонара. Справа показано вероятностное распределение показаний сонара $p(z|M)$ для данной ситуации. Пики функции $p(z|M)$ соответствуют расстоянию до препятствий [119]

кой информации нет; однако можно вычислить вероятность с помощью формулы Байеса:

$$p(z|M) = \sum_{c \in C} p(z|M, c) p(c|M) = \sum_{c \in C} p(z|M, c) p(c),$$

где $p(c)$ – вероятность получения того или иного соответствия, то есть, как описано выше:

$$p(c) = \begin{cases} p_{rand} & c_* = 1 \\ (1 - p_{rand})(1 - p_{hit})^K & c_0 = 1 \\ (1 - p_{rand})(1 - p_{hit})^{i-1} & c_i = 1 \end{cases}$$

4.4. Стохастический метод восстановления карты

Первый предлагаемый в данной работе метод основан на вероятностной модели сонара, описанной в разделе 4.3. Метод осуществляет поиск в пространстве

бинаризованных карт M (то есть значение $m_{x,y}$ каждой ячейки принимает значения 0 или 1) такой карты, которая максимально хорошо объясняет имеющиеся показания сонаров. Рассмотрим этот метод подробнее.

4.4.1. Восстановление карты

Состояние карты проходимости при начале работы алгоритма – пустое, то есть значения проходимости всех клеток равны 0. Один шаг алгоритма заключается в (i) инвертировании значения в одной из ячеек карты проходимости и (ii) принятия решения о сохранении измененного значения или возврата ячейки в начальное состояние. Ячейка для инвертирования выбирается случайно. Для принятия решения о сохранении инвертирования используется потенциал Φ карты, задаваемый выражением

$$\Phi = \phi_{sonars} + \phi_{occ} + \phi_{borders}. \quad (4.4)$$

Если инвертирование ячейки уменьшило значение Φ , то значение ячейки сохраняется инвертированным; в противном случае, инвертирование сохраняется с малой вероятностью p_{rand} . Подобный случайный переход необходим для избегания “застревания” в локальных минимумах.

Рассмотрим выражение (4.4) и слагаемые, из которых состоит потенциал Φ . Первое слагаемое ϕ_{sonars} показывает, насколько хорошо карта M объясняет показания сонаров и вычисляется согласно: каждое измерение z_i сонара оценивается с точки зрения соответствия карте проходимости. Реальные территории обычно содержат несколько препятствий размеров больше, чем одна ячейка сетки проходимости (например, в помещении это могут быть коробки, столы и т.д.). Поэтому логично потребовать, чтобы восстанавливаемая карта проходимости также состояла из сгруппированных занятых ячеек, а не занятых. Например, пустая ячейка, окруженная со всех сторон занятыми, вообще не может быть обнаружена сонаром. Кроме того, реальные территории, где возможно движение робота, содержат не так много препятствий. Для того, чтобы восстанавливаемая кар-

та территории обладала такими же свойствами, мы вводим слагаемые регуляризации ϕ_{occ} и $\phi_{borders}$: слагаемое $\phi_{borders} = \sum_{x,y} \omega_{borders} N_b^2(x,y)$ вводит штраф для каждой ячейки карты, пропорциональный квадрату числа границ N_b между проходимой и непроходимой областью у клетки с координатами (x,y) . Потенциал $\phi_{occ} = \sum_{x,y} \omega_{occ} Occ(x,y)$, где $Occ(x,y) = 0$, если ячейка (x,y) проходима, и $Occ(x,y) = 1$, если занята, вводит штраф за проходимость область: мы предполагаем, что большая часть территории является проходимой, поэтому каждая ячейка является с большей вероятностью свободной, чем занятой. Весовые коэффициенты $\omega_{borders}$ и ω_{occ} позволяют регулировать относительную значимость условий регуляризации и измерений сонаров.

Рассмотрим эффективный способ расчета потенциала Φ на каждом шаге алгоритма. При инвертировании какой-либо ячейки карты изменяется потенциал только тех слагаемых суммы ϕ_{sonars} , для которых ячейка находится в поле зрения. Поэтому, чтобы быстро подсчитывать значение потенциала на каждом шаге алгоритма, мы храним вспомогательную таблицу, где для каждой ячейки карты хранятся номера измерений показаний сонаров z_i , которые содержат эту ячейку в поле зрения. Это позволяет проводить оценку изменения потенциала на каждом шаге алгоритма достаточно быстро: необходимо пересчитать лишь малую часть слагаемых из ϕ_{sonars} , потенциал $\phi_{borders}$ и ϕ_{occ} зависят вообще только от локальной конфигурации карты.

При работе в режиме реального времени оптимизация ведется одновременно с получением новых данных. Обратим внимание, что если робот движется достаточно долго и непрерывно собирает данные, то в определенный момент количество показаний сонаров превысит объем, который он способен обрабатывать в режиме реального времени. Поэтому необходимо выбирать лишь часть измерений, которые будут использоваться при оптимизации карты. Мы предлагаем следующие альтернативные способы выбора измерений сонаров, используемых для оптимизации:

1. Для оценки карты по существующим показаниям в оффлайн-режиме — использовать все измерения.
2. Для оценки в реальном времени — использовать скользящее окно и рассматривать последние N измерений.

3. Для оценки в реальном времени – фильтровать измерения пространственно: как описано выше, для каждой ячейки хранятся номера измерений сонаров, которые содержат эту ячейку в поле зрения. Ограничение число привязанных к каждой ячейке измерений значением N_{max} гарантирует, что один шаг алгоритма займет не больше некоторого времени $t_{max} = N_{max}t_{sonar} + t_{occ} + t_{border}$, где t_{sonar} – время, необходимое для пересчета одного слагаемого суммы ϕ_{sonars} , а t_{occ} и t_{border} – времена, необходимые для оценки потенциалов ϕ_{occ} и $\phi_{borders}$ соответственно. Для выбора N_{max} измерений, содержащих выбранную ячейку, мы предлагаем выбирать N_{max} самых свежих измерений.

4.4.2. Численные эксперименты

При исследовании алгоритма стохастического восстановления были использованы следующие параметры настроек:

$$\begin{aligned}
 p_{rand} &= 0.001, \\
 p_{hit} &= 0.35, \\
 R_{max} &= 3.5 \text{ м}, \\
 \omega_{borders} &= 0.05, \\
 \omega_{occ} &= 0.25
 \end{aligned} \tag{4.5}$$

Как видно из описанной модели, мы считаем вероятность получения случайного измерения p_{rand} очень низкой.

Исследование метода стохастического восстановления проводилось на искусственных данных. Две карты, с которых были сняты синтетические данные, приведены на рисунке 4.4. Синтетические данные для экспериментов снимались с использованием прямой вероятностной модели сонара 4.3. Снятие показаний осуществлялось при виртуальном проезде робота по коридору между сериями коробок. Положение сонаров на роботе приведено на рисунке 4.7.

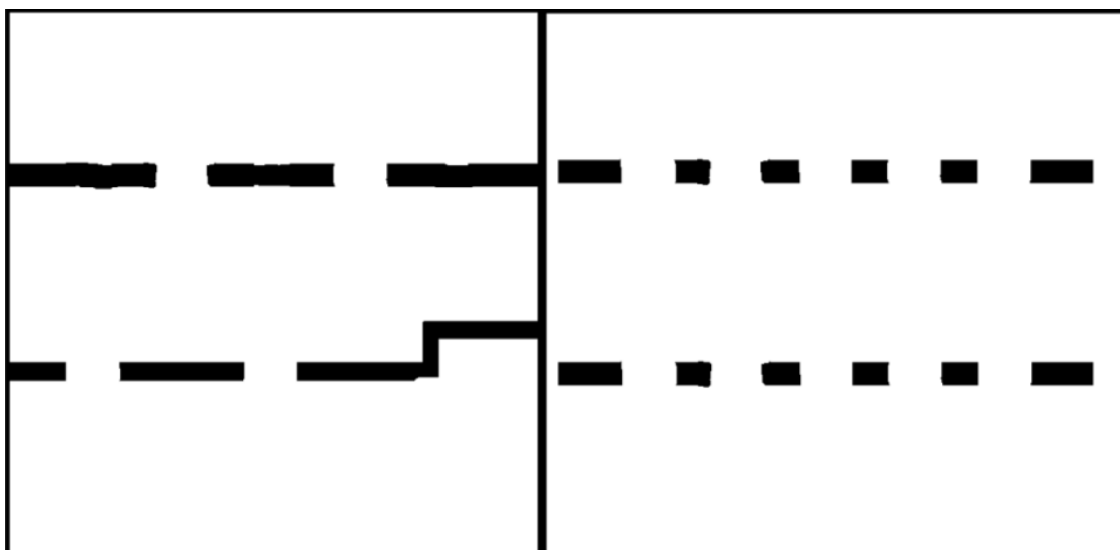


Рисунок 4.4. Карты территорий, использованные для генерация синтетических данных

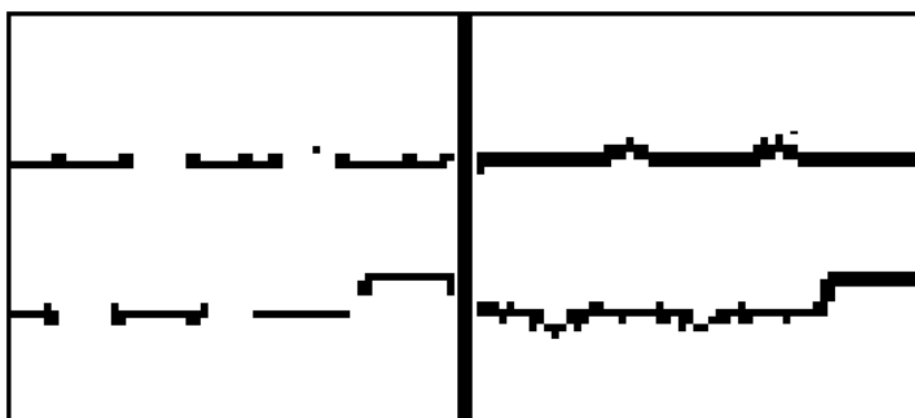


Рисунок 4.5. Результаты восстановления карты проходимости на основе синтетических данных методом стохастического градиента (слева) и традиционным методом (справа)

Результаты восстановления с использованием стохастического и классического методов восстановления приведены на рисунках 4.5 и 4.6 (при этом карта - результат восстановления классическим методом бинаризована). Как видно из приведенных рисунком, стохастический метод восстановления способен более точно восстановить подобную картину.

Важно отметить, что такой алгоритм восстанавливает “минимальную” карту, то есть такую, которая объясняет показания сонаров, но содержит минимальное количество занятых ячеек. Этим объясняется некоторая неровность поверхностей препятствий.

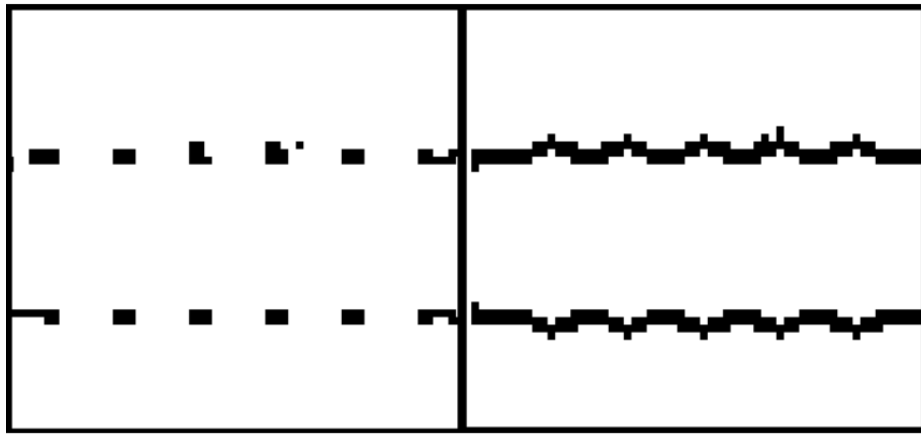


Рисунок 4.6. Результаты восстановления карты проходимости на основе синтетических данных методом стохастического градиента (слева) и традиционным методом (справа)

4.5. Метод градиентного спуска

Метод градиентного спуска использует детерминированную прямую модель сонара, описанную в разделе 4.2. Ячейки карты проходимости принимают непрерывные значения.

4.5.1. Восстановление карты

В разделе 4.2 введена невязка P^j , описывающая насколько плохо показания сонара z_j объяснены картой территории (чем хуже согласованы показания сонаров и карта, тем больше невязка). Метод, описанный в данном разделе, минимизирует сумму невязок P^j , используя метод градиентного спуска.

Обратим внимание, что описанная в разделе 4.2 модель предназначена для карт проходимости, где переменные x_i принимают значения только 0 и 1. Оптимизация без ограничений на значения x_i может привести к возникновению значительных искажений на восстановленной карте в случаях, когда переменные x_i принимают значения больше 1 или меньше 0 или значения, близкие к 0.5. Поэтому, для того, чтобы значения x_i были близки к 0 и 1, мы вводим дополнительные

члены регуляризации в сумму S . Похожий способ борьбы с артефактами может быть найден, например, в статье [143].

Чтобы заключить значения проходимости ячеек построенной карты между 0 и 1 вводится регуляризация:

$$R_1(x_i) = \begin{cases} c_1 \cdot (x_i - 1), & x_i > 1 \\ 0, & 0 \leq x_i \leq 1 \\ -c_1 x_i, & x_i < 0 \end{cases},$$

Чтобы значения проходимости принимали значения, близкие к 0 и 1, вводится слагаемое, которое штрафует за близость значения переменной к 0.5:

$$R_2(x_i) = c_2 |x_i - 0.5|$$

Доопределим производные функций:

$$P'_{free}(X_0) = P'_{occ}(X_0) = 0,$$

$$R'_1(0) = R'_1(1) = 0,$$

$$R'_2(0.5) = 0$$

Поскольку $R'_2(0.5) = 0$, то значения ячеек клеток, которые не попали в луч ни одного сонара, останутся равными 0.5.

Таким образом, задача построения оптимальной карты переходит в задачу минимизации следующей суммы:

$$S = \sum_{z \in Z} \left(P_{free}(\sum_{x \in \Omega_{free}(z)} M(x)) + P_{occ}(\sum_{x \in \Omega_{occ}(z)} M(x)) \right) + \sum_{x \in \Omega_M} (R_1(M(x)) + R_2(M(x))),$$

здесь Ω_M – множество ячеек карты, $M(x)$ – значение проходимости ячейки x , Z – множество показаний сонаров z , $\Omega_{free}(z)$ и $\Omega_{occ}(x)$ – множества ячеек, которые попали в луч сонара при измерении z , и которые, согласно этому показанию, не могут ($\Omega_{free}(x)$) и могут содержать ($\Omega_{occ}(x)$) препятствия.

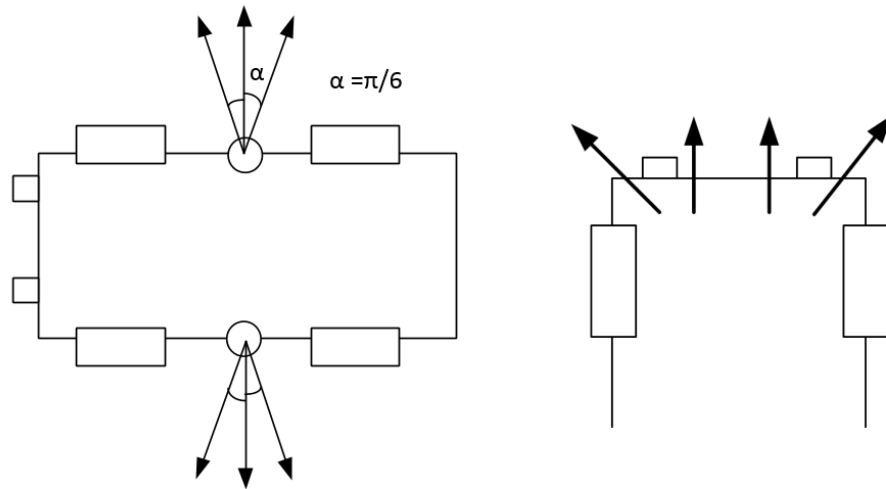


Рисунок 4.7. Расположение сонаров на роботе для эксперимента с использованием синтетических (слева) и реальных (справа) данных

После получения карты проходимости, производится бинаризация значений с порогом разделения 0.5 (на практике значения ячеек будут близки к 0 или 1 из-за слагаемых регуляризации). В данной работе предложенный алгоритм использовался только для обработки уже собранных данных. Скорость работы алгоритма позволяет использовать его в реальном времени, однако для работы в таком режиме (и для работы в течение неограниченного времени) необходимо осуществлять фильтрацию измерения, например, с помощью временного окна.

4.5.2. Численные эксперименты

Для проведения экспериментов были выбраны следующие численные параметры:

$$\begin{aligned}
 c_1 &= 10.0 \\
 c_2 &= 1.0 \\
 \alpha_1 &= 0.05 \\
 \alpha_2 &= 0.001
 \end{aligned}
 \tag{4.6}$$

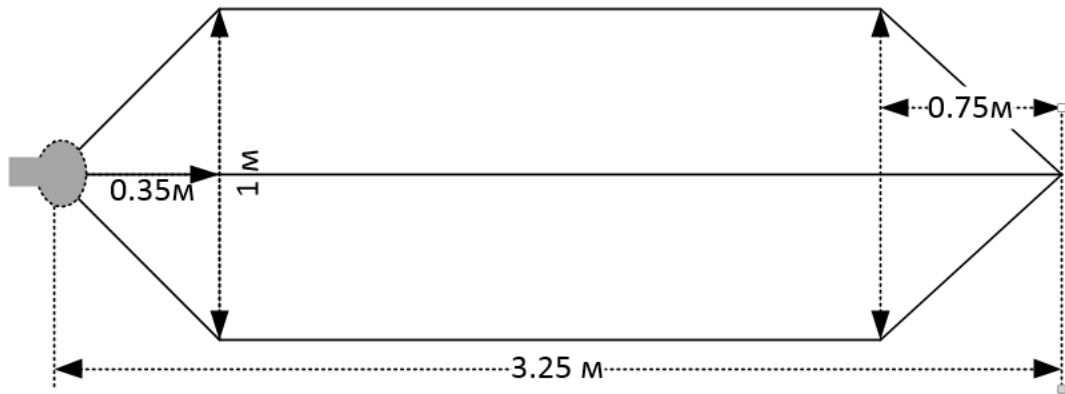


Рисунок 4.8. Диаграмма направленности сонаров, использованных в экспериментах

Форма луча (диаграмма направленности) сонара, использованного в численных экспериментах, приведена на рисунке 4.8. Области Ω_{free} и Ω_{occ} определяются следующим образом: область Ω_{occ} содержит все точки луча сонара, расстояние до которых лежит в диапазоне $[r - \Delta r; r + \Delta r]$. Область Ω_{free} содержит все точки луча сонара, расстояние до которых меньше $r - \Delta r$.

Для численных экспериментов выбраны два набора данных. Первый набор данных синтезирован на основе искусственной карты, представленной на слева на рисунке 4.4, и все показания сонаров являются точными. В этом эксперименте робот движется по прямой снизу вверх (по центру карты), и на его корпусе располагаются три сонара с каждой стороны (слева и справа): один сонар с каждой стороны смотрит строго перпендикулярно в направлении к стене, и еще два сонара с каждой стороны развернуты к нему под углом 30 градусов, как показано на рисунке 4.7 (слева). Для экспериментов использовались следующие параметры: $\alpha_1 = 0.05$, $\alpha_2 = 0.01$, $c_1 = 5$, $c_2 = 1$

На рис. 4.9 приведены результаты экспериментов для территории, на которой расставлены препятствия на различных расстояниях друг от друга с использованием предложенного и традиционного методов. На графике 4.10 приведены зависимости ширины восстановленного прохода в зависимости от реальной ширины прохода. Как следует из графика, предложенный метод обнаруживает более точный размер проходов, а минимальная ширина прохода, обнаруживаемого предложенным методом, примерно на 30 % меньше по сравнению с традиционным методом.

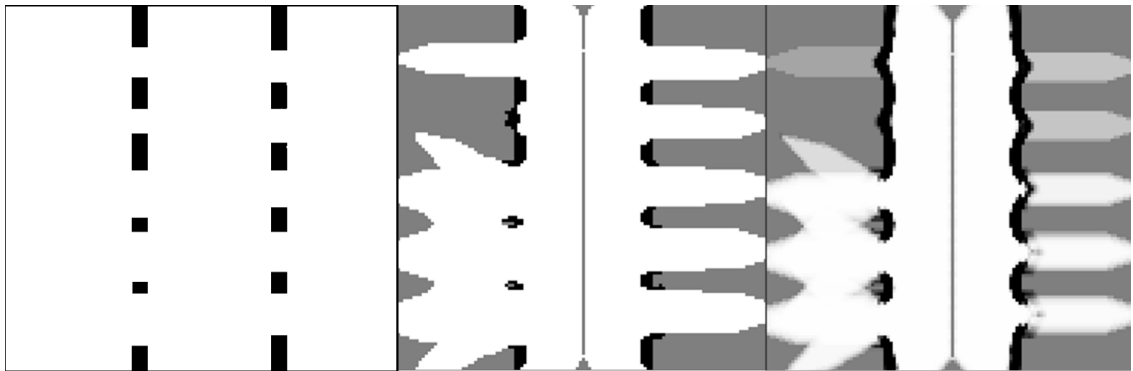


Рисунок 4.9. Карта с проходами различных размеров (слева), восстановленные карты: предложенным методом(в центре), традиционным методом (справа)

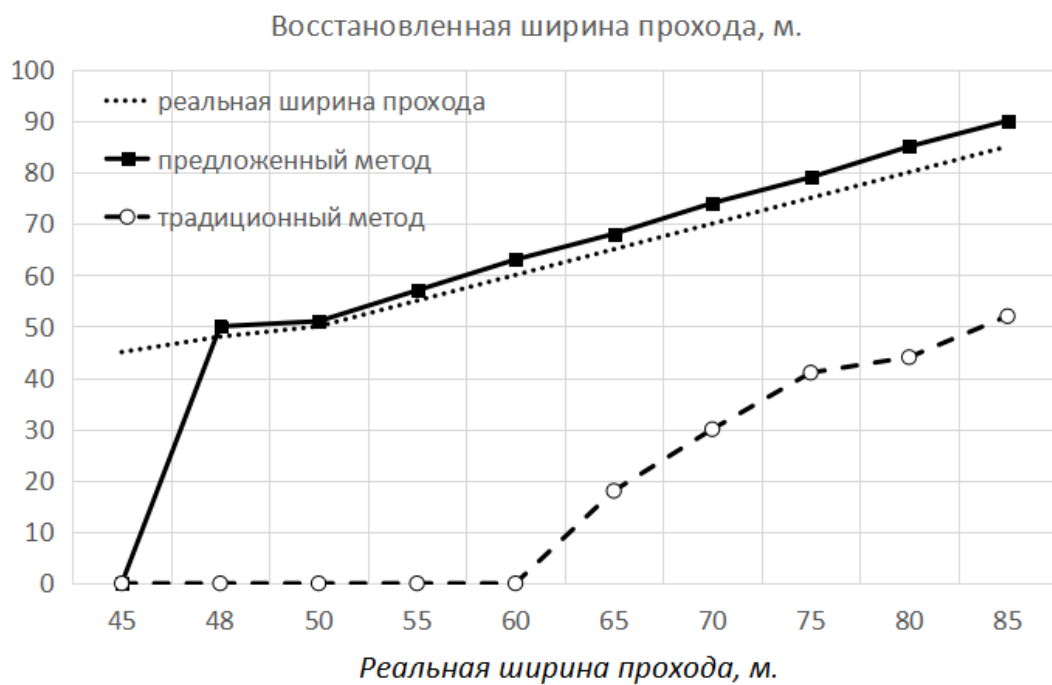


Рисунок 4.10. Зависимость восстановленной ширины прохода от реальной

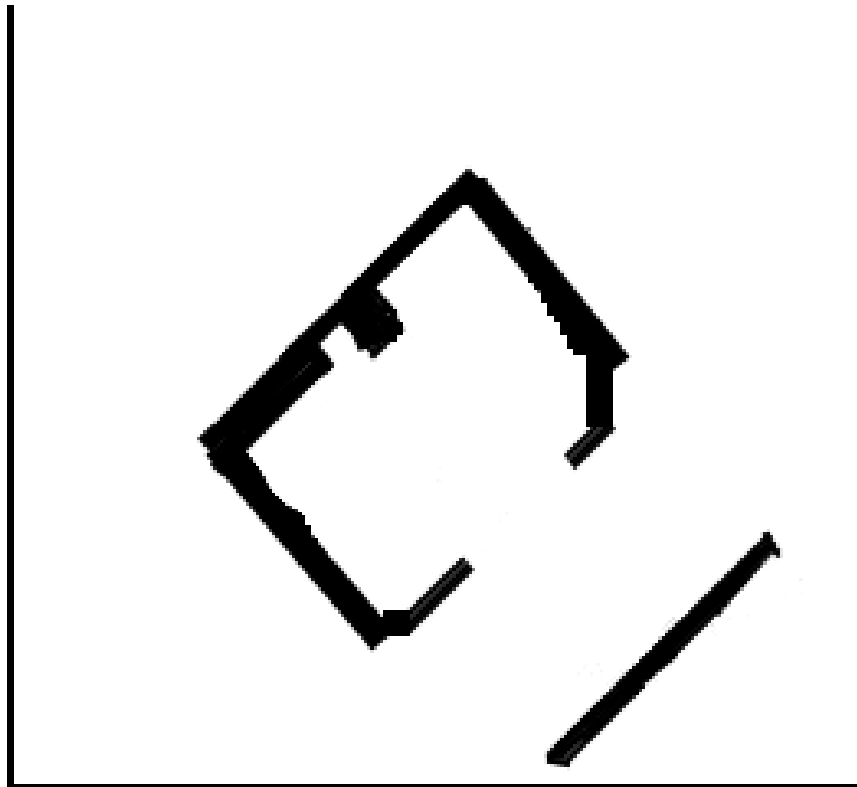


Рисунок 4.11. Реальная территория, на которой прозводилось тестирование предложенных алгоритмов

Второй набор данных собран с использованием настоящего робота, с передней стороны которого расположены 4 сонара как показано на рис. 4.7 (справа). При этом робот осуществляет полное вращение на месте в комнате, карта которой представлена справа на рисунке 4.11. Показания распределены неравномерно по углу (поскольку вращение осуществлялось не с постоянной скоростью). При этом некоторые показания сонаров являются выбросовыми, а также позиции сонаров измерены не совсем точно. Этот случай позволяет определить, какой эффективность обладает наш алгоритм на реальных данных по сравнению с традиционным. Результаты восстановления представлены на рисунке 4.12. Вследствие неравномерности снятых измерений, а также особенностей используемых методов, оба метода не смогли восстановить точную карту проходимости, однако метод на основе градиента сумел успешно восстановить дверное отверстие, в то время как традиционный метод – нет.

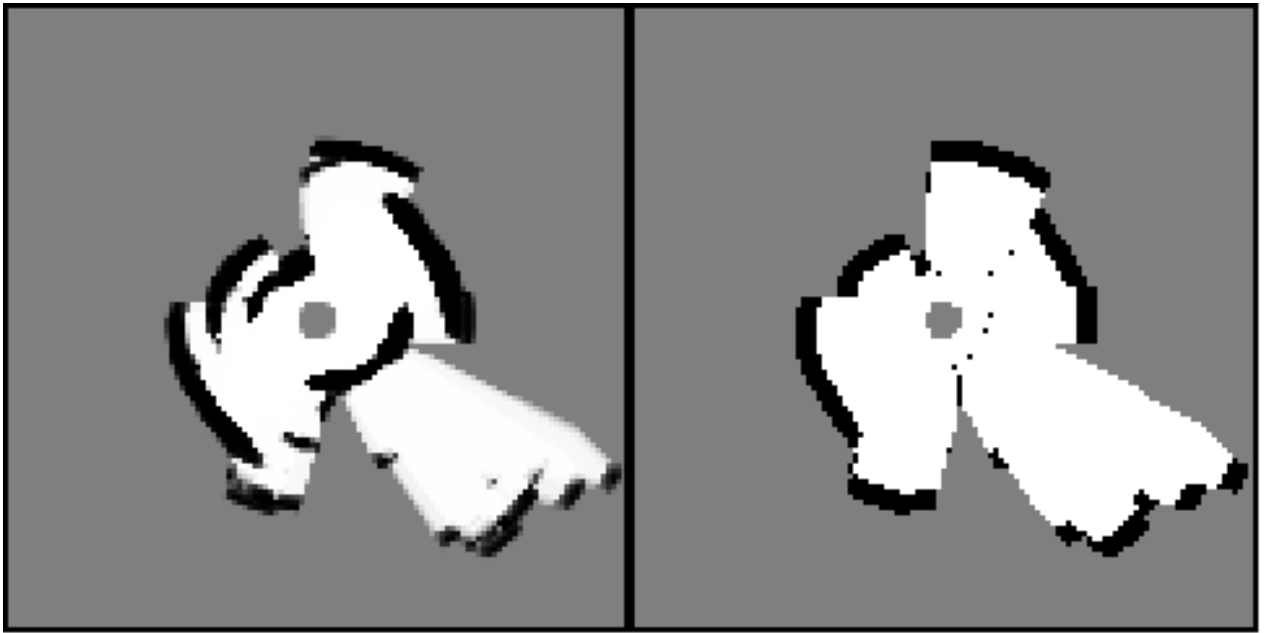


Рисунок 4.12. Результаты восстановления карты проходимости на основе реальных данных

4.6. Сравнение двух предложенных методов восстановления

Значительным недостатком обоих методов является отсутствие учета информации о нормалях. Фактически, при оценке невязки оба метода опираются на количество занятых и незанятых клеток в определенных частях луча сонара.

Если сравнить два предложенных метода восстановления карты, то видно, что стохастический метод восстановления уступает методу восстановления градиентным спуском. Скорее всего, это объясняется неточными параметрами настройки стохастического градиента. Вероятностная модель сонара позволяет учитывать выбросовые показания сонара. Рассмотрим пример: пусть сонар, находясь в одной точке, снял 10 показаний на расстоянии 2 метра и одно показание на расстоянии 1 метр. С использованием вероятностной модели сонара последнее измерение, скорее всего, будет учтено как случайное, и ближайшее препятствие будет восстановлено на расстоянии двух метров от сонара. Недостатком является (i) более сложная процедура восстановления и (ii) необходимость точно подобрать статистические параметры вероятностной модели.

Детерминистическая же модель не позволяет считать измерения выбросовыми, и следовательно, в описанной выше ситуации будет восстановлено неболь-

шое препятствие на расстоянии один метр, чтобы уменьшить невязку последнего измерения. Для того, чтобы проиллюстрировать это, рассмотрим, как оба алгоритма восстанавливают карту территории на основе реальных показаний сонаров.

4.7. Выводы к главе

В данной главе:

1. Разработано программное обеспечение для имитационного моделирования показаний сонаров на основе карт проходимости и система тестирования алгоритмов построения карт проходимости.
2. Предложена детерминистическая прямая модель сонара (forward sonar model). С использованием этой модели разработан алгоритм для построения карт методом градиентного спуска в пространстве карт проходимости. Алгоритм способен оперировать в режиме реального времени, обнаруживает проходы в 1.3 раза меньшей ширины, чем традиционные алгоритмы и существенно увеличивает точность определения ширины проходов.
3. Построен метод построения карт проходимости на основе показаний сонаров на основе вероятностной прямой модели сонара методом стохастического градиентного спуска в пространстве бинарных карт проходимости в режиме реального времени. В ряде сценариев алгоритм позволяет обнаруживать проходы в 1.3 раза меньшей ширины, чем традиционные алгоритмы.

Заключение

1. Разработана архитектура для обмена данных, хранения и обработки поступающей информации, а также оценки ее надежности и актуальности для системы продолжительного картирования меняющейся местности коллективом роботов.
2. Разработан комплекс программ имитационного моделирования для исследования и сравнения алгоритмов коллективного патрулирования.
3. Разработан и реализован алгоритм коллективного стохастического патрулирования, основанный на методе потенциалов, для которого получены численные результаты для некоторых типов территорий.
4. Разработано программное обеспечение для имитационного моделирования показаний сонаров на основе карт проходимости и система для тестирования алгоритмов построения карт проходимости.
5. Предложена детерминистическая прямая модель сонара (forward sonar model) для построения карт методом градиентного спуска в пространстве карт проходимости в режиме реального времени.
6. Предложен алгоритм построения карт проходимости на основе показаний сонаров, способный оперировать в режиме реального времени, обнаруживающий проходы в 1.3 раза меньшей ширины, чем традиционные алгоритмы и существенно увеличивающий точность определения ширины проходов.

Список сокращений и условных обозначений

РАН - Российская академия наук

ОПИК - Одновременное позиционирование и картирование

ОЕД - Основная единица данных

УИД - Уникальный идентификатор

ИР - Идентификатор региона

ХЛД - Хранилище локальных данных

ХПД - Хранилище полученных данных

ХЗР - Хранилище замыкающих ребер

ИК - Иерархическая кластеризация

БПФ - Быстрое преобразование Фурье

Словарь терминов

Ориентир – характерный объект, легко опознаваемый роботом или ключевая точка с оценкой положения и дескриптором.

Поза – структура, содержащая информацию об ориентирах и карту проходимости вблизи определенной точки территории.

Ассоциирование данных – процедура сравнения положения ориентиров двух поз для нахождения их взаимного положения.

Замыкающее (визуальное) ребро – вероятностная оценка пространственного расположения двух поз, полученная в результате ассоциирования данных.

Невизуальное ребро – вероятностная оценка пространственного расположения двух поз, полученная интегрированием показаний инерциальной навигационной системы и датчиков поворота колес.

Прямая модель сонара – вероятностное распределение показаний $p(z|M)$ сонара при заданном положении и известной карте M .

Метод потенциалов – метод принятия решений о направлении движения, основанный на введении и анализе виртуальных сил, действующих на робота со стороны различных объектов.

Голономный робот – робот, число управляемых степеней свободы которого равно полному числу степеней свободы.

Неголономный робот – робот, число управляемых степеней свободы которого меньше полного числа степеней свободы.

Список литературы

1. *Elfes Alberto*. Using occupancy grids for mobile robot perception and navigation // *Computer*. — 1989. — Vol. 22, no. 6. — Pp. 46–57.
2. *Borenstein Johann, Koren Yoram*. The vector field histogram-fast obstacle avoidance for mobile robots // *Robotics and Automation, IEEE Transactions on*. — 1991. — Vol. 7, no. 3. — Pp. 278–288.
3. Many robots make short work: Report of the SRI International mobile robot team / *Didier Guzzoni, Adam Cheyer, Luc Julia, Kurt Konolige* // *AI magazine*. — 1997. — Vol. 18, no. 1. — P. 55.
4. Experiences with an interactive museum tour-guide robot / *Wolfram Burgard, Armin B Cremers, Dieter Fox et al.* // *Artificial intelligence*. — 1999. — Vol. 114, no. 1. — Pp. 3–55.
5. *Chatila Raja, Laumond Jean-Paul*. Position referencing and consistent world modeling for mobile robots // *Robotics and Automation. Proceedings. 1985 IEEE International Conference on / IEEE*. — Vol. 2. — 1985. — Pp. 138–145.
6. *Shatkay Hagit, Kaelbling Leslie Pack*. Learning topological maps with weak local odometric information // *IJCAI (2)*. — 1997. — Pp. 920–929.
7. *Kortenkamp David, Weymouth Terry*. Topological mapping for mobile robots using a combination of sonar and vision sensing // *AAAI*. — Vol. 94. — 1994. — Pp. 979–984.
8. Integrating topological and metric maps for mobile robot navigation: A statistical approach / *Sebastian Thrun, Jens-Steffen Gutmann, Dieter Fox et al.* // *AAAI/I-AAI*. — 1998. — Pp. 989–995.
9. *Konolige Kurt, Marder-Eppstein Eitan, Marthi Bhaskara*. Navigation in hybrid metric-topological maps // *Robotics and Automation (ICRA), 2011 IEEE International Conference on / IEEE*. — 2011. — Pp. 3041–3047.

10. *Thrun Sebastian*. Learning metric-topological maps for indoor mobile robot navigation // *Artificial Intelligence*. — 1998. — Vol. 99, no. 1. — Pp. 21–71.
11. *Thrun Sebastian et al.* Robotic mapping: A survey // *Exploring artificial intelligence in the new millennium*. — 2002. — Pp. 1–35.
12. *Boberg Anders, Bishop Adrian N, Jensfelt Patric*. Robocentric mapping and localization in modified spherical coordinates with bearing measurements // *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2009 5th International Conference on / IEEE*. — 2009. — Pp. 139–144.
13. *Bishop Adrian N, Jensfelt Patric*. A stochastically stable solution to the problem of robocentric mapping // *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on / IEEE*. — 2009. — Pp. 1615–1622.
14. *Konolige Kurt*. Improved occupancy grids for map building // *Autonomous Robots*. — 1997. — Vol. 4, no. 4. — Pp. 351–367.
15. *Rusu Radu Bogdan, Cousins Steve*. 3d is here: Point cloud library (pcl) // *Robotics and Automation (ICRA), 2011 IEEE International Conference on / IEEE*. — 2011. — Pp. 1–4.
16. Towards 3D point cloud based object maps for household environments / Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow et al. // *Robotics and Autonomous Systems*. — 2008. — Vol. 56, no. 11. — Pp. 927–941.
17. *Se Stephen, Lowe David, Little Jim*. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks // *The international Journal of robotics Research*. — 2002. — Vol. 21, no. 8. — Pp. 735–758.
18. *Durrant-Whyte Hugh, Bailey Tim*. Simultaneous localization and mapping: part I // *Robotics & Automation Magazine, IEEE*. — 2006. — Vol. 13, no. 2. — Pp. 99–110.
19. *Kweon In So, Kanade Takeo*. Extracting topographic terrain features from elevation maps // *CVGIP: image understanding*. — 1994. — Vol. 59, no. 2. — Pp. 171–182.

20. *Triebel Rudolph, Pfaff Patrick, Burgard Wolfram*. Multi-level surface maps for outdoor terrain mapping and loop closing // *Intelligent Robots and Systems*, 2006 IEEE/RSJ International Conference on / IEEE. — 2006. — Pp. 2276–2282.
21. *Dryanovski Ivan, Morris William, Xiao Jizhong*. Multi-volume occupancy grids: An efficient probabilistic 3D mapping model for micro aerial vehicles // *Intelligent Robots and Systems (IROS)*, 2010 IEEE/RSJ International Conference on / IEEE. — 2010. — Pp. 1553–1559.
22. 6D SLAM with an application in autonomous mine mapping / *Andreas Nuchter, Hartmut Surmann, Kai Lingemann et al.* // *Robotics and Automation*, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on / IEEE. — Vol. 2. — 2004. — Pp. 1998–2003.
23. *Bauer Johannes, Sunderhauf Niko, Protzel Peter*. Comparing several implementations of two recently published feature detectors // *Proc. of the International Conference on Intelligent and Autonomous Systems*. — Vol. 6. — 2007.
24. A comparative evaluation of interest point detectors and local descriptors for visual SLAM / *Arturo Gil, Oscar Martinez Mozos, Monica Ballesta, Oscar Reinoso* // *Machine Vision and Applications*. — 2010. — Vol. 21, no. 6. — Pp. 905–920.
25. *Brenneke Christian, Wulf Oliver, Wagner Bernardo*. Using 3d laser range data for slam in outdoor environments // *Intelligent Robots and Systems*, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on / IEEE. — Vol. 1. — 2003. — Pp. 188–193.
26. *Frintrop Simone, Jensfelt Patric*. Attentional landmarks and active gaze control for visual SLAM // *Robotics, IEEE Transactions on*. — 2008. — Vol. 24, no. 5. — Pp. 1054–1065.
27. *Eade Ethan, Drummond Tom*. Scalable monocular SLAM // *Computer Vision and Pattern Recognition*, 2006 IEEE Computer Society Conference on / IEEE. — Vol. 1. — 2006. — Pp. 469–476.
28. *Garcia Miguel Angel, Solanas Agusti*. 3D simultaneous localization and modeling from stereo vision // *Robotics and Automation*, 2004. Proceedings. ICRA'04.

- 2004 IEEE International Conference on / IEEE. — Vol. 1. — 2004. — Pp. 847–853.
29. *Scharstein Daniel, Szeliski Richard*. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms // *International journal of computer vision*. — 2002. — Vol. 47, no. 1-3. — Pp. 7–42.
 30. Occupancy grid computation from dense stereo and sparse structure and motion points for automotive applications / Henning Lategahn, Wojciech Derendarz, Thorsten Graf et al. // *Intelligent Vehicles Symposium (IV)*, 2010 IEEE / IEEE. — 2010. — Pp. 819–824.
 31. Visual odometry and mapping for autonomous flight using an RGB-D camera / Albert S Huang, Abraham Bachrach, Peter Henry et al. // *International Symposium on Robotics Research (ISRR)*. — 2011. — Pp. 1–16.
 32. *Stachniss Cyrill, Hähnel Dirk, Burgard Wolfram*. Exploration with active loop-closing for FastSLAM // *Intelligent Robots and Systems*, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on / IEEE. — Vol. 2. — 2004. — Pp. 1505–1510.
 33. *Neira José, Tardós Juan D*. Data association in stochastic mapping using the joint compatibility test // *Robotics and Automation, IEEE Transactions on*. — 2001. — Vol. 17, no. 6. — Pp. 890–897.
 34. Data association for mobile robot navigation: A graph theoretic approach / Tim Bailey, Eduardo Mario Nebot, JK Rosenblatt, Hugh F Durrant-Whyte // *Robotics and Automation*, 2000. Proceedings. ICRA'00. IEEE International Conference on / IEEE. — Vol. 3. — 2000. — Pp. 2512–2517.
 35. *Thrun Sebastian*. Probabilistic robotics // *Communications of the ACM*. — 2002. — Vol. 45, no. 3. — Pp. 52–57.
 36. *Thrun Sebastian*. A probabilistic on-line mapping algorithm for teams of mobile robots // *The International Journal of Robotics Research*. — 2001. — Vol. 20, no. 5. — Pp. 335–363.

37. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems / Kai M Wurm, Armin Hornung, Maren Bennewitz et al. // Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation. — Vol. 2. — 2010.
38. *Durrant-Whyte Hugh, Bailey Tim*. Simultaneous localization and mapping: part I // *Robotics & Automation Magazine, IEEE*. — 2006. — Vol. 13, no. 2. — Pp. 99–110.
39. *Durrant-Whyte Hugh F*. Uncertain geometry in robotics // *Robotics and Automation, IEEE Journal of*. — 1988. — Vol. 4, no. 1. — Pp. 23–31.
40. *Smith Randall, Self Matthew, Cheeseman Peter*. Estimating uncertain spatial relationships in robotics // *Autonomous robot vehicles*. — Springer, 1990. — Pp. 167–193.
41. *Leonard John J, Durrant-Whyte Hugh F*. Simultaneous map building and localization for an autonomous mobile robot // *Intelligent Robots and Systems' 91. Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on / Ieee*. — 1991. — Pp. 1442–1447.
42. *Borenstein Johann, Feng Liqiang, Everett HR*. Navigating mobile robots: systems and techniques. — AK Peters, Ltd., 1996.
43. *Filliat David, Meyer Jean-Arcady*. Map-based navigation in mobile robots:: I. a review of localization strategies // *Cognitive Systems Research*. — 2003. — Vol. 4, no. 4. — Pp. 243–282.
44. *Moravec Hans P*. Sensor fusion in certainty grids for mobile robots // *AI magazine*. — 1988. — Vol. 9, no. 2. — P. 61.
45. *Bailey Tim, Durrant-Whyte Hugh*. Simultaneous localization and mapping (SLAM): Part II // *IEEE Robotics & Automation Magazine*. — 2006. — Vol. 13, no. 3. — Pp. 108–117.
46. *Grisetti Giorgio, Stachniss Cyrill, Burgard Wolfram*. Improved techniques for grid mapping with rao-blackwellized particle filters // *Robotics, IEEE Transactions on*. — 2007. — Vol. 23, no. 1. — Pp. 34–46.

47. A bayesian algorithm for simultaneous localisation and map building / Hugh Durrant-Whyte, Somajyoti Majumder, Sebastian Thrun et al. // *Robotics Research*. — Springer, 2003. — Pp. 49–60.
48. A solution to the simultaneous localization and map building (SLAM) problem / MWMG Dissanayake, Paul Newman, Steven Clark et al. // *Robotics and Automation, IEEE Transactions on*. — 2001. — Vol. 17, no. 3. — Pp. 229–241.
49. A random-finite-set approach to Bayesian SLAM / John Mullane, Ba-Ngu Vo, Martin D Adams, Ba-Tuong Vo // *Robotics, IEEE Transactions on*. — 2011. — Vol. 27, no. 2. — Pp. 268–282.
50. Blanco Jose-Luis, Fernández-Madrigal Juan-Antonio, Gonzalez Jose. Toward a Unified Bayesian Approach to Hybrid Metric–Topological SLAM // *Robotics, IEEE Transactions on*. — 2008. — Vol. 24, no. 2. — Pp. 259–270.
51. Sonar-based mapping with mobile robots using EM / Wolfram Burgard, Dieter Fox, Hauke Jans et al. // MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE- / MORGAN KAUFMANN PUBLISHERS, INC. — 1999. — Pp. 67–76.
52. Map building with mobile robots in dynamic environments / Dirk Hahnel, Rudolph Triebel, Wolfram Burgard, Sebastian Thrun // *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on / IEEE*. — Vol. 2. — 2003. — Pp. 1557–1563.
53. Moon Tood K. The expectation-maximization algorithm // *Signal processing magazine, IEEE*. — 1996. — Vol. 13, no. 6. — Pp. 47–60.
54. A tutorial on graph-based SLAM / Giorgio Grisetti, Rainer Kümmeler, Cyrill Stachniss, Wolfram Burgard // *Intelligent Transportation Systems Magazine, IEEE*. — 2010. — Vol. 2, no. 4. — Pp. 31–43.
55. Thrun Sebastian, Montemerlo Michael. The graph SLAM algorithm with applications to large-scale mapping of urban structures // *The International Journal of Robotics Research*. — 2006. — Vol. 25, no. 5-6. — Pp. 403–429.

56. g 2 o: A general framework for graph optimization / Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat et al. // Robotics and Automation (ICRA), 2011 IEEE International Conference on / IEEE. — 2011. — Pp. 3607–3613.
57. *Le Corff S, Fort G, Moulines E*. Online expectation maximization algorithm to solve the slam problem // Statistical Signal Processing Workshop (SSP), 2011 IEEE / IEEE. — 2011. — Pp. 225–228.
58. An evaluation of the RGB-D SLAM system / Felix Endres, Jürgen Hess, Nikolas Engelhard et al. // Robotics and Automation (ICRA), 2012 IEEE International Conference on / IEEE. — 2012. — Pp. 1691–1696.
59. Towards robust data association and feature modeling for concurrent mapping and localization / John J Leonard, Paul M Newman, Richard J Rikoski et al. // Robotics Research. — Springer, 2003. — Pp. 7–20.
60. *Ho Kin Leong, Newman Paul*. Loop closure detection in SLAM by combining visual and spatial appearance // *Robotics and Autonomous Systems*. — 2006. — Vol. 54, no. 9. — Pp. 740–749.
61. *Latif Yasir, Cadena César, Neira José*. Robust loop closing over time for pose graph SLAM // *The International Journal of Robotics Research*. — 2013. — P. 0278364913498910.
62. *Aragues Rosario, Montijano Eduardo, Sagüés Carlos*. Consistent data association in multi-robot systems with limited communications. // Robotics: Science and Systems / Citeseer. — 2010. — Pp. 97–104.
63. Robust map optimization using dynamic covariance scaling / Prabhakar Agarwal, Gian Diego Tipaldi, Luciano Spinello et al. // Robotics and Automation (ICRA), 2013 IEEE International Conference on / IEEE. — 2013. — Pp. 62–69.
64. Robust map optimization using dynamic covariance scaling / Prabhakar Agarwal, Gian Diego Tipaldi, Luciano Spinello et al. // Robotics and Automation (ICRA), 2013 IEEE International Conference on / IEEE. — 2013. — Pp. 62–69.

65. *Lee Donghwa, Myung Hyun*. Solution to the SLAM problem in low dynamic environments using a pose graph and an RGB-D sensor // *Sensors*. — 2014. — Vol. 14, no. 7. — Pp. 12467–12496.
66. Position estimation for mobile robots in dynamic environments / Dieter Fox, Wolfram Burgard, Sebastian Thrun, Armin B Cremers // *AAAI/IAAI*. — 1998. — Pp. 983–988.
67. *Hähnel Dirk, Schulz Dirk, Burgard Wolfram*. Map building with mobile robots in populated environments // *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on / IEEE*. — Vol. 1. — 2002. — Pp. 496–501.
68. *Biber Peter, Duckett Tom*. Experimental analysis of sample-based maps for long-term SLAM // *The International Journal of Robotics Research*. — 2009. — Vol. 28, no. 1. — Pp. 20–33.
69. *Bahr Alexander, Walter Matthew R, Leonard John J*. Consistent cooperative localization // *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on / IEEE*. — 2009. — Pp. 3415–3422.
70. Graph-based distributed cooperative navigation for a general multi-robot measurement model / Vadim Indelman, Pini Gurfil, Ehud Rivlin, Hector Rotstein // *The International Journal of Robotics Research*. — 2012. — P. 0278364912446325.
71. Simultaneous localization and mapping with sparse extended information filters / Sebastian Thrun, Yufeng Liu, Daphne Koller et al. // *The International Journal of Robotics Research*. — 2004. — Vol. 23, no. 7-8. — Pp. 693–716.
72. *Cunningham Alexander, Paluri Manohar, Dellaert Frank*. DDF-SAM: Fully distributed slam using constrained factor graphs // *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on / IEEE*. — 2010. — Pp. 3025–3030.
73. Coordinated multi-robot exploration / Wolfram Burgard, Mark Moors, Cyrill Stachniss, Frank E Schneider // *Robotics, IEEE Transactions on*. — 2005. — Vol. 21, no. 3. — Pp. 376–386.

74. Recent advances on multi-agent patrolling / Alessandro Almeida, Geber Ramalho, Hugo Santana et al. // *Advances in Artificial Intelligence—SBIA 2004*. — Springer, 2004. — Pp. 474–483.
75. *Cunningham Andrew, Indelman Vadim, Dellaert Frank*. DDF-SAM 2.0: Consistent distributed smoothing and mapping // *Robotics and Automation (ICRA), 2013 IEEE International Conference on / IEEE*. — 2013. — Pp. 5220–5227.
76. *Kretzschmar Henrik, Grisetti Giorgio, Stachniss Cyrill*. Lifelong map learning for graph-based slam in static environments // *KI-Künstliche Intelligenz*. — 2010. — Vol. 24, no. 3. — Pp. 199–206.
77. *Snavely Noah, Seitz Steven M, Szeliski Richard*. Skeletal graphs for efficient structure from motion. // *CVPR*. — Vol. 1. — 2008. — P. 2.
78. *Konolige Kurt, Bowman James*. Towards lifelong visual maps // *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on / IEEE*. — 2009. — Pp. 1156–1163.
79. *Milford Michael, Wyeth Gordon*. Persistent navigation and mapping using a biologically inspired SLAM system // *The International Journal of Robotics Research*. — 2010. — Vol. 29, no. 9. — Pp. 1131–1153.
80. Multi-agent patrolling: An empirical analysis of alternative architectures / Aydano Machado, Geber Ramalho, Jean-Daniel Zucker, Alexis Drogoul // *Multi-Agent-Based Simulation II*. — Springer, 2002. — Pp. 155–170.
81. *Portugal David, Rocha Rui P*. Multi-robot patrolling algorithms: examining performance and scalability // *Advanced Robotics*. — 2013. — Vol. 27, no. 5. — Pp. 325–336.
82. Swarm approaches for the patrolling problem, information propagation vs. pheromone evaporation / Hoang-Nam Chu, Arnaud Glad, Olivier Simonin et al. // *Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on / IEEE*. — Vol. 1. — 2007. — Pp. 442–449.

83. Multi-agent movement coordination in patrolling / Aydano Machado, Alessandro Almeida, G Ramaldo et al. // Proceedings of the 3rd International Conference on Computer and Game. — 2002. — Pp. 155–170.
84. *Elmaliach Yehuda, Agmon Noa, Kaminka Gal A.* Multi-robot area patrol under frequency constraints // *Annals of Mathematics and Artificial Intelligence*. — 2009. — Vol. 57, no. 3-4. — Pp. 293–320.
85. *Elor Yotam, Bruckstein Alfred M.* Autonomous multi-agent cycle based patrolling // *Swarm Intelligence*. — Springer, 2010. — Pp. 119–130.
86. *Sempé Francois, Drogoul Alexis.* Adaptive patrol for a group of robots // *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on / IEEE*. — Vol. 3. — 2003. — Pp. 2865–2869.
87. *Chen Kuan-Chieh, Tsai Wen-Hsiang.* Vision-based autonomous vehicle guidance for indoor security patrolling by a SIFT-based vehicle-localization technique // *Vehicular Technology, IEEE Transactions on*. — 2010. — Vol. 59, no. 7. — Pp. 3261–3271.
88. *Chen Ming-Che, Tsai Wen-Hsiang.* Vision-based security patrolling in indoor environments using autonomous vehicles // *Proc. of 2005 Conf. on Computer Vision, Graphics & Image Processing*. — 2005.
89. *Kolling Andreas, Carpin Stefano.* Extracting surveillance graphs from robot maps // *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on / IEEE*. — 2008. — Pp. 2323–2328.
90. *Wurm Kai M, Stachniss Cyrill, Burgard Wolfram.* Coordinated multi-robot exploration using a segmentation of the environment // *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on / IEEE*. — 2008. — Pp. 1160–1165.
91. *Portugal David, Rocha Rui.* Msp algorithm: multi-robot patrolling based on territory allocation using balanced graph partitioning // *Proceedings of the 2010 ACM Symposium on Applied Computing / ACM*. — 2010. — Pp. 1271–1276.

92. Multi-agent patrolling with reinforcement learning / Hugo Santana, Geber Ramalho, Vincent Corruble, Bohdana Ratitch // *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 3* / IEEE Computer Society. — 2004. — Pp. 1122–1129.
93. *Sutton Richard S, Barto Andrew G. Reinforcement learning: An introduction.* — MIT press, 1998.
94. *Menezes Talita, Tedesco Patrícia, Ramalho Geber.* Negotiator agents for the patrolling task // *Advances in Artificial Intelligence-IBERAMIA-SBIA 2006.* — Springer, 2006. — Pp. 48–57.
95. *Barnes Laura, Fields MaryAnne, Valavanis Kimon.* Unmanned ground vehicle swarm formation control using potential fields // *Control & Automation, 2007. MED'07. Mediterranean Conference on / IEEE.* — 2007. — Pp. 1–8.
96. *Spears William M, Spears Diana F.* Physicomimetics: Physics-based swarm intelligence. — Springer Science & Business Media, 2012.
97. Heterogeneous swarm formation control using bivariate normal functions to generate potential fields / Laura Barnes, Wendy Alvis, MaryAnne Fields et al. // *Distributed Intelligent Systems: Collective Intelligence and Its Applications, 2006. DIS 2006. IEEE Workshop on / IEEE.* — 2006. — Pp. 85–94.
98. *Kim Dong Hun, Wang Hua, Shin Seiichi.* Decentralized control of autonomous swarm systems using artificial potential functions: Analytical design guidelines // *Journal of Intelligent and Robotic Systems.* — 2006. — Vol. 45, no. 4. — Pp. 369–394.
99. *Gazi Veysel.* Swarm aggregations using artificial potentials and sliding-mode control // *Robotics, IEEE Transactions on.* — 2005. — Vol. 21, no. 6. — Pp. 1208–1214.
100. *Barraquand Jerome, Langlois Bruno, Latombe Jean-Claude.* Numerical potential field techniques for robot path planning // *Systems, Man and Cybernetics, IEEE Transactions on.* — 1992. — Vol. 22, no. 2. — Pp. 224–241.

101. *Ge Shuzhi Sam, Cui Yan Juan*. New potential functions for mobile robot path planning // *IEEE Transactions on robotics and automation*. — 2000. — Vol. 16, no. 5. — Pp. 615–620.
102. *Tsourveloudis Nikos C, Valavanis Kimon P, Hebert Timothy*. Autonomous vehicle navigation utilizing electrostatic potential fields and fuzzy logic // *Robotics and Automation, IEEE Transactions on*. — 2001. — Vol. 17, no. 4. — Pp. 490–497.
103. *Reif John H, Wang Hongyan*. Social potential fields: A distributed behavioral control for autonomous robots // *Robotics and Autonomous Systems*. — 1999. — Vol. 27, no. 3. — Pp. 171–194.
104. *Kolling Andreas, Carpin Stefano*. Extracting surveillance graphs from robot maps // *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on / IEEE*. — 2008. — Pp. 2323–2328.
105. The “Ariadne’s clew” algorithm: global planning with local methods / Pierre Bessiere, Juan-Manuel Ahuactzin, El-Ghazali Talbi, Emmanuel Mazer // *Intelligent Robots and Systems’ 93, IROS’93. Proceedings of the 1993 IEEE/RSJ International Conference on / IEEE*. — Vol. 2. — 1993. — Pp. 1373–1380.
106. Recent progress in local and global traversability for planetary rovers / Sanjiv Singh, Reid Simmons, Trey Smith et al. // *Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE International Conference on / IEEE*. — Vol. 2. — 2000. — Pp. 1194–1200.
107. *Hagras Hani A*. A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots // *Fuzzy Systems, IEEE Transactions on*. — 2004. — Vol. 12, no. 4. — Pp. 524–539.
108. *Botea Adi, Müller Martin, Schaeffer Jonathan*. Near optimal hierarchical path-finding // *Journal of game development*. — 2004. — Vol. 1, no. 1. — Pp. 7–28.
109. *Kanayama Yutaka, Hartman Bruce I*. Smooth local path planning for autonomous vehicles // *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on / IEEE*. — 1989. — Pp. 1265–1270.

110. *Reeds James, Shepp Lawrence*. Optimal paths for a car that goes both forwards and backwards // *Pacific journal of mathematics*. — 1990. — Vol. 145, no. 2. — Pp. 367–393.
111. *Giordano Paolo Robuffo, Vendittelli Marilena*. Shortest paths to obstacles for a polygonal Dubins car // *Robotics, IEEE Transactions on*. — 2009. — Vol. 25, no. 5. — Pp. 1184–1191.
112. *Moras Julien, Cherfaoui Véronique, Bonnifait Philippe*. Credibilist occupancy grids for vehicle perception in dynamic environments // *Robotics and Automation (ICRA), 2011 IEEE International Conference on / IEEE*. — 2011. — Pp. 84–89.
113. *Thrun Sebastian*. Learning metric-topological maps for indoor mobile robot navigation // *Artificial Intelligence*. — 1998. — Vol. 99, no. 1. — Pp. 21–71.
114. Efficient occupancy grid computation on the GPU with lidar and radar for road boundary detection / Florian Hogg, Nico Kaempchen, Jeff Ota, Darius Burschka // *Intelligent Vehicles Symposium (IV), 2010 IEEE / IEEE*. — 2010. — Pp. 1006–1013.
115. *Latombe Jean-Claude*. Robot Motion Planning (The Kluwer International Series in Engineering and Computer Science). — 1990.
116. Robust mapping and localization in indoor environments using sonar data / Juan D Tardós, José Neira, Paul M Newman, John J Leonard // *The International Journal of Robotics Research*. — 2002. — Vol. 21, no. 4. — Pp. 311–330.
117. *Illingworth John, Kittler Josef*. A survey of the Hough transform // *Computer vision, graphics, and image processing*. — 1988. — Vol. 44, no. 1. — Pp. 87–116.
118. *Paskin Mark, Thrun Sebastian*. Robotic mapping with polygonal random fields // *arXiv preprint arXiv:1207.1399*. — 2012.
119. *Thrun Sebastian*. Learning occupancy grid maps with forward sensor models // *Autonomous robots*. — 2003. — Vol. 15, no. 2. — Pp. 111–127.

120. *Elfes Alberto*. Occupancy grids: a probabilistic framework for robot perception and navigation. — 1989.
121. *Leonard John J, Durrant-Whyte Hugh F, Cox Ingemar J*. Dynamic map building for an autonomous mobile robot // *The International Journal of Robotics Research*. — 1992. — Vol. 11, no. 4. — Pp. 286–298.
122. *Papoulis Athanasios, Pillai S Unnikrishna*. Probability, random variables, and stochastic processes. — Tata McGraw-Hill Education, 2002.
123. *Shvets Evgeny, Nikolaev Dmitry*. Complex approach to long-term multi-agent mapping in low dynamic environments // Proceedings of SPIE 9875, Eighth International Conference on Machine Vision (ICMV 2015). – Barcelona / International Society for Optics and Photonics. — Vol. 9875A. — 2015. — Pp. 1–10.
124. Communication-constrained multi-AUV cooperative SLAM / Liam Paull, Guoquan Huang, Mae Seto, John J Leonard // 2015 IEEE International Conference on Robotics and Automation (ICRA) / IEEE. — 2015. — Pp. 509–516.
125. *Scharstein Daniel, Szeliski Richard*. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms // *International journal of computer vision*. — 2002. — Vol. 47, no. 1-3. — Pp. 7–42.
126. *Hartley Richard, Zisserman Andrew*. Multiple view geometry in computer vision. — Cambridge university press, 2003.
127. *Johnson Stephen C*. Hierarchical clustering schemes // *Psychometrika*. — 1967. — Vol. 32, no. 3. — Pp. 241–254.
128. *Muja Marius, Lowe David G*. Flann, fast library for approximate nearest neighbors // International Conference on Computer Vision Theory and Applications (VISAPP'09). — 2009.
129. *Fischler Martin A, Bolles Robert C*. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography // *Communications of the ACM*. — 1981. — Vol. 24, no. 6. — Pp. 381–395.

130. *Sheather Simon J.* Density estimation // *Statistical Science* 19.4. — 2004. — Pp. 588–597.
131. *Garfinkel Robert S, Nemhauser George L.* Integer programming. — Wiley New York, 1972. — Vol. 4.
132. *Grisetti Giorgio et al.* A tutorial on graph-based SLAM // *Intelligent Transportation Systems Magazine, IEEE* 2.4. — 2010. — Pp. 31–43.
133. *C. Sidney S. Burrus, and Thomas W. Parks.* DFT/FFT and Convolution Algorithms: Theory and Implementation // John Wiley & Sons, Inc., 1991.
134. *Shvets Evgeny.* Stochastic multi-agent patrolling using social potential fields // *Proc. of 29th European Conf. on Modelling and Simulation.* — 2015. — Pp. 42–49.
135. *Швец Е.А.* Зависимость эффективности коллективного стохастического патрулирования от связности и надежности беспроводной сети // *Труды ИСА РАН – М.: Изд-во УРСС.* — 2016. — Vol. 2016-66, no. 1. — Pp. 70–73.
136. *Stroustrup Bjarne.* The C++ programming language. — Pearson Education India, 1986.
137. *Elmaliach Yehuda, Shiloni Asaf, Kaminka Gal A.* A realistic model of frequency-based multi-robot polyline patrolling // *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1 / International Foundation for Autonomous Agents and Multiagent Systems.* — 2008. — Pp. 63–70.
138. *Rubin Frank.* The Lee path connection algorithm // *Computers, IEEE Transactions on.* — 1974. — Vol. 100, no. 9. — Pp. 907–914.
139. *Otsu Nobuyuki.* A threshold selection method from gray-level histograms // *Automatica.* — 1975. — Vol. 11, no. 285-296. — Pp. 23–27.
140. *Швец Е. А. Николаев Д. П. Шенелев Д. А.* Восстановление карты проходимости с использованием прямой модели сонаров методом

- градиентного спуска // *Информационные процессы*. – М.: ИППИ РАН. — 2016. — Vol. 16, no. 1. — Pp. 61–71.
141. Швец Е. А. Николаев Д. П. Шепелев Д. А. Построение карты проходимости на основе показаний датчиков расстояния методом стохастического градиента // *Труды ИСА РАН* – М.: Изд-во УРСС. — 2016. — Vol. 66, no. 1. — Pp. 64–69.
142. LV MaxSonar EZ Series Data sheet // http://maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf.
143. CT metal artifact reduction by soft inequality constraints / Marina Chukalina, Dmitry Nikolaev, Valerii Sokolov et al. // Eighth International Conference on Machine Vision / International Society for Optics and Photonics. — 2015. — Pp. 98751C–98751C.

Приложение

Справки о внедрении результатов диссертационной работы

VISILLECT
SERVICE

ООО «Визиллект Сервис»
ИНН/КПП: 7707761588/770701001
ОГРН 1117746856248

127055, Россия, г. Москва
Порядковый пер., д. 21, оф. 401
тел./факс: +7 (499) 504-42-62
e-mail: office@visillect.com

№ 93/1 от 05.08.2016г.

на № _____ от _____

АКТ

о внедрении результатов диссертационной работы Швеца Евгения Александровича
«Разработка моделей картирования и патрулирования коллективом беспилотных
наземных роботов, использующих техническое зрение и эхолокацию»

Настоящим подтверждаем, что в разработанном ООО «Визиллект-Сервис» в рамках проекта «Стая» прототипе беспилотной наземной робототехнической платформы использованы следующие результаты диссертационной работы Е.А. Швеца:

1. Система тестирования и визуализации алгоритмов построения карты проходимости с использованием имитационной модели показаний сонаров;
2. Алгоритм построения карты проходимости на основе показаний сонаров;
3. Алгоритмы локализации и навигации по визуальным данным, основанные на методах сопоставления облаков особых точек.

Алгоритм восстановления карты проходимости на показаний сонаров работает в реальном времени в условиях загрузки процессора (Intel® Core™ i7-3610QM) другими программами прототипа и позволяет обнаруживать проходы, в 1,3 раза более узкие, чем традиционные алгоритмы реального времени, а также существенно увеличить точность определения ширины проходов.

Указанный прототип беспилотной робототехнической платформы продемонстрирован на выставке Skolkovo Robotics в мае 2016 года.

Генеральный директор, к.т.н.



Д.Н. Мацнев



ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ
ИНСТИТУТ ПРОБЛЕМ ПЕРЕДАЧИ ИНФОРМАЦИИ
 им. А.А. Харкевича Российской академии наук

Большой Каретный пер., д. 19, стр. 1, Москва, 127051
 Тел.: (495) 650-42-25. Факс: (495) 650-05-79. E-mail: director@iitp.ru
 ОКПО: 02699464 ОГРН: 1037700064940 ИНН/КПП: 7707020131/770701001
<http://www.iitp.ru>

04.04 2016 г. № 11615 – 2175 1380

На № _____ от _____

Справка о внедрении

теоретических и практических результатов диссертационной работы Швеца Е.А. «Разработка моделей картирования и патрулирования коллективом беспилотных наземных роботов, использующих техническое зрение и эхолокацию», представленной на соискание ученой степени кандидата технических наук, при разработке НИР, проводимых ИППИ РАН

Теоретические и практические результаты диссертационной работы Швеца Е.А. «Разработка моделей картирования и патрулирования коллективом беспилотных наземных роботов, использующих техническое зрение и эхолокацию», а именно: 1) создание среды имитационного моделирования совместного поведения группы беспилотных аппаратов с учетом ограничений управления, ошибок навигации, ограничений канала связи; 2) создание модели интеграции и комплексирования данных в группе беспилотных аппаратов при условии возможной длительной потери связи и ограниченной памяти роботов – были использованы в рамках выполнения НИР по направлению «Фундаментальные проблемы разработки интеллектуальных автономных систем» проекта «Цифровые технологии и их применения» Российского Научного Фонда (соглашение № 14-50-00150 от 26.12.2014 г.).

Врио директора ИППИ РАН
 д.ф.-м.н., Профессор РАН



А.Н. Соболевский