

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра Математического обеспечения ЭВМ**

**ОТЧЕТ**

**Тема: «Исследование алгоритмов навигации группы мобильных роботов в  
2D окружении»**  
**(«Исследование методов одновременной локализации и построения карты  
группой мобильных роботов»)**

Студент гр. 2304

\_\_\_\_\_

Кузьмин М.Е.

Руководитель

\_\_\_\_\_

Кринкин К.В.

Санкт-Петербург

2017

## ЗАДАНИЕ

Студент: Кузьмин М.Е.

Группа: 2304

Тема практики: «Исследование алгоритмов навигации группы мобильных роботов в 2D окружении»

Задание на практику:

Исследовать существующие алгоритмы SLAM (алгоритмы локализации и картографирования) роботов. Рассмотреть существующие расширения данных методов для применения к группам роботов. Составить классификацию. Провести сравнительный анализ алгоритмов.

## СОДЕРЖАНИЕ

Содержание.....	3
Введение.....	5
1. Метды одновременной локализации робота и построения карты (SLAM) .....	7
1.1. Классификация.....	8
1.2. Основные back-end методы SLAM .....	11
1.2.1. Extended Kalman Filter SLAM .....	12
1.2.1. Sparse Extended information filter SLAM .....	14
1.2.2. FastSLAM .....	15
1.2.3. DP-SLAM .....	18
1.2.4. Grid-based FastSLAM и Gmapping .....	21
1.2.5. Graph-based SLAM .....	23
1.2.1. Neural SLAM.....	25
1.3. Основные front-end методы SLAM .....	26
1.3.1. Normal Distributions Transform.....	28
1.3.2. Hector Mapping (scan-matching) .....	29
1.3.3. RGBD-SLAM .....	30
1.4. Выводы .....	31
2. Методы одновременной локализации группы роботов и построения карты	
33	
2.1. MultiAgent SLAM (MASLAM) .....	35
2.2. Multi-robot SLAM with unknown initial correspondence.....	37
2.3. Multi-Robot SLAM With Sparse Extended Information Filers.....	39
2.4. Cooperative Multi-Robot Map Merging Using Fast-SLAM.....	41

2.5. Grid-based Rao-Blackwellized Particle Filters Multi Robot SLAM .....	43
2.6. Multi-Robot SLAM using Condensed Measurements .....	45
2.7. Multi-Robot SLAM with Topological/Metric Maps.....	47
2.8. Decentralized Cooperative SLAM for Sparsely-Communicating Robot Networks: A Centralized-Equivalent Approach.....	49
2.9. Multi-robot visual SLAM using a Rao-Blackwellized Particle Filter .....	51
2.10. Multi-Robot Marginal-SLAM.....	53
2.11. Multi-Robot Range-Only SLAM by Active Sensor Nodes for Urban Search and Rescue .....	55
2.12. Multi-Robot Pose Graph Localization and Data Association from Unknown Initial Relative Poses via Expectation Maximization.....	58
2.13. A Neural Network-based Multiple Robot Simultaneous Localization and Mapping .....	60
2.14. Сравнение алгоритмов SLAM для группы роботов .....	63
2.15. Проблемы методов SLAM для группы роботов.....	65
2.16. Выводы .....	66
3. Список использованных источников .....	68

## ВВЕДЕНИЕ

Одним из наиболее интересных и изучаемых разделов в робототехнике в настоящее время является проблема одновременной локализации роботов и построения карты (SLAM). Ее можно определить, как проблему непрерывной оценки и уточнения карты окружающего мира на основе данных, получаемых с подвижного датчика (робота), и в то же время, определение положения этого датчика на карте. Речь идет о любых подвижных роботах – наземных или летающих, снабженных собственным вычислительным устройством и датчиками, позволяющими получать информацию из внешнего мира – лазерным дальномером, видеокамерой, одомером и др. [1]

Сейчас уже существует множество эффективных и применяемых на практике алгоритмов, решающих данную проблему для одиночного робота.

Однако, ввиду расширения сферы применения роботов, бурного развития технических средств и эволюции ПО изучение данного вопроса становится все более актуально, применительно к группам мобильных роботов. Группа представлена достаточно независимыми для автономной работы роботами, которые объединены в единую сеть для достижения поставленной цели. Именно в их кооперации лежит успех достижения этой цели, в нашем случае речь идет о кооперации при решении проблемы SLAM[6].

Как показывает практика, на данный момент не существует какого-либо конкретного достаточно надежного, протестированного и общепризнанного алгоритма навигации для групп роботов. В то же время, ввиду заинтересованности научного сообщества данной темой, постоянно появляются все новые разработки. В тоже время, большая часть новых попыток решения проблемы SLAM не находят широкого распространения, поскольку зачастую авторы либо не раскрывают в своих статьях подробности реализации их алгоритмов и не делятся исходными кодами, либо потому что эти алгоритмы сами по себе недостаточно проработаны, чтобы претендовать на роль законченного решения. Эти трудности наряду с разобщенностью групп

исследователей, отсутствием единой теоретической и практической базы для данной области робототехники существенно замедляет прогресс.

Данное исследование как раз призвано привнести свой небольшой вклад в данном направлении. Его целью, в первую очередь, является сбор, анализ и классификация существующих методов навигации групп роботов, а также их оценка. А во-вторых, реализация наиболее перспективных из этих методов на открытой и доступной сообществу платформе, их сравнение на основе проведенных экспериментов.

Перед тем как перейти непосредственно к рассмотрению Multi-SLAM, необходимо изучить существующие алгоритмы SLAM для одного робота, поскольку, как правило, они лежат в основе всех групповых методов.

## 1. МЕТОДЫ ОДНОВРЕМЕННОЙ ЛОКАЛИЗАЦИИ РОБОТА И ПОСТРОЕНИЯ КАРТЫ (SLAM)

Как было сказано, основными требованиями к роботу при решении задачи SLAM является его мобильность и способность получать информацию об окружающей среде с помощью датчиков. Как правило, в таких задачах рассматриваются роботы, способные передвигаться по относительно плоской поверхности: по естественному рельефу или внутри здания.

Общую схему SLAM можно представить в три простых этапа [1][14]:

1. Робот самостоятельно передвигается на новую позицию. При этом повышается неопределенность его положения (ввиду наличия шума в данных с сенсора движения – одометра – и ошибок округления). Данный этап математически описывается моделью движения, которая зависит от типа сенсора, типа платформы и способа передвижения робота.

Например:

$st = f(st - 1, ut) + N(0, \Lambda u)$  – модель движения, нелинейная функция, показывающая что новое положение робота  $st$  зависит от предыдущего положения  $st - 1$ , сигнала управления  $ut$  и Гауссова шума  $N$

2. Робот получает новые данные с сенсора. Данный этап описывается моделью прямых наблюдений, которая зависит от типа сенсора и его характеристик.

Например:

$zt, i = h(st, \theta ct, i) + N(0, \Lambda z)$  – модель наблюдений показывает, что текущее измерение сенсора  $zt, i$  зависит от положения робота  $st$ , ориентиров  $\theta ct, i$  и Гауссова шума  $N$

3. Робот уточняет свое положение и карту, путем интеграции по определенному алгоритму старых и только что собранных данных. Данный этап является ключевым в SLAM и описывается конкретным алгоритмом, вид которого существенно зависит от лежащего в основе механизма обработки и интеграции данных с датчика и отсеивания шумов.

## 1.1. Классификация

В соответствие с современным положением дел, концептуально все методы можно разделить на две взаимосвязанные группы front-end и back-end алгоритмы [13].

1. Back-end ~~фильтрующие~~ методы выполняют построение карты и локализацию по всей совокупности данных, полученных как по модели движения робота, так и по модели наблюдения. Они имеют под собой сложный вероятностный аппарат, и, как правило, основаны на алгоритмах фильтрации. back-end методы известны достаточно давно, первые их них разрабатывались в условиях отставания технической базы. Именно поэтому они хорошо справляются с зашумленными или неточными данными с несовершенных сенсоров. Многие из них предполагают, как самостоятельное использование, так и в совокупности с front-end. К ним относятся:
  - a. Непосредственно фильтрующие методы
  - b. Методы оптимизации графов передвижения
  - c. Методы, основанные на машинном обучении
2. Front-end ~~сглаживающие~~ методы нацелены, в первую очередь, на обработку данных, получаемых по модели прямых наблюдений. Построены на методах минимизации ошибки – сглаживания. Они могут служить как предварительным этапом работы для back-end методов, так и использоваться самостоятельно, при наличии высокоточных сенсоров. Именно наличие таких сенсоров помогает не использовать сложный математический аппарат в данных алгоритмах без потери качества результата. К ним относятся:
  - a. Методы, использующие в качестве входных данных последовательности сканов с лазерного дальномера или аналогичного высокоточного устройства.



- б. Методы, использующие последовательности изображений с камеры. Они основаны на алгоритмах компьютерного зрения.

Кроме того, все методы можно разделить в зависимости от того, на каком их этапе выполняется алгоритм SLAM - онлайн или оффлайн [14].

1. Оффлайн методы выполняются уже на заранее собранном наборе данных и предназначены для построения траектории движения робота и карты уже после завершения активной фазы деятельности робота. Их главный плюс – возможность построения очень точной карты ввиду отсутствия ограничений по времени и вычислительным ресурсам. В большинстве реальных задач, когда требуется принятие решений роботом в реальном времени на основе имеющейся карты, такие методы неприменимы.
2. Онлайн методы работают параллельно передвижению робота и сбору данных. Они применимы в случаях, когда основное требование к роботу – автономность и возможность самостоятельно принимать решения. Далее рассматриваются в основном такие методы.

Наконец, существует классификация методов по типу используемой карты[13].

1. Карта представлена набором ориентиров на местности (предметов особой формы, углов, уникальных текстур и т.д.)
2. Карта в виде сетки с бинарными ячейками (где 1 – в данной области пространства находится препятствие, 0 – препятствие отсутствует)
3. Карта в виде графа перемещений робота (где – ребра – вектора перемещений, вершины – наблюдения сенсора из данной точки)

Стоит отметить, что исторически первыми появились методы с картой, представленной ориентирами и использующее фильтрацию в своей основе (EKF – Расширенный фильтр Калмана, FastSLAM). Сейчас все более широкое распространение получают методы с картой в виде сетки. Несмотря на большие затраты по хранению и обработке таких карт, их гораздо проще строить из сырых

данных с лазерного дальномера и поддерживать, благодаря существованию большого набора вариантов решения проблемы сопоставления данных. [6, стр. 18] Кроме того, карты такого рода могут быть успешно адаптированы под любой класс алгоритмов SLAM.

## 1.2. Основные back-end методы SLAM

Рассмотрим основные наиболее популярные алгоритмы. Стоит заметить, что на данный момент существует большое множество модификаций представленных методов, однако все они в своей сути несут несколько базовых идей. Эти идеи можно разделить на три группы:

1. Представление карты в виде сетки ИЛИ в виде совокупности ориентиров.
2. Хранение одного экземпляра карты ИЛИ использование фильтра частиц для хранения множества экземпляров карты.
3. Использование математического аппарата расширенного фильтра Калмана (EKF) ИЛИ фильтра частиц.

Как правило, в наиболее успешных методах вместе с EKF используется карта из ориентиров, а с фильтрами Байеса частиц – сеточная карта.

### 1.2.1. Extended Kalman Filter SLAM

Классический метод решения проблемы SLAM. Карта представлена в виде вектора координат ориентиров. В оригинальной интерпретации для обнаружения ориентиров используется лазерный дальномер, возможно применение камеры. Для измерения перемещения робота используется одометр.

Метод основан на расширенном фильтре Калмана[1]. Состояние фильтра хранится в двух переменных:

1.  $\mu_t = [s_t, \theta_1, \theta_2, \dots, \theta_N]$  – вектор математического ожидания, где  $s_t$  – положение робота,  $\theta_N$  – положения ориентиров
2.  $\Sigma_t$  - матрица ковариации, содержащая перекрестные ковариации все элементов  $u_t$

Как можно заметить, здесь используется предположение о том, что все ориентиры взаимосвязаны.

Этапы работы фильтра [8][1][14]:

1. Движение робота, предсказание нового положения робота по данным одометрии. Иными словами, обновление переменных:

$$\begin{aligned}\hat{\mu}_t &= f(\mu_{t-1}, u_t) \\ \hat{\Sigma}_t &= A_t \Sigma_{t-1} A_t^T + G \Lambda_u G^T\end{aligned}$$

Где  $A_t$  – якобиан (вектор производных по всем переменным) модели движения  $f$ ,  $G$  – матрица проекции

2. Считывание данных с сенсора, уточнение состояния фильтра на основе повторно обнаруженных ориентиров.

$$\begin{aligned}K_t &= \hat{\Sigma}_t C_t^T (C_t \hat{\Sigma}_t C_t^T + \Lambda_z)^{-1} \\ \mu_t &= \hat{\mu}_t + K_t (z_t - h(\hat{\mu}_t, c)) \\ \Sigma_t &= (I - K_t C_t) \hat{\Sigma}_t\end{aligned}$$

$K_t$  - калмановское усиление,  $C_t$  – якобиан модели наблюдений  $h$

3. Обнаружение и добавление новых ориентиров в вектор состояния.

Среди недостатков EKF можно выделить:

1. Невозможность применения на местности с большим количеством ориентиров, ввиду сложности вычисления матрицы ковариации
2. Проблема с повторным обнаружением ориентиров
3. Накопление ошибки, в результате линеаризации моделей движения и наблюдения - вычисление якобиан на каждой итерации происходит относительно оценочного положения робота, а не действительного.

Некоторые из этих проблем решаются модификациями метода, например, использованием информационного фильтра вместо калмановского.

### **1.2.1. Sparse Extended information filter SLAM**

To do

### 1.2.2. FastSLAM

Другой широко распространенный метод SLAM, появившийся в начале 2000-х. Аналогично EKF используется карта в виде вектора ориентиров, в качестве сенсоров - лазерный дальномер (по умолчанию) или камера.

Алгоритм объединяет в себе фильтр частиц для оценки траектории движения робота и фильтры Калмана для оценки положений ориентиров.

В основе определения положений ориентиров лежит предположение о том, что ориентиры не взаимосвязаны, а, следовательно, могут оцениваться по отдельности. Полное состояние системы выражается через произведение вероятностей [3]:

$$\begin{aligned} p(s^t, \theta | z^t, u^t, n^t) \\ = p(s^t | z^t, u^t, n^t) \prod_k p(\theta_k | s^t, z^t, u^t, n^t) \end{aligned}$$

Иными словами, задача SLAM разбивается на задачу определения положения робота  $p(s_t)$  и  $k$  задач определения положений ориентиров  $p(\theta_k)$ , в зависимости от положения  $s_t$ , наблюдений  $z_t$ , сигнала управления  $u_t$ .

Для хранения состояния каждого ориентира  $\theta_k$  применяется отдельный фильтр Калмана, представленный математическим ожиданием положения ориентира  $\mu_{ti}$  и матрицей ковариации  $\Sigma_{ti}$ . Математическое ожидание, как правило, состоит из двух координат  $(x, y)$ . Матрица ковариации показывает взаимосвязь между данными компонентами.

Для оценки положения робота  $s_t$  применяется фильтр частиц Рэо-Блеквела. Каждая частица в фильтре представляет собственный вариант положения робота и карты (набор ориентиров). Таким образом, если в фильтре содержится  $N$  частиц и карта каждой представлена  $K$  ориентирами, то всего в фильтре  $K * N$  расширенных фильтров Калмана.

Принцип работы фильтра частиц заключен в том, чтобы на каждой итерации обновлять состояние карты, рассчитывать весовые коэффициенты частиц и на их основе выбирать только те, которые наиболее близко соответствуют реальному состоянию карты и положению робота.

Следует ввести два важных понятия [12, с. 447]:

- Предполагаемое (предлагаемое) распределение вероятности положения робота – вычисляется на основе модели движения робота, используя только данные одометрии и распределение с предыдущей итерации. На основе этого распределения производится выборка частиц в фильтре.
- Целевое распределение вероятности положения робота – соответствует действительному положению робота. Рассчитывается на основе наблюдений с сенсора и данных одометрии. Вычисление данного распределения в прямой форме и генерация на его основе выборки частиц невозможны, именно поэтому в FastSLAM используется фильтр частиц.

Этапы работы фильтра[14][3][2]:

1. На основе уже существующих частиц и предполагаемого распределения вероятности, генерируется новая выборка из  $N$  частиц.
  - a. Карта каждой частицы обновляется согласно измерениям одометрии. Иными словами, вычисляется новое состояние фильтров Калмана по воздействию  $u_t$  для каждого ориентира.
  - b. Считываются данные с сенсора. Для каждого повторно обнаруженного ориентира уточняется его состояние используя алгоритм работы фильтра Калмана (см. EKF SLAM).
$$p(\theta_k | s^t, z^t, u^t, n^t) \stackrel{\text{Markov}}{=} p(z_t | \theta_k, s_t, n_t) p(\theta_k | s^{t-1}, z^{t-1}, u^{t-1}, n^{t-1})$$
  - c. Новые обнаруженные ориентиры добавляются в карту. Поскольку в фильтре частиц имеется  $N$  вариантов карты, то для каждого из них производится собственная оценка обнаруженных ориентиров. Для различных частиц один и тот же ориентир может быть, как новым, так и повторно обнаруженным.
2. Рассчитываются веса частиц  $w_t$  и выбираются наиболее подходящие частицы.

$$w_t^{[m]} = \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{p(s_t^{[m]} | z^t, u^t, n^t)}{p(s_t^{[m]} | z^{t-1}, u^t, n^{t-1})} = \frac{p(z_t, n_t | s_t^{[m]}, z^{t-1}, u^t, n^{t-1})}{p(z_t, n_t | z^{t-1}, u^t, n^{t-1})}$$



Вес частицы показывает, насколько хорошо положение робота в данной частице соответствует целевому распределению вероятности положения робота. В идеальном случае, когда веса всех частиц равны, вся выборка частиц в точности соответствуют целевому распределению. При этом шаг генерации новой выборки для подгонки под целевое распределение не потребовался бы.

Существует версия FastSLAM 2.0 [2][4], в которой для расчёта предполагаемого распределения вероятности положения робота помимо сигнала управления (одометрии)  $u_t$  используются и данные с сенсоров  $z_t$ . Ввиду того, что сенсоры гораздо точнее одометров, это существенно добавляет эффективности методу, позволяет повысить точность выборки частиц и уменьшить их количество.

Из существенных недостатков можно назвать большие затраты памяти для обслуживания набора частиц, а также проблемы возникающие ввиду использования ЕKF, связанные с линеаризацией.

В то же время,

- наличие в памяти одновременно нескольких вариантов карты частично позволяет решить проблему повторного обнаружения ориентиров.
- Высокая, по сравнению с ЕFK, скорость работы ввиду независимости оценок ориентиров.

### 1.2.3. DP-SLAM

Метод, как и FastSLAM, основан на фильтре частиц. Карта представляется в виде бинарной сетки, где каждая единичная ячейка является препятствием, а каждая нулевая – свободным пространством. В качестве сенсора используется лазерный дальномер. Ключевая особенность метода - distributed particle (DP) mapping - представление множества вариантов карты, хранимых в частицах, в виде одного общего дерева, что позволяет обрабатывать достаточно большие сеточные карты при использовании небольшого количества памяти.

Здесь элементарными частями карты являются не ориентиры, а отдельные сканы местности (размер скана зависит от поля видимости лазерного дальномера). Общее состояние карты можно описать как произведение распределений вероятностей всех сканов:

$$P_i = \prod_k P_{\mathcal{L}}(\delta(i, k) | s_i, m),$$

Где  $P_L$  – нормальное распределение вероятности скана,  $\delta(i, k)$  – разница между ожидаемым и полученным сканом сенсора  $k$  для частицы  $i$ , при условии, что дана карта  $m$  и положение робота  $s_i$ .

Алгоритм имеет такие шаги [9][10][12]:

1. Генерация нового набора частиц, каждая из которых содержит собственную карту и оценку положения робота. В качестве источника частиц служит предполагаемое распределение, основанное на нормально распределенном шуме модели движения и предыдущее положение робота.
2. Определение нового положения робота на основе модели наблюдения и одометрических данных:

$$x_i = a_x * x + b_x + \mathcal{N}(0, \sigma_x)$$

$$y_i = a_y * y + b_y + \mathcal{N}(0, \sigma_y)$$

$$\theta_i = a_\theta * \theta + b_\theta + \mathcal{N}(0, \sigma_\theta)$$

где  $a$ ,  $b$  - линейные коррекции движения,  $N$  – нормальный шум, которые зависят от конкретной платформы робота и одометра.

3. Вычисление веса частицы на основе, с одной стороны - модели наблюдения и данных с сенсора, а с другой - распределения вероятности состояния карты. Он показывает, насколько хорошо данные с сенсора совпадают с имеющейся картой при данном положении робота. Т.е. чем меньше разница, тем больший вес имеет частица.
4. Нормализация весов частиц

Вторая версия алгоритма DP-SLAM 2.0 нацелена на оптимизацию вычислительного процесса и сокращение количества операций. Другое нововведение – вероятностное распределение карты [10]. Поскольку при дискретном представлении карты не принимаются во внимание прозрачные и частичные препятствия, это вносит дополнительную ошибку в расчеты. Здесь каждая ячейка хранит свою «прозрачность»  $\rho$ . А вероятность того, что лазерный луч, испускаемый сенсором, будет остановлен именно в этой ячейке, определяется экспоненциальным распределением

$$P_c(x, \rho) = 1 - e^{-x/\rho}$$

перемноженным на распределения всех предыдущих ячеек на пути лазерного луча:

$$P_c(\mathbf{x}, \boldsymbol{\rho}) = \sum_{i=1}^n P_c(x_i, \rho_i) \prod_{j=1}^{i-1} (1 - P_c(x_j, \rho_j)).$$

Тогда общее состояние карты представляется следующим распределением:

$$\sum_{i=1}^n P_{\mathcal{L}}(\delta_i | \text{stop} = i) P(\text{stop} = i).$$

Из преимуществ алгоритма можно отметить [9]:

- Решение проблемы вырождения частиц, присущей иным методам, благодаря иному подходу к определению весов частиц. Проблема заключается в отбрасывании частиц, содержащих наиболее близкие к действительности карту и положение робота, но имеющим маленький вес на этапе генерации новой выборки.

- Возможность работы с большими картами высокого разрешения с сохранением скорости и точности
- Отсутствие проблемы повторного определения ориентиров, и тем самым решение проблемы замыкания циклов.

#### 1.2.4. Grid-based FastSLAM и Gmapping

Данный метод является глубокой модификацией FastSLAM, направленной на оптимизацию скорости работы фильтра частиц, уменьшения количества частиц и устранение известных проблем [18].

Метод рассматривается здесь отдельно, поскольку является актуальным и применяемым в настоящее время. Кроме того, имеется его отрытая и готовая к применению реализация в операционной системе ROS – Robot Operating System. Также собраны большие экспериментальные данные применения метода.

Карта представляется в виде сетки, в качестве сенсора используется высокоточный лазерный дальномер. Сам алгоритм состоит из двух частей [18][16]:

1. Хорошо известный фильтр частиц, работающий по принципу Fast-SLAM
2. front-end алгоритм, который отвечает за соотнесение с известной картой новых сканов, получаемых с сенсора - scan-matching

В основе Scan-matching лежит поиск (путем градиентного спуска) такого положения робота  $x$  в заданной окрестности, в котором вероятность получения данного скана  $z_t$  максимальна:

$$\hat{x}_t^{(i)} = \underset{x}{\operatorname{argmax}} p(x \mid m_{t-1}^{(i)}, z_t, x_t'^{(i)})$$

Где  $x_t'^{(i)}$  – положение робота только по данным одометрии,  $z_t$  – данные с сенсора,  $\hat{x}_t^{(i)}$  – искомое уточненное положение,  $m_{t-1}^{(i)}$  – известная карта

Ключевые оптимизации метода [18]:

- Улучшенный этап генерации новой выборки частиц, благодаря уточнению положения робота на основе не только одометрии, но и данных с лазерного дальномера (scan-matching). В связи с этим представлен новый метод вычисления распределения вероятности положения робота. Преимущество – частиц требуется меньше при большей точности.

- Улучшенный метод взвешивания и отсева плохих частиц (ресемплинга), решающий проблему вырождения частиц. Ресемплинг производится не после каждой итерации фильтра, а при необходимости.

Стоит отметить, что в оригинальном методе не уточняется способ хранения набора сеточных карт в памяти, разрешается применение стороннего метода, например, distributed particle mapping из DP-SLAM).

Существует расширение метода для использования в качестве сенсора камеры или RGBD - сенсора.

### 1.2.5. Graph-based SLAM

Алгоритм относится к группе оффлайн методов. В его основе лежит представление карты в виде графа – сети Байеса. В качестве сенсора применяется лазерный дальномер. Вершины графа представляют положения робота  $x$ , в которых производятся наблюдения, а дуги – вероятности перехода между этими положениями, рассчитываемые на основе данных одометра и сенсора как распределение вероятности перехода из предыдущей вершины графа в текущую.

Задача разбивается на подзадачи [13]:

1. Построение графа в процессе передвижения робота по данным с одометра и сенсора. Разрешается использовать как сырые данные с сенсора, так и scan-matcher алгоритм, позволяющий уже на данном шаге утонять положение робота.
2. Оптимизация графа и построение карты местности. Оптимизация может выполняться, например, методом максимального правдоподобия [20]. Граф считается оптимальным, если для каждой его вершины ошибка между ожидаемыми данными с сенсора и реальными данными минимальна:

$$l_{ij} \propto [z_{ij} - \hat{z}_{ij}(x_i, x_j)]^T \Omega_{ij} [z_{ij} - \hat{z}_{ij}(x_i, x_j)].$$

$$e_{ij}(x_i, x_j) = z_{ij} - \hat{z}_{ij}(x_i, x_j).$$

$$F(x) = \sum_{\langle i,j \rangle \in C} \underbrace{e_{ij}^T \Omega_{ij} e_{ij}}_{F_{ij}},$$

Где  $l_{ij}$  – логарифмическая функция правдоподобия,

$e_{ij}(x_i, x_j)$  – ошибка между ожидаемым наблюдением  $\hat{z}_{ij}$  и реальными данными с сенсора  $z_{ij}$  в текущей вершине,

$\Omega_{ij}$  – информационная матрица текущего ребра графа,

$F_x$  – искомая функция, которую требуется минимизировать.

Сама же минимизация выполняется итеративно, например, методом Гаусса-Ньютона.

Сильные стороны алгоритма в том, что

- Не используются какие-либо фильтры, а значит отсутствуют и присущие им ошибки линеаризации.
- Кроме того, алгоритм хорошо подходит для случаев с трехмерной картой.
- Использует полные данные о движении робота, а, следовательно, дает результаты высокой точности, хорошо справляется с замыканием циклов.

Главные же его минусы – трудность применения онлайн, возрастание сложности вычислений с увеличением пройденного роботом пути.

Стоит отметить, что в настоящее время уже появились достаточно мощные роботизированные платформы и достаточно быстрые алгоритмы оптимизации для выполнения Graph-SLAM в реальном времени.



### **1.2.6. Neural SLAM**

To do

### 1.3. Основные front-end методы SLAM

В общем случае методы этой группы ориентированы на оценку траектории движения робота по группе последовательных сканов с лазерного дальномера или по набору ориентиров, полученных с изображений с камеры.

Эти методы не используют сложных вероятностных моделей и очень ограниченно используют информацию, полученную на предыдущих шагах. front-end алгоритмы можно использовать как отдельно, так и в связке с back-end для достижения лучшего результата.

Например, к этой группе можно отнести такие методы компьютерного зрения, как:

- **ICP, IDC** [5] [16] — самые простые алгоритмы для установления соответствия между облаками точек с двух сканов или двух групп ориентиров. В основе данного алгоритма лежит метод последовательных приближений для минимизации суммы квадратов расстояний между соответствующими точками двух сканов.

Первый шаг алгоритма – поиск соответствующих пар точек между текущим сканом и опорным. Для каждой точки текущего скана выбирается ближайшая по Евклидовому расстоянию из опорного скана. Основным ограничением ICP является факт, что ближайшая точка в общем случае не соответствует той же самой точке сканируемой поверхности, особенно если два скана выполнены с отдаленных друг от друга позиций. Тем не менее, в успешных приложениях алгоритма ICP, он сходится к некоторому полезному решению.

Самой трудоемкой задачей является поиск точек на двух сканах, соответствующих одной точке на поверхности реального объекта. Одним из вариантов решения этой задачи является использование взвешенных пар точек, при этом наибольший вес имеют пары, наиболее вероятно соответствующие одной точке поверхности.

- *SIFT, SURF* [38] — методы построения пути робота на основе поиска особенностей и установления соответствия между ними на двух изображениях с сенсора - камеры.

Сопоставление каждой новой найденной особенности уже известным особенностям и поиска ее пары основано на вычислении евклидового расстояния. Затем из всего списка соответствий выбираются только такие, которые инвариантны к масштабу, положению и повороту. Также отбрасываются неоднозначные соответствия. На последнем этапе вычисляется вероятность того, что пара особенностей представляет один объект исходя из точности соответствия и числе вероятных ложных соответствий на изображении.

Затем, по координатному преобразованию набора особенностей между кадрами с камеры строится траектория движения робота, а сами особенности становятся основой для карты.

Рассмотренные выше методы, скорее, можно отнести к визуальной одометрии, чем к решению задачи SLAM, т.к. они определяют перемещение робота между двумя точками сканирования без учета шума сенсора, что приводит к накоплению ошибки. Но в то же время они могут быть применены, как составная часть для back-end методов.

Ниже рассмотрим несколько самостоятельных решений, ориентированных на сеточные карты и сопоставление сканов (scan-matching).

### 1.3.1. Normal Distributions Transform

Алгоритм соотнесения сканов лазерного дальномера, предназначенный специально для SLAM с картой в виде сетки, и без использования одометрии.

Normal Distributions Transform – преобразование нормальных распределений. Ключевым элементом в этом алгоритме является представление опорного скана. Вместо непосредственного сопоставления точек текущего скана с опорным, моделируется вероятность нахождения точки рельефа в определенном положении с помощью линейной комбинации нормальных распределений. Такой подход позволяет получить кусочно-гладкую функцию, являющуюся представлением опорного скана, имеющую непрерывные первые и вторые производные. Это дает возможность использовать стандартные методы оптимизации для поиска соответствия по кросс-корреляционной функции, например, метод Ньютона. Дополнительным преимуществом построения нормальных распределений является возможность использовать накопленную информацию в том же виде без увеличения вычислительной сложности, поскольку информация с сенсора обрабатывается единожды при поступлении.

К недостаткам данного алгоритма следует отнести сильную зависимость сходимости от начального приближения и дискретизации пространства.

### 1.3.2. Hector Mapping (scan-matching)

В отличие от предыдущих, данный алгоритм использует в качестве модели рабочей среды всю накопленную карту. Карта представлена в виде сетки занятости, где значение функции карты  $M(Pm) = 1$ , если в соответствующей ячейке находится объект, и  $M(Pm) = 0$ , если ячейка свободна. Чтобы смоделировать вероятностный характер распределения точек, относящихся к объектам на карте, авторы предлагают использовать билинейную интерполяцию. При этом функция карты становится непрерывной, и легко вычислить ее градиент в любой точке. В качестве меры совпадения скана и карты используется среднеквадратическая ошибка всех точек скана [17]. Тогда для оценки положения  $\xi$  \* мобильного робота необходимо воспользоваться методом наименьших квадратов:

$$\xi^* = \arg \min_{\xi} \sum_{i=1}^n [1 - M(S_i(\xi))]^2,$$

$S_i(\xi)$  – координаты  $i$ -й точки скана, преобразованной в систему координат помещения, если считать, что скан получен из положения  $\xi$ .

Учитывая простоту вычисления градиента, минимизация данной функции выполняется методом Гаусса-Ньютона. Однако оптимизируемая функция имеет локальные минимумы, поэтому авторы предложили искать решение в несколько этапов, каждый из которых имеет дело с картой различного разрешения, постепенно выбирая все меньшие разрешения, эта оценка улучшается до тех пор, пока не достигнет необходимой точности.

Наконец, применительно к задаче SLAM, оценка положения мобильного робота обновляется с использованием фильтра Калмана или любого другого back-end метода. Накапливаемая ошибка в методе мала вследствие использования всей карты для сопоставления скана. В качестве недостатка можно назвать необходимость построения и обновления нескольких карт с различным разрешением, что повышает требования к вычислительным ресурсам.

Следует отметить, что имеется свободная реализация метода для ROS, что повышает его привлекательность для практического применения.

### 1.3.3. RGBD-SLAM

Гибридный метод построения карт на основе сенсора - RGB-D камеры (например, Kinect). Данный сенсор совмещает в себе обычную камеру и лазерный дальномер, таким образом один кадр представляет из себя цветной трехмерный скан местности.

Карта в методе представляется в виде трехмерной сетки, но для определения новых положений робота используются представление его пути в виде графа. Каждая вершина графа представлена положением робота, а каждое ребро – сканом местности. При этом для оптимизации данного графа применяются не сырые данные с сенсора, а обнаруженные на изображениях ориентиры. Для этой цели служит описанный выше алгоритм SIFT [19].

В качестве back-end составляющей, которая выполняет построение оптимального пути оффлайн можно использовать, к примеру, алгоритм g2o.

Таким образом на выходе алгоритм дает очень детализированную карту, однако ввиду большой вычислительной сложности и дороговизны оборудования его применение для навигации групп роботов на данный момент не представляет интереса.

## 1.4. Выводы

Рассмотрев вышеописанные алгоритмы можно сделать вывод, что задача SLAM для одного робота успешно решается в настоящее время. Все большую популярность набирают современные методы, лишенные недостатков «классических» расширенного фильтра Калмана и фильтра частиц. Большое развитие получают сглаживающие front-end алгоритмы, быстрые и эффективные, появляется множество гибридных методов, совмещающие в себе только лучшие черты как front-end, так и back-end алгоритмов.

Особенно перспективны те из них, где карта имеет сеточное представление, а в качестве сенсора используется лазерный дальномер.

Предпосылками к этому являются:

- Во-первых, высокий уровень развития измерительной техники – на данный момент лазерные сенсоры дают очень точные измерения, по сравнению с камерами и одометрами.
- Во-вторых, хорошей степенью изученности алгоритмов оптимизации, которые используются для построения сеточных карт.
- В-третьих, большей вычислительной эффективностью подобных методов при более высокой точности, поскольку не требуется предобработка лазерных сканов, в отличие от фотографий.
- Наконец, существует и применяется множество открытых реализаций данных алгоритмов.

Отдельно можно отметить методы на основе построения графа передвижения, которые показывают результаты высокой точности. Это тот самый пример гибридного алгоритма, поскольку отдельные ребра графа, содержащие измерения с сенсора, как правило, представлены локальными сеточными картами. Однако, основной вопрос, который встает при их использовании – возможность выполнения оптимизации графа в реальном времени, ввиду большой ресурсоемкости данной операции.

Что касается карт на основе ориентиров, то наилучшие результаты удастся достичь при использовании совместно с фильтрами. Однако фильтры достаточно сложны и требовательны к ресурсам, и, кроме того, имеют ряд трудноразрешимых проблем. Наконец, актуальными остаются вопросы с повторным обнаружением ориентиров и ассоциацией данных.

В связи со сказанным, в дальнейшем упор в этом исследовании делается именно на алгоритмы с сеточным и графовым представлением и с минимальным применением фильтрации.



## **2. МЕТОДЫ ОДНОВРЕМЕННОЙ ЛОКАЛИЗАЦИИ ГРУППЫ РОБОТОВ И ПОСТРОЕНИЯ КАРТЫ**

Целью данной работы является изучение именно алгоритмов SLAM для групп роботов. Все алгоритмы, попадающие в данную категорию, являются по сути модификациями методов навигации для одного робота. Эти модификации нацелены на решение ряда возникающих вызовов [21][6], среди которых:

- По каким каналам роботы должны обмениваться информацией
- Какую топологию должна иметь сеть, связывающая группу
- Какой информацией о состоянии карты и своем положении должны обмениваться роботы
- Как должна происходить интеграция полученной от других роботов данных

В случае успешного решения этих проблем можно добиться ряда преимуществ:

- Большая скорость построения карты по сравнению с одним роботом ввиду наличия нескольких вычислительных устройств и нескольких независимых источников данных
- Большая точность алгоритма по тем же причинам
- Устойчивость алгоритма к воздействию отрицательных факторов на отдельных роботов группы (поломка, плохие внешние условия)

Один из приоритетных вопросов, требующих решения является топология сети, связывающей роботов. Возможны централизованная и распределенная сети. Централизованная система более проста в реализации, позволяет наладить лучшую координацию между роботами, а также менее требовательна в плане ресурсов к периферийным устройствам – роботам. Однако среди недостатков выделяются высокие требования к каналу передачи данных, необходимость наличия априорной информации о положении роботов. Кроме того при больших

размерах исследуемого пространства либо при большом количестве роботов система становится труднореализуемой на практике.

Децентрализованные системы имеют свои недостатки: большая степень неопределенности, проблема взаимного обнаружения роботов, сложность координации. Однако, данная схема позволяет строить большие карты большими группами роботов, при этом достигается высокая стойкость к шумам и параллельное выполнение задач.

Описанные качества не сегодняшний момент обеспечили большое распространение децентрализованных методов среди специалистов по робототехнике. Далее рассматриваются преимущественно такие алгоритмы [6].

## 2.1. MultiAgent SLAM (MASLAM)

Один из первых разработанных методов и являющийся модификацией EKF, который, однако, так и не был до конца реализован и проверен в реальных условиях. Авторы, как и в оригинальном методе для одного робота, используют в данном случае карту, состоящую из ориентиров и предположение, что все ориентиры взаимосвязаны.

Пусть состояние одного робота описывается вектором математических ожиданий положения ориентиров  $x$  и матрицей ковариации  $P$ .

Базовый алгоритм на фильтре Калмана расширен следующими шагами [22]:

1. Получение информации от других роботов: вектора состояния  $y$  и матрицы ковариации  $S$
2. Полученные данные рассматриваются как совершенное наблюдение местности. Обновление положения известных ориентиров, используя стандартную модель наблюдения

$$y = Hr * x + vk$$

3. Добавление новых ориентиров из  $y$  и  $S$  к своему набору ориентиров
4. Обновление своего состояния

$$x_k = \hat{x}_k + K_k(y_k - H_r \hat{x}_k)$$

5. Выполнение шагов согласно алгоритму EKF-SLAM

Предполагается, что связь между роботами идеальна и роботы способны определять ошибочные сообщения. Кроме того, принято допущение, что все роботы начинают свое движение с заранее известных мест и с одинаковым углом поворота относительно друг друга. А все обнаруживаемые особенности имеют устойчивые идентификаторы, и роботы способны однозначно определять их.

Для уменьшения передачи данных решено передавать не полное состояние  $y$  и  $S$  робота, а только новые обнаруженные ориентиры и дополнительную

информацию, идентифицирующую эти ориентиры, чтобы другие роботы могли интегрировать их в свою карту.

Проблема вычисления положения роботов относительно друг друга в теории решается, если роботы устанавливают прямой визуальный контакт. Тогда робот может вычислить смещение карты другого робота и интегрировать ее в свою.

$$s_{oa}^b = s_b^b + m_a^b - s_a^a$$

Где  $s_{oa}^b$  – положение ориентира в координатах первого робота,  $s_b^b$  – положение первого робота,  $m_a^b$  – расстояние между роботами,  $s_a^a$  – положение второго робота.

При реализации данного алгоритма авторы решали проблему взаимного обнаружения роботов путем использования специального выделенного сервера, через который проходят все сообщения и который хранит актуальную информацию о положении всех роботов и о реальном положении особенностей карты. Этот механизм позволяет симулировать прямой визуальный контакт между роботами и применять вышеописанную формулу для определения взаимного положения. Также сервер помогает идентифицировать наблюдаемые особенности.

Алгоритм имеет вычислительную сложность  $O(r^2 * N^3)$  и затраты памяти  $O(r * P)$ , где  $r$  – количество роботов и  $P$  – периметр карты. Сведения о точности работы алгоритма и поддерживаемых размерах карты авторами не указаны.

Возникает сомнение в практической применимости данного метода в текущем виде в реальных условиях. Особенно большие вопросы вызывает наличие централизованного сервера в качестве решения проблемы повторного обнаружения ориентиров и взаимного обнаружения роботов. Кроме того, лежащий в основе ЕКФ накладывает свои ограничения на размер карты и точность результата.

## 2.2. Multi-robot SLAM with unknown initial correspondence

Очередной алгоритм основанный на EKF и карте, представленной набором ориентиров. В отличие от [22], способен работать в условиях абсолютной неопределённости, когда о начальных положениях роботов ничего неизвестно.

Для того, чтобы роботы могли однозначно обнаруживать друг друга и определять взаимное положение, на них наносятся специальные цветные метки. Расстояние между роботами  $\rho_m$  определяется с помощью лазерного дальномера как среднее значение измерений двух роботов с учетом соответствующих нормальных шумов  $\sigma$ .

$$\rho_m = \sigma_\rho^2 \left( \frac{{}^1\rho_m}{\sigma_{1\rho}^2} + \frac{{}^2\rho_m}{\sigma_{2\rho}^2} \right), \quad \frac{1}{\sigma_\rho^2} = \frac{1}{\sigma_{1\rho}^2} + \frac{1}{\sigma_{2\rho}^2}$$

Угол  $\theta$  определяется как разница между углами двух роботов с учетом нормального шума  $\pi$  по формуле

$$\theta = \pi + {}^1\theta_2 - {}^2\theta_1$$

Обмен картами между роботами выполняется по шагам [23]:

1. Робот преобразует положение полученных ориентиров в свою координатную плоскость, основываясь на вычисленных  $\rho_m$  и  $\theta$
2. Карты объединяются, путем простого совмещения двух векторов математических ожиданий положения ориентиров:

$$\mathbf{X}' = [G_1 \mathbf{X}_1^T \quad G_2 \mathbf{X}_2^T]^T$$

При этом матрица ковариации рассчитывается по формуле:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{11}\mathbf{H}_1^T \\ \mathbf{H}_1\mathbf{P}_{11} & \mathbf{H}_1\mathbf{P}_{11}\mathbf{H}_1^T + \mathbf{H}_2\mathbf{P}_{22}\mathbf{H}_2^T + \mathbf{\Gamma}_2\mathbf{R}\mathbf{\Gamma}_2^T \end{bmatrix}$$

Где  $\mathbf{\Gamma}$  и  $\mathbf{H}$  – используемые в фильтре Калмана Якобианы

3. Удаляются повторяющиеся ориентиры. Ориентиры сравниваются любым доступным методом. Например, методом ближайшего соседа или JSBV (модификация метода ветвей и границ).

В качестве меры похожести используется расстояние Махаланобиса (Евклидово расстояние с учетом корреляции между переменными). Кроме того, принимается во внимание расстояние от роботов до ориентиров – чем оно меньше, тем точнее определено положение ориентиров.

Алгоритм слияния карт имеет вычислительную сложность от  $O(n^2)$  в худшем случае до  $O(n \log n)$  в среднем.

По заявлению авторов алгоритм был протестирован в реальных условиях (на карте площадью 4800 м<sup>2</sup>) и дает приемлемые по точности результаты для небольших групп роботов (как минимум для 2х). Из недостатков можно выделить возможность обмена данными между роботами только при непосредственном контакте и только при использовании специальных цветковых маркеров на корпусе, а также все недостатки, свойственные EKF.

## 2.3. Multi-Robot SLAM With Sparse Extended Information Filers

Метод основан на информационном фильтре, который по сути является модернизированный фильтром Калмана (вместо представления данных в обычных координатах используется информация Фишера).

Карта представлена ориентирами. Проблема определения взаимного расположения роботов решается путем применения нового алгоритма сравнения карт [27]:

1. В локальной карте робота  $\langle H_t^k, b_t^k \rangle$  рассчитывается взаимное расположение ориентиров в небольшом радиусе. Например, берется триплет ориентиров и вычисляются расстояния и углы между ними. Эта информация сохраняется в древовидной структуре (SR-дерево).
2. Данная структура затем передается между роботами и может легко сравниваться с подобными структурами, представляющими другие локальные карты.
3. Найденные совпадения затем используются как начальные гипотезы о взаимном расположении роботов. По результатам сравнения из наиболее вероятной гипотезы вычисляется угол  $R$  и расстояние  $D$  между роботами и их карты интегрируются.

$$\begin{aligned} H_t^{k \leftarrow j} &= R^T H_t^j R \\ b_t^{k \leftarrow j} &= (b_t^j + H_t^j D^T) R^T \end{aligned}$$

4. Определяются повторяющиеся особенности (также по сравнению SR-деревьев) и происходит их свертка:

$$\begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{pmatrix} \rightarrow \begin{pmatrix} h_{11} & h_{12}+h_{14} & h_{13} \\ h_{21}+h_{41} & h_{22}+h_{42}+h_{24}+h_{44} & h_{23}+h_{43} \\ h_{31} & h_{32}+h_{34} & h_{33} \end{pmatrix}$$

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} \rightarrow \begin{pmatrix} b_1 \\ b_2+b_4 \\ b_3 \end{pmatrix}$$

Локальная часть алгоритма полностью соответствует алгоритму информационного фильтра.

Несомненным плюсом является отсутствие необходимости устанавливать дополнительные пометки или сенсоры на роботов для определения прямых контактов между ними. Проблемой метода является то, что карты роботов должны достаточно сильно коррелировать, чтобы их интеграция происходила успешно. В реальных условиях это не всегда возможно, поэтому метод не выглядит достаточно устойчивым. Возникает вопрос также об эффективности вычислений: с одной стороны, информационный фильтр работает быстрее ЕKF (интеграция наблюдений происходит за линейное время, а не кубическое), с другой – дополнительные шаги при расчете SR дерева так же требуют вычислительных ресурсов.

Экспериментально алгоритм был успешно проверен на данных, предварительно собранных в Виктория Парке в Сиднее, площадь которого приблизительно равна 0.12 км<sup>2</sup>. Один общий набор данных был разбит на 8 частей, что симулирует советующую группу роботов.



## 2.4. Cooperative Multi-Robot Map Merging Using Fast-SLAM

Группа алгоритмов, основанных на оригинальном FastSLAM и использующих для представления карты набор ориентиров. В качестве сенсора применяется камера. Дополнительно на роботе может быть установлен лазерный дальномер для определения препятствий или дополнительного представления карты. Предполагается, что в начальный момент времени роботы ничего не знают о положениях других роботов.

Один из методов данной группы [24] использует допущение о том, что особенности помечены специальными маркерами для упрощения их обнаружения по фотографиям с камеры. Кроме того, на каждом роботе установлены маркеры другого вида для отделения роботов от окружающей среды и определения ими взаимного расположения.

При прямом контакте роботы обмениваются сообщениями, которые включают [24]:

1. Карту, которая в фильтре частиц представлена набором гипотез  $m_{other,i}$ . Каждая гипотеза, в свою очередь представлена собственными оценками положений ориентиров (математическим ожиданием  $p_{other,i}$  и ковариационной матрицей  $\Sigma_{other,i}$ ).
2. Положение робота  $p_{other}$
3. Данные с сенсора о взаимном положении двух роботов  $z_{other,self} = \{l_{other,self}, \theta_{other,self}\}$

Этап объединения двух карт состоит из шагов [24]:

1. Преобразования полученной карты в собственную систему координат

$$\begin{bmatrix} p_{other,x,i'} \\ p_{other,y,i'} \\ 1 \end{bmatrix} = T \begin{bmatrix} p_{other,x,i} \\ p_{other,y,i} \\ 1 \end{bmatrix}$$
$$\Sigma_{other,i'} = T^T \Sigma_{other,i} T$$

Где  $T$  – матрица трансформации

$$T = \begin{bmatrix} \cos(tr_\theta) - \sin(tr_\theta) & tr_x \\ \sin(tr_\theta) & \cos(tr_\theta) & tr_y \\ 0 & 0 & 1 \end{bmatrix}$$

2. Удаления повторяющихся ориентиров, используя метод ближайшего соседа.

Алгоритм был протестирован на реальных данных и доказал свою пригодность. Для эксперимента использовались 2 робота и небольшая комната в качестве исследуемой местности (<25м<sup>2</sup>). Преимущества метода по скорости работы и точности определяются использованием фильтра частиц. Однако, как и в случае с предыдущими методами, для применения в реальных условиях требуется решение проблемы с взаимным определением положения роботов без использования специальных отметок.

## 2.5. Grid-based Rao-Blackwellized Particle Filters Multi Robot SLAM

Метод, описанный в [25], основан на Grid-based FastSLAM, то есть использует фильтр частиц и сеточное представление карты. Для сбора данных применяется лазерный дальномер и одометр. Это позволяет принципиально избавиться от проблемы повторного обнаружения особенностей на местности. Обмен данными между роботами производится при их прямом контакте, однако авторы не акцентируют внимания на том как именно это должно происходить.

Метод описанный в [26] является модификацией [25]. Здесь проблема прямых контактов роботов решается с помощью установки дополнительной омни-камеры и специальных светоотражающих маркеров на корпус робота. Кроме того, авторы приводят в статье гораздо более полные экспериментальные данные.

Алгоритм интеграции карт роботов включает следующие этапы [26]:

1. Передача сообщения, содержащего изменения на карте с момента последней встречи роботов. Их можно представить, как последовательность одометрических измерений с соответствующими им сканами лазера.
2. Преобразование карты в локальную систему координат. Для этого используются угол поворота  $\theta_{ij}$ , полученный по данным омни-камеры, и расстояние с лазерного дальномера  $\rho_{ij}$ . Учитываются также возможные гауссовы шумы измерений. Относительное положение между роботами представляется математическим ожиданием

$$p_{R_i R_j} = [\rho_{ij} \cos \theta_{ij}, \rho_{ij} \sin \theta_{ij}, \pi + \theta_{ij} - \theta_{ji}]$$

и матрицей ковариации

$$\begin{aligned}
P_{11} &= \sigma_{\rho_{ij}}^2 \cos^2 \theta_{ij} + \rho_{ij}^2 \sigma_{\theta_{ij}}^2 \sin^2 \theta_{ij} \\
P_{12} &= P_{21} = \frac{\sigma_{\rho_{ij}}^2 - \rho_{ij}^2 \sigma_{\theta_{ij}}^2}{2} \sin(2\theta_{ij}) \\
P_{13} &= P_{31} = -\rho_{ij} \sigma_{\theta_{ij}}^2 \sin \theta_{ij} \\
P_{22} &= \sigma_{\rho_{ij}}^2 \sin^2 \theta_{ij} + \rho_{ij}^2 \sigma_{\theta_{ij}}^2 \cos^2 \theta_{ij} \\
P_{23} &= P_{32} = \rho_{ij} \sigma_{\theta_{ij}}^2 \cos \theta_{ij} \\
P_{33} &= \sigma_{\theta_{ij}}^2 + \sigma_{\theta_{ji}}^2
\end{aligned}$$

Где  $\sigma^2$  – дисперсия нормального шума измерения

3. Интеграция полученных данных в карту. Каждый полученный скан карты расценивается как данные с собственного сенсора и обрабатываются фильтром частиц по стандартному алгоритму FastSLAM.

Пусть  $S1$  и  $S2$  – стартовые позиции роботов,  $F1$  и  $F2$  – положения роботов при встрече. Тогда процесс интеграции заключается в присоединении к данным, собранным роботом на интервале движения  $S1 - F1$ , данных, полученных другим роботом на интервале  $F2 - S2$ . То есть траектория второго робота присоединяется к траектории первого робота в обратном порядке. Данные с лазерного дальномера передаются без изменений.

4. После завершения интеграции полученного пути работа движение робота продолжается с точки  $F1$  в обычном порядке.

Несмотря на решение проблемы с ориентирами, очевидным недостатком метода [25] является количество передаваемой информации между роботами и необходимые вычислительные ресурсы на ее обработку, особенно в случае редких контактов. Помимо этого, авторы алгоритма явно указывают, что точность работы может быть существенно повышена при применении алгоритма замыкания циклов. Данная версия была протестирована на предварительно собранных данных размером до 2500 м2 и 4х роботах.

В то же время алгоритм [26] более устойчив для работы в реальных условиях, поскольку уменьшает количество передаваемой информации, а также не использует каких-либо допущений о взаимных положениях роботов. Он был экспериментально проверен на 2х роботах и реальной местности площадью 200 м2 и показал хорошие результаты.

## 2.6. Multi-Robot SLAM using Condensed Measurements

Метод является расширением Graph-based SLAM, где карта представляется в виде графа с вершинами – положениями робота в процессе движения - и ребрами – измерениями, полученными сенсором в данных положениях. Сенсор – лазерный дальномер.

В работе рассматриваются методы для уменьшения количества передаваемых между роботами и обрабатываемых данных, поскольку главными вызовами для алгоритмов, основанных на графах, являются: 1) зависимость размера графа от времени работы алгоритма, 2) возникновение проблем с интеграцией локальных графов разных роботов (ассоциацией данных).

Метод использует концепцию «уплотненных измерений» - между роботами передается только часть графа, необходимая для его интеграции, которая состоит из [28]:

- Последнего измерения с сенсора
- Последних  $N$  вершин графа, содержащих актуальные оценки  $N$  положений робота

Иными словами, каждое сообщение содержит один скан с сенсора и текущую оценку траектории робота и занимает не более 1.5 Кб.

Для того чтобы восстановить локальную карту из  $N$  сканов, требуется получить  $N$  сообщений со сканами и добавить их в собственный граф на основе самой актуальной оценки положения робота. Непосредственно интеграция сканов выполняется с помощью алгоритма, основанного на RANSAC и scan matching.

Кроме того, на каждом шаге роботы отсылают следующие сообщения:

- Список вершин из графа другого робота, которые данный робот уже интегрировал в свой граф
- Часть своего графа, содержащую новые вершины и некоторое количество вершин и ребер, которые уже были интегрированы другим роботом

Оптимизация графа выполняется с помощью алгоритмов g2o или SAM путем минимизации функции:

$$F(\mathbf{x}) = \sum_{\langle i,j \rangle \in \mathcal{C}} \mathbf{e}_{ij}^T \Omega_{ij} \mathbf{e}_{ij}$$

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} F(\mathbf{x})$$

Где  $e_{ij}$  – ошибка между предсказанным по модели движения положением робота  $\hat{z}_{ij}$  и данными с сенсора  $z_{ij}$ ,  $\Omega$  – информационная матрица измерения с сенсора

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}$$

Оптимизация может проводится на каждом шаге и не требует больших вычислительных мощностей.

На основании вышесказанного можно выявить такие преимущества метода, как ориентированность на практическую реализацию и применение, низкие требования к каналу передачи информации, высокая точность оценки карты, решение проблемы с взаимной оценкой положения роботов. По заявлениям авторов метод был реализован в виде пакета ROS.

Среди недостатков выделяется тот факт, что авторы не поясняют каким образом роботы должны обнаруживать друг друга при встречах.

Алгоритм экспериментально проверен на группах в 2, 4 и 8 роботов на местности размером более 500м<sup>2</sup> и показал точность соответствия построенной карты 97-99%.

## 2.7. Multi-Robot SLAM with Topological/Metric Maps

Метод [29] можно одновременно отнести к группам методов graph-based SLAM и FastSLAM. Его ключевая идея состоит в применении карт двух типов - топологической карты и метрической.

Метрическая карта фактически является сеточной и получается путем интеграции сканов с дальномера с помощью фильтра частиц Рэо-Блеквела или Калмана.

На топологической карте в виде графа представлено множество фрагментов локальных метрических карт роботов (размером например 20x20 м). Вершины содержат положения роботов и соответствующие им фрагменты карты, а ребра – матрицы преобразования координат между соответствующими вершинами. Каждое перемещение робота на определенное расстояние добавляет новое ребро и вершину. Обмен накопленными картами между роботами происходит при их непосредственных контактах, в момент создания очередной вершины.

Для интеграции всех фрагментов метрических карт и нахождения взаимного положения роботов требуется минимизировать ребра.

Иными словами, в TM-SLAM алгоритме проблема разбивается на несколько под проблем ограниченного размера по построению метрических карт  $M_i$ , которые могут быть решены по отдельности с помощью фильтра частиц. При этом для совмещения отдельных карт воедино требуется определение относительных положений робота  $\phi_i$ , советующих этим картам  $M_i$ , что выполняется путем оптимизации ребер топологического графа.

В вероятностной формулировке данную задачу можно записать как [29]:

$$\begin{aligned} & P(M_i, \bar{X}_i, \phi_i | \Psi, \hat{M}_i, \hat{\mathbf{X}}_i, \hat{\Phi}_i) \\ &= P(M_i, \bar{X}_i | \bar{\Psi}_i, \phi_i) P(\phi_i | \bar{\Psi}_i, \Phi_{\text{adj}(M_i)}, M_{\text{adj}(M_i)}) \\ &= P(M_i, \bar{X}_i | \bar{\Psi}_i, \phi_i) \prod_{\alpha \in \text{adj}(M_i)} P(T_{\{i, \alpha\}} | \phi_\alpha, M_\alpha, \bar{\Psi}_i) \end{aligned}$$

Где  $X_i^-$  - траектории роботов,  $\Psi_i^-$  - измерения, используемые для построения  $M_i$ ,  $T_{\{i,\alpha\}}$  - матрицы трансформации между положениями  $i$  и  $\alpha$ ,  $M_{adj(M_i)}$  – смежные с  $M_i$  карты.

Оптимизация топологической карты выполняется по следующему условию:

$$\begin{aligned}\hat{\mathbf{T}} &= \arg \max_{\mathbf{T}} \prod_{m,n} P(M_m, M_n, \phi_n | \phi_m, T_{ini}) \\ &= \arg \max_{\mathbf{T}} \sum_{m,n} L(M_m, M_n, \phi_n | \phi_m, T_{ini})\end{aligned}$$

Где  $m, n$  – вершины топологической карты,  $T$  – набор всех ребер-матриц трансформации,  $L()$  – запись распределения вероятностей через логарифмы.

Экспериментальная проверка данного метода проводилась группой из 3х роботов в реальном помещении площадью около 2800 м2. Полученные результаты показали хорошую точность построения карты. Однако, оптимизация графа проводилась оффлайн, что не позволяет применять алгоритм для исследования местности в режиме реального времени.



## 2.8. Decentralized Cooperative SLAM for Sparsely-Communicating Robot Networks: A Centralized-Equivalent Approach

Данный алгоритм является попыткой адаптировать централизованные методы SLAM для группы роботов с полными связями с центральным сервером под группу роботов с редкими взаимными связями. [30] По словам авторов метода, для построения карты используются данные с одометра и лазерного дальномера, однако для интеграции вариантов карт разных роботов используются ориентиры.

Для сокращения количества хранимой информации используется правило Маркова: состояние карты робота зависит только от текущего шага и не зависит от предыдущих. Это означает, что для оценки карты роботами может использоваться любой удовлетворяющий данному правилу метод, например, основанный на фильтре Калмана или фильтре частиц (см. раздел «Методы навигации мобильного робота»).

Авторами вводится понятие чекпойнта  $C(kc, ke)$  – точки встречи всех роботов, в которой они обмениваются собранными данными с момента последней встречи. Чекпойнт между роботами  $i$  и  $j$  возможен, если робот  $i$  располагает информацией о состоянии робота  $j$  на каком-либо предыдущем шаге  $ksj < kc$  и всеми измерениями с датчика и одометрией робота  $j$  с шага  $ksj$  до  $kc$ .

В статье приводится ряд теорем, обосновывающих метод. В частности, доказываем, что существование частичных чекпойнов (точки встречи только двух роботов) и применение правила Маркова для оценки карты при частичном чекпойнте не влияет на возможность построения централизованной (полной) карты.

Чтобы избавиться от необходимости внешнего наблюдателя (сервера), который обладает информацией о положении каждого робота и чекпойнтах, используется следующий прием: каждый робот знает количество роботов в группе. Таким образом роботы могут попарно обмениваться картами образуя подгруппы с одинаковыми картами. Для того, чтобы завершить процесс

составления централизованной (полной) карты, требуется наличие частичного чекпойнта между роботами из двух последних подгрупп. При этом ввиду ограниченности связей не гарантируется, что по завершению алгоритма все роботы получают полную карту.

При использовании фильтра вычислительная сложность алгоритма Калмана -  $O(n^4 + n^3m)$ , требуемая память -  $O(n + m)^2$

Проверка метода в реальных условиях проводилась группой из 9 роботов в помещении 8 на 15 м. Эксперименты доказали, что данный метод в условиях редких взаимных связей между роботами (до 0.6% от времени эксперимента) дает оценку карты не хуже, чем централизованный метод с постоянными связями.

Из недостатков можно выделить то, что авторы совсем не затрагивают проблемы ассоциации данных с сенсора и определения взаимных положений роботов.

## 2.9. Multi-robot visual SLAM using a Rao-Blackwellized Particle Filter

Очередной метод, являющийся расширением FastSLAM и использующий в своей основе фильтр частиц для оценки карты. В качестве сенсора используется камера, карта представлена набором ориентиров.

Стоит отметить, что в рамках метода упор сделан именно на решение проблемы ассоциации данных в контексте FastSLAM для группы роботов и здесь не рассматриваются способы обмена данными и взаимодействия между роботами. Вместо этого принимается допущение о том, что все роботы знают взаимные положения в начальный момент времени.

В процессе исследования местности роботы накапливают данные о траектории своего движения  $x_{\langle 1:K \rangle}^t = \{x_{\langle 1 \rangle}^t, x_{\langle 2 \rangle}^t, \dots, x_{\langle K \rangle}^t\}$  под воздействиями  $u_{\langle 1:K \rangle}^t = \{u_{\langle 1 \rangle}^t, u_{\langle 2 \rangle}^t, \dots, u_{\langle K \rangle}^t\}$  и осуществляют наблюдения в каждой точке траектории  $z_{\langle 1:K \rangle}^t = \{z_{\langle 1 \rangle}^t, z_{\langle 2 \rangle}^t, \dots, z_{\langle K \rangle}^t\}$ , которые представляются соответствиями  $c^t = \{c_1, c_2, \dots, c_t\}$  между ориентирами на шагах от 1 до  $t$ . Имеем следующее выражение для решения проблемы SLAM для группы роботов:

$$p(x_{\langle 1:K \rangle}^t, L | z_{\langle 1:K \rangle}^t, u_{\langle 1:K \rangle}^t, c^t) = p(x_{\langle 1:K \rangle}^t | z_{\langle 1:K \rangle}^t, u_{\langle 1:K \rangle}^t, c^t) \prod_{k=1}^N p(l_k | x_{\langle 1:K \rangle}^t, z_{\langle 1:K \rangle}^t, u_{\langle 1:K \rangle}^t, c^t)$$

Таким образом задача SLAM заключается в оценке  $K$  путей роботов и карты  $L$  при условии, что роботы осуществили ряд движений  $u_{\langle 1:K \rangle}^t$  и ряд наблюдений  $z_{\langle 1:K \rangle}^t$  связанных с конкретными ориентирами на карте  $c^t$ . [31, стр. 9]. Приведенное выражение показывает, что вся задача разбивается на задачу оценки положений роботов с помощью фильтра частиц  $p(x_{\langle 1:K \rangle}^t | z_{\langle 1:K \rangle}^t, u_{\langle 1:K \rangle}^t, c^t)$  и  $N$  независимых оценок ориентиров (с помощью фильтров Калмана) при данных положениях каждого из группы роботов. Таким образом, данный метод полностью повторяет метод решения SLAM с помощью фильтра частиц FastSLAM, за исключением того, что в данном случае каждая частица в фильтре содержит положения всей группы роботов  $x_{t, \langle 1:K \rangle} = \{x_{t, \langle 1 \rangle}, x_{t, \langle 2 \rangle}, \dots, x_{t, \langle K \rangle}\}$

В качестве модели шума камеры при фильтрации используются стандартные уравнения для стереопары.

По результатам ряда экспериментов авторами было выбран способ описания ориентиров с применением SIFT дескрипторами в несколько этапов:

1. Для всех ориентиров на карте вычисляется расстояние Махаланобиса  $D$  и выбираются те, у которых  $D$  меньше определенного порога.
2. Для только что обнаруженного SIFT дескриптора  $d_{ti}$  и нескольких уже имеющихся на карте  $d_j$  вычисляется квадрат евклидового расстояния

$$E = (d_{t,(i)} - d_j)(d_{t,(i)} - d_j)^T$$

3. Для ассоциации с  $d_{ti}$  выбирается  $d_j$  который минимизирует это расстояние  $E$ . Если же подходящего кандидата нет, то  $d_{ti}$  добавляется на карту как новый ориентир.

Экспериментальная проверка метода выполнялась в помещении размером  $30 \times 30 \times 2$  метров группой из 3 роботов. В результате определено, что алгоритм дает высокоточные результаты на большом диапазоне настраиваемых параметров (количество роботов, параметры фильтра).

## 2.10. Multi-Robot Marginal-SLAM

По сути является очередным методом решения проблемы SLAM с помощью фильтра частиц с картой, представленной ориентирами. Однако, в данном случае используется маргинальный фильтр частиц. Маргинальный - означает, что в фильтре оценивается только подмножество переменных из всех доступных, в то время как остальным присваивается некоторое значение. В данном случае фильтрация происходит напрямую на маргинальном распределении  $p(x_t|y_{1:t})$  вместо полного распределения  $p(x_{1:t}|y_{1:t})$ , которое используется в FastSLAM. Здесь:  $\theta$  – карта, представленная набором ориентиров,  $x_{1:t}$  – траектория движения робота на интервале от 1 до  $t$ ,  $y_{1:t}$  – наблюдения с сенсоров. Наблюдения условно-независимы от времени с учетом известного предыдущего пути и карты.

В результате маргинализации получаем следующие уравнения фильтра [32]:

$$p_{\theta}(x_t|y_{1:t}) \triangleq \frac{\xi_{\theta}(x_t, y_{1:t})}{\int \xi_{\theta}(x_t, y_{1:t}) dx_t}$$
 - маргинальное апостериорное распределение вероятности положения робота, где  $\xi_{\theta}(x_t, y_{1:t}) = p_{\theta}(y_t|x_t) \int p(x_t|x_{t-1}) p_{\theta}(x_{t-1}|y_{1:t-1}) dx_{t-1}$

$$q_{\theta}(x_t|y_{1:t}) = \sum_{j=1}^N w_{t-1}^{(j)} q_{\theta}(x_t|y_t, x_{t-1}^{(j)}).$$
 - предполагаемое распределение вероятности положения робота

$$w_t = \frac{p_{\theta}(x_t|y_{1:t})}{q_{\theta}(x_t|y_{1:t})}$$
 - веса важности частиц в фильтре, на основании которых производится их выборка

Статья, описывающая метод, также содержит аппроксимации данных уравнений для упрощения их последующей реализации.

В качестве модели движения используются стандартные уравнения для FastSLAM с гауссовым шумом.

Для объединения карт роботов применяется алгоритм, основанный на соотнесении особенностей с помощью SIFT дескрипторов [36]. Суть его такова: если карты разных роботов содержат одинаковые особенности, то вычисляются параметры преобразования между советующими системами координат, затем обе карты преобразуются в глобальную систему координат и интегрируются. Это положительно отличает алгоритм от описанных ранее, поскольку позволяет избавиться от допущения, что роботам для обмена информацией требуются непосредственные встречи друг с другом.

Кроме того, метод может быть расширен для устойчивой работы в пространствах с динамически движущимися объектами и для возможности восстановления после ошибок ассоциации данных на начальных этапах работы.

Проверка алгоритма проводилась в специально созданном окружении квадратной формы (40 на 40 м) группой из 2х роботов. Выявлено, что с увеличением времени работы алгоритма точность построения карты повышается практически до полного соответствия оригиналу. В то же время при наличии большого количества шумов и ошибок ассоциации данных требуются большое количество шагов обновления карты результата (то есть многократное прохождение роботов по одному маршруту) для получения приемлемого результата.

## 2.11. Multi-Robot Range-Only SLAM by Active Sensor Nodes for Urban Search and Rescue

Метод, нацеленный на применение при проведении поисково-спасательных операций группами роботов в городской среде. Основная особенность и требование алгоритма – размещение на исследуемой местности специальных сенсорных узлов, содержащих: передатчик, собственную память, датчик ориентации в пространстве и прикладные датчики для мониторинга параметров среды (температура, давление). Данные узлы служат ориентирами на местности, а также позволяют передавать данные о состоянии карты между роботами. Сами роботы (или как называют их авторы - роботизированные платформы) оснащены лазерным дальномером, одометром и мобильным сенсорным узлом [33].

Вычисление взаимного положения сенсорных узлов и роботов осуществляется путем расчёта мощности сигнала от передатчиков, установленных на каждом аппарате.

В основе решения задачи картографирования и локализации FastSLAM со стандартными вероятностными моделью, фильтром частиц и картой, состоящей из ориентиров (которыми являются сенсорные узлы). В то же время имеется возможность построения сеточной карты путем наложения на карту данных с дальномера.

Оценка расстояния между сенсорными узлами в закрытых помещениях со стенами и прочими препятствиями выполняется по модели распространения радиосигнала:

$$P(d)[dB] = p(d_0)[dB] - 10n \log \frac{d}{d_0} + X_{\sigma}[dB],$$

Где  $P$  – мощность сигнала,  $d$  – расстояние до передатчика,  $p(d_0)$  – известные значения мощности при прямой видимости,  $n$  – среднее значение затухания в зависимости от типа помещения,  $X_{\sigma}$  – стандартная девиация сигнала.

После получения наблюдения (сигнала от передатчика от узла)  $z = r$  происходит его связывание с конкретным ориентиром, поскольку каждый сенсорный узел имеет уникальный ID. По известным данным одометрии  $(rx, ry)$  и предыдущим координатам  $(bx, by)$  вычисляется новое ожидаемое расстояние до узла:

$$h_i(s) = \left( \sqrt{(r_x - b_x)^2 + (r_y - b_y)^2} \right)$$

Далее вычисляется ошибка (инновация)  $v$  и ее ковариация  $\sigma_v$ , происходит обновление фильтра Калмана, отвечающего за оценку положения сенсорного узла:

$$\begin{aligned} v_i &= z - h_i(s) \\ \sigma_{v_i} &= \nabla h_s \Sigma_s \nabla h_s^T + \sigma_z \end{aligned}$$

$$\begin{aligned} K &= \Sigma_{s_i} \nabla h_s^T \sigma_{v_i}^{-1} \\ s_{i+1} &= s_i + K v_i \\ \Sigma_{s_{i+1}} &= \Sigma_{s_i} - K \sigma_{v_i} K^T. \end{aligned}$$

На последующих этапах выполняются стандартные процедуры фильтрации частиц в соответствии с FastSLAM. Веса частиц  $w_i$  вычисляются как расстояние Махаланобиса:

$$w_{i+1} = w_i - v_i^T \sigma_{v_i}^{-1} v_i.$$

Обмен данными о карте происходит путём вычисления разниц координат между локальной картой робота и той, которая хранится в памяти вновь обнаруженного узла, и далее интеграции двух карт посредством фильтра частиц.

Ввиду вышесказанного можно сделать вывод, что алгоритм неприменим для построения карты и проведения спасательных работ на полностью неизвестной местности. В то же время затраты на разработку и предварительную установку сенсорных узлов компенсируются, во-первых, отсутствием проблемы ассоциации данных и замыкания циклов. Во-вторых, существенным



сокращением количества особенностей и, соответственно, экономией вычислительных ресурсов. В-третьих, более частым взаимодействием между роботами.

Эксперименты с алгоритмом проводились как на симуляции размером более 1000 м<sup>2</sup>, так и в реальных условиях. В реальном эксперименте участвовали 4 робота и 18 датчиков, размещенных в помещении размером примерно 500 м.

## 2.12. Multi-Robot Pose Graph Localization and Data Association from Unknown Initial Relative Poses via Expectation Maximization

Алгоритм является модификацией Graph-based SLAM, в котором проблема локализации и построения карты разбивается на задачу оптимизации графа передвижения робота и установления корректных соответствий между новыми и уже имеющимися данными. Кроме того, применительно к группам роботов встает дополнительная задача – первичное определение их относительных положений. В качестве сенсора в роботе используется лазерный дальномер, следовательно, локальные участки карты, содержащиеся в узлах графа, представлены в виде сетки.

Авторы используют для решения описанной проблемы метод максимизации ожиданий. Так, для оптимизации графа передвижения одного робота требуется выполнить максимизацию [34]:

$$\hat{X}^r = \arg \max_{X^r} p(X^r | Z^r)$$

Где  $X^r$  – положение робота  $r$ ,  $Z^r$  – наблюдения робота  $r$

Преобразование локальной карты робота  $r1$  в момент времени  $k$  в локальную карту робота  $r2$  в момент времени  $l$  обозначим как  $u_{k,l}^{r1,r2}$ . Данное преобразование может быть найдено как при прямом контакте роботов, так и при наблюдении двумя роботами одной и той же местности.

Тогда можно записать функцию плотности вероятности совместного распределения для группы роботов:

$$p(X|Z) \propto \prod_r p(X^r | Z^r) \prod_{(r1,r2,k,l) \in \mathcal{F}} p(u_{k,l}^{r1,r2} | x_k^{r1}, x_l^{r2})$$

Для решения проблемы неизвестных ассоциации данных разных роботов генерируется набор соответствий  $u_{k,l}^{r1,r2}$  (путем совмещения сканов местности разных роботов алгоритмом ICP). Каждому из них присваивается скрытая переменная  $j_{k,l}^{r1,r2}$  показывающая, является ли данное соответствие инлайером или аутлайером и нужно ли его учитывать при оценке карты. Однако, для

нахождения данной переменной требуются знания о преобразовании  $T_{r2}^{r1}$  из локальных координат робота  $r1$  в координаты робота  $r2$ .

Предложенный авторами метод нахождения  $T_{r2}^{r1}$  использует тот факт, что каждый кандидат на соответствие  $u_{k,l}^{r1,r2}$  предполагает свое решение задачи нахождения  $T_{r2}^{r1}$ , однако только у инлайеров эти решения совпадают. Традиционным решением данной задачи является кластеризация наблюдений сдвигом среднего значения (meanshift), однако авторы используют метод максимизации ожиданий (Expectation-Maximization), который можно сформулировать как:

$$\hat{T}_{r2}^{r1} = \arg \max_{T_{r2}^{r1}} \sum_{(r1, r2, k, l)} p(j_{k,l}^{r1,r2} | \hat{T}_{r2}^{r1}, \hat{X}^{SR}, Z) \log p(j_{k,l}^{r1,r2}) p(u_{k,l}^{r1,r2} | \hat{x}_k^{r1}, \hat{x}_l^{r2}, j_{k,l}^{r1,r2}, T_{r2}^{r1})$$

Где  $X^{SR}$  – траектория движения одного робота, а суммирование относится ко всем возможным ассоциациям данных между двумя роботами.

Начальное значение для  $T_{r2}^{r1}$  определяется путем предварительной кластеризации ассоциаций данных и выбора значения для которого имеется наибольшее количество инлайеров.

После выполнения описанных процедур графы роботов можно преобразовать в общую систему координат, интегрировать и выполнить стандартную процедуру оптимизации.

Экспериментальная проверка данного метода на группе из 3 роботов в помещении размером 15 на 15 м (225 м<sup>2</sup>) показала его приемлемую точность. Однако авторами при тестировании использовались некоторые допущения - замыкание циклов передвижения вручную, оптимизация графа в офлайн режиме.

## 2.13. A Neural Network-based Multiple Robot Simultaneous Localization and Mapping

Первый метод с принципиально иным подходом к проблеме SLAM. Использует нейронные сети и машинное обучение. В тоже время для обработки сырых данных с сенсоров каждый из роботов использует расширенный фильтр Калмана. В качестве источников данных для роботов служат высокоточный одометр и лазерный дальномер.

Авторами предложены:

1. Высокоуровневый алгоритм сегментации карты местности, применимый для обработки сеточных карт.
2. Применение самоорганизующихся карт Кохонена для кластеризации предварительно обработанных сеточных карт.
3. Оценка матриц трансформации для карт двух различных роботов используя кластерные точки.
4. Использование нормалей поверхностей для сопоставления кластерных точек двух различных карт.

Предложенный нейросетевой метод позволяет выявить кластеры на карте, которые являются по сути особенностями (ориентирами) на местности. Основная цель метода – найти относительное преобразование между двумя различными картами с учетом присутствующей неопределённостей. Неопределенность выражается в том, что ячейки карты представляются числами с плавающей точкой в интервале  $[-1; 1]$ .

Алгоритм сегментации карты таков [35]. Пусть  $M$  – совокупность всех препятствий (не пустых ячеек) на сеточной карте,

$S$  – множество начальных точек, которое предварительно содержит произвольную точку  $p_0 = (x_0, y_0)$ ,

$C^n$  – множество всех точек в сегменте  $n$ .

Для каждой начальной точки  $p_s$  можно найти множество соседних к ней точек  $C_s$ :

$$C_s = \{p = (x, y) | (x - x_s)^2 + (y - y_s)^2 < r^2\}.$$

Затем вычисляется множество новых потенциальных стартовых точек:

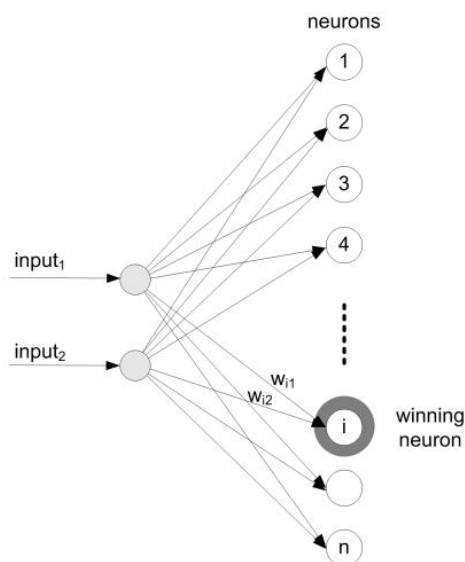
$$S_s = \{p = (x, y) | r^2 < (x - x_s)^2 + (y - y_s)^2 < (r + 1)^2\}$$

В  $S$  в итоге включаются только те точки из  $S_s$ , которые еще не входят в сегмент  $C^n$ . Затем сегмент  $C^n$  обновляется так, чтобы он включал в себя точки из  $S_s$  и  $C_s$ .

Данный алгоритм сегментации повторяется, пока  $S$  содержит точки.

Следующий и основной этап алгоритма – кластеризация полученных сегментов с помощью карт Кохонена. Кластер может содержать либо сырые данные – сегмент сеточной карты, либо его форму, описанную кривыми.

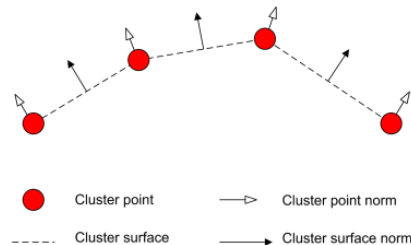
Вид нейросети – карты Кохерена - представлен на рисунке:



На этапе обучения входными данными являются координаты всех занятых препятствиями ячеек карты. Нейросеть находит нейрон, чей вес наилучшим образом соответствует входным данным. Затем веса этого и соседних с ним нейронов обновляются. В результате обучения получаем веса нейронов, которые и являются искомыми кластерами.

На финальном этапе алгоритма по известным кластерам и их координатам вычисляются параметры преобразования между двумя картами:

- 1) Относительный угол находится путем построения для каждой карты круговой гистограммы по направлениям нормалей к кластерной поверхности, и, затем, их сравнения.
- 2) Относительное положение находится с помощью нормалей к кластерным точкам итеративным методом, близким к ICP:



Из преимуществ представленного алгоритма можно выделить отсутствие необходимости непосредственных контактов между роботами для совмещения их карт.

Текущий вариант метода дает результаты достаточной точности – 95% соответствия построенной карты с местностью. Это было доказано в ходе реальных экспериментов с двумя роботами на местности размером 10 на 17 метров (170 м<sup>2</sup>).

## 2.14. Сравнение алгоритмов SLAM для группы роботов

Ниже представлена сводная таблица сравнения всех описанных методов по ключевым параметрам.

В столбце «точность» показана оценка алгоритмов по шкале «плохо-удовлетворительно-хорошо-отлично». Оценка производилась по следующим параметрам:

5. Наличие данных о реальных экспериментах
6. Размер пространства, где проводился эксперимент (больше – лучше)
7. Кол-во роботов, принимавших участие в эксперименте
8. Качество проработки метода в статье (какие аспекты алгоритма описаны, какие допущения использовались)
9. Лежащий в основе алгоритм решения задачи SLAM для одного робота

Алгоритм/статья	Основана на	Тип карты	Сложность слияния карт (для одного робота)	Затраты памяти	Размер карты (в эксперименте) м <sup>2</sup> или м*м	Точность (оценочно)	Кол-во роботов (в эксперименте)	Реальный эксперимент
[22]	EKF	Features	$O(n*m^3)$	$O(n*P)$		Удовлетворительно	2	Нет
[23]	EKF	Features	$O(m \log m)$		4800 / 60*80	Хорошо	2	Да
[27]	SEIF	Features	$O(m \log m)$	$O(n+m)$	350*350/120000	Хорошо	8	Нет
[24]	Particle filter	Features			<5*5/25	Удовлетворительно	2	Да
[25]	Particle filter	Grid			2500	Хорошо	4	Нет
[26]	Particle filter	Grid			200	Хорошо	2	Да
[28]	Scan matching+ graph optimization	Graph+Grid			>13*38/500	Отлично (97-99%)	2-8	Да
[29]	Particle filter + graph optimization	Graph+Grid	$O(1)$		77*36/2800	Хорошо+	3	Да

[30]	EKF	Features	$O(k - k_c)(n^4 + n^3m)$	$O(k - k_c)(n + m)^2$	15*8/120	Хорошо	9	Да
[31]	Particle filter	Features			30*30/900	Хорошо	3	Да
[32]	Particle filter	Features			40*40/320	Хорошо	2	Да
[33]	Particle filter	Features+Grid			17*29/500	Хорошо	4	Да
[34]	Scan matching+ graph optimization	Graph+Grid			15*15/225	Хорошо	3	Да
[35]	EKF + Neural network	Grid			10*17/170	Хорошо+	2	Да

Здесь:

P – периметр карты

n – кол-во роботов, m – кол-во ориентиров

(k – k<sub>c</sub>) – кол-во шагов между чекпойнтами [30]



## 2.15. Проблемы методов SLAM для группы роботов

Метод	Преимущества	Недостатки, нерешенные проблемы
[22]	1	a,b,d,e,f,m
[23]	1, 5, 6, 7	a, c, e
[27]	1, 2, 3, 5, 7	k
[24]	1, 5, 7	c, g, e, m
[25, 26]	1, 3, 5, 7	j, c, l, n
[28]	1, 5, 8, 3, 6, 7	c
[29]	5, 7, 6	h, l, c
[30]	1, 5, 7	a, g, c, l
[31]	1, 4, 3, 5, 6	c, f, l
[32]	1, 5, 7, 2	k, m, f
[33]	1, 2, 3	f, g, i, l
[34]	2, 3, 5, 7	n, h, f
[35]	1, 2, 3, 5, 7	k

### Недостатки:

- Размер карты ограничен ввиду применения EKF и карты, состоящей из особенностей
- Централизованный сервер ограничивает автономность роботов
- Для взаимного определения положения роботам необходимы прямые контакты; кроме того, требуется установка специальных маркеров на роботов
- Алгоритм не реализован до конца
- Проблема ассоциации данных/повторного распознавания ориентиров
- Авторами приняты допущения о начальных условиях (как правило о том, что положения роботов либо соответствия данных известны априори)
- Требуется установка специальных ориентиров/устройств на местности
- Невозможность выполнения метода онлайн, ввиду больших требований к ресурсам
- Необходимы сложные или неэффективные схемы передачи данных между роботами
- Требуется много памяти ввиду использования фильтра частиц и сеточной карты
- Недостаточная устойчивость метода при различных сценариях – в условиях ограниченных контактов, в условиях больших шумов данных
- Описание алгоритма недостаточно детализировано для его реализации, некоторые важные аспекты не рассмотрены, например, проблема ассоциации данных или обмена данными между роботами вынесены авторами за пределы статьи
- Сомнительные экспериментальные данные
- Проблема замыкания циклов

### Преимущества:

- Интеграция карт нескольких роботов онлайн
- Роботам не требуются прямые контакты для определения относительного положения
- Проблема ассоциации данных успешно решена
- Возможно комбинация метода с другими для получения наилучших результатов, либо модификация
- Роботы полностью автономны / децентрализованная топология
- Получение приемлемых результатов подтверждается множеством реальных экспериментов
- Не требуются допущения о том, что начальные положения роботов известны
- Эффективный алгоритм обмена данными между роботами

## 2.16. Выводы

В результате проведения исследования большинства существующих методов группового SLAM можно сделать следующие выводы:

1. Несмотря на большое разнообразие все рассмотренные методы имеют те или иные недостатки, и, кроме того, сопровождаются весьма скромными данными по их тестированию в реальных условиях и на больших группах роботов.
2. Аналогично методам SLAM для одного робота, алгоритмы на фильтре частиц имеют преимущества как по точности, так и по скорости работы перед алгоритмами на Расширенном фильтре Калмана.
3. В то же время классические фильтрующие методы просты в реализации, поскольку разрабатываются и применяются относительно давно.
4. Хорошую точность построения карты дают алгоритмы, основанные на оптимизации графа передвижения робота, однако из выполнения не всегда возможно в режиме on-line из-за высоких требований к вычислительным мощностям.
5. Перспективными являются методы, основанные на машинном обучении и нейросетях, несмотря на их слабую проработку в настоящее время. Ид преимущество и недостаток – необходимость обучения.
6. Методы, не использующие ориентиры для построения карты, предпочтительнее, поскольку алгоритмы машинного зрения несовершенны и не всегда удастся решить проблемы ассоциации данных и замыкания циклов.
7. Наибольший интерес представляют алгоритмы, не зависящие от лежащего в основе SLAM для одного робота, поскольку: 1) основа не влияет на качество работы групповой части метода, 1) данные методы могут быть легко модифицированы под конкретное ПО и оборудование

8. Очевидно, методы, в которых не является обязательным условие прямых контактов роботов, предпочтительнее, поскольку в таком случае заметно расширяются границы условий устойчивой работы алгоритма.

### 3. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Simultaneous localization and mapping with the extended Kalman filter // URL: [http://www.iri.upc.edu/people/jsola/JoanSola/objectes/curs\\_SLAM/SLAM2D/SLAM%20course.pdf](http://www.iri.upc.edu/people/jsola/JoanSola/objectes/curs_SLAM/SLAM2D/SLAM%20course.pdf)
2. FastSLAM: An Efficient Solution to the Simultaneous Localization And Mapping Problem with Unknown Data Association // URL: <http://ai.stanford.edu/~koller/Papers/Thrun+al:04b.pdf>
3. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem // URL: <http://robots.stanford.edu/papers/montemerlo.fastslam-tr.pdf>
4. FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges // URL: <http://robots.stanford.edu/papers/Montemerlo03a.pdf>
5. Fast ICP-SLAM for a Bi-Steerable Mobile Robot // URL: [https://www.researchgate.net/profile/O\\_Azouaoui/publication/281815827\\_Fast\\_ICP-SLAM\\_for\\_a\\_bi-steerable\\_mobile\\_robot\\_in\\_large\\_environments/links/5731c33c08ae6cca19a30772/Fast-ICP-SLAM-for-a-bi-steerable-mobile-robot-in-large-environments.pdf](https://www.researchgate.net/profile/O_Azouaoui/publication/281815827_Fast_ICP-SLAM_for_a_bi-steerable_mobile_robot_in_large_environments/links/5731c33c08ae6cca19a30772/Fast-ICP-SLAM-for-a-bi-steerable-mobile-robot-in-large-environments.pdf)
6. MRSLAM Multi-Robot Simultaneous Localization and Mapping // URL: [http://ap.isr.uc.pt/archive/MRSLAM\\_dissertacao\\_Joao\\_Martins-vfinal-040913\\_235208.pdf](http://ap.isr.uc.pt/archive/MRSLAM_dissertacao_Joao_Martins-vfinal-040913_235208.pdf)
7. A Tutorial Approach to Simultaneous Localization and Mapping // URL: [https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam\\_blas\\_repo.pdf](https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam_blas_repo.pdf)
8. Алгоритмы локальной навигации и картографии для управления роботом // URL: <https://m.cyberleninka.ru/article/v/algoritmy-lokalnoy-navigatsii-i-kartografii-dlya-bortovoy-sistemy-upravleniya-avtonomnogo-mobilnogo-robota>

9. DP-SLAM: Fast, Robust Simultaneous Localization and Mapping Without // URL: <http://people.ee.duke.edu/~lcarin/Lihan4.21.06a.pdf>
10. DP-SLAM 2.0 // URL: <https://users.cs.duke.edu/~parr/dpslam2.pdf>
11. Grid-Based FastSLAM // URL: <http://robots.stanford.edu/papers/Haehnel03b.pdf>
12. Probabilistic robotics - Sebastian Thrun, Dieter Fox, Wolfram Burgard // URL: <https://docs.ufpr.br/~danielsantos/ProbabilisticRobotics.pdf>
13. A Tutorial on Graph-Based SLAM // URL: <http://www2.informatik.uni-freiburg.de/~stachnis/pdf/grisetti10titsmag.pdf>
14. Algorithms for Simultaneous Localization and Mapping // URL: [http://cseweb.ucsd.edu/~yuc007/documents/re\\_report.pdf](http://cseweb.ucsd.edu/~yuc007/documents/re_report.pdf)
15. The Normal Distributions Transform A New Approach to Laser Scan Matching // URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.10.7059&rep=rep1&type=pdf>
16. Система навигации сервисного робота в среде с динамическими препятствиями - Герасимов В. Н. // URL: [http://www.bmstu.ru/assets/dissertations/%D0%93%D0%B5%D1%80%D0%B0%D1%81%D0%B8%D0%BC%D0%BE%D0%B2\\_-\\_%D0%B4%D0%B8%D1%81%D1%81%D0%B5%D1%80%D1%82%D0%B0%D1%86%D0%B8%D1%8F%D0%BE%D1%82%D0%B7%D1%8B%D0%B2.pdf](http://www.bmstu.ru/assets/dissertations/%D0%93%D0%B5%D1%80%D0%B0%D1%81%D0%B8%D0%BC%D0%BE%D0%B2_-_%D0%B4%D0%B8%D1%81%D1%81%D0%B5%D1%80%D1%82%D0%B0%D1%86%D0%B8%D1%8F%D0%BE%D1%82%D0%B7%D1%8B%D0%B2.pdf)
17. A Flexible and Scalable SLAM System with Full 3D Motion Estimation // URL: [http://www.sim.informatik.tu-darmstadt.de/publ/download/2011\\_SSRR\\_KohlbrecherMeyerStrykKlingauf\\_Flexible\\_SLAM\\_System.pdf](http://www.sim.informatik.tu-darmstadt.de/publ/download/2011_SSRR_KohlbrecherMeyerStrykKlingauf_Flexible_SLAM_System.pdf)
18. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters // URL: <http://www2.informatik.uni-freiburg.de/~stachnis/pdf/grisetti07tro.pdf>

19. 3D Mapping with an RGB-D Camera // URL: <http://www2.informatik.uni-freiburg.de/~endres/files/publications/endres13tro.pdf>
20. Метод максимального правдоподобия, информационный метод Фишера // URL: [https://en.wikipedia.org/wiki/Fisher\\_information](https://en.wikipedia.org/wiki/Fisher_information)
21. Modern Multi Agent SLAM approaches survey - Krinkin Kirill, Filatov Anton, Filatov Artyom // URL: <https://fruct.org/publications/abstract20/files/Kri.pdf>
22. Extending SLAM to Multiple Robots // URL: [http://web.mit.edu/16.412j/www/html/Final%20Projects/FIN\\_Novosad\\_Howe.pdf](http://web.mit.edu/16.412j/www/html/Final%20Projects/FIN_Novosad_Howe.pdf)
23. Multi-robot SLAM with Unknown Initial Correspondence // URL: <http://www-users.cs.umn.edu/~stergios/papers/IROS06-MapAlign.pdf>
24. N. Ergin Ozkucur and H. Levent Akin, "Cooperative Multi-Robot Map Merging Using Fast-SLAM" // URL: [https://www.researchgate.net/publication/220797575\\_Cooperative\\_Multi-robot\\_Map\\_Merging\\_Using\\_Fast-SLAM](https://www.researchgate.net/publication/220797575_Cooperative_Multi-robot_Map_Merging_Using_Fast-SLAM)
25. Multi-robot Simultaneous Localization and Mapping using Particle Filters // URL: <http://people.bordeaux.inria.fr/pierre.delmoral/p27.pdf>
26. Rao-Blackwellized Particle Filters Multi Robot SLAM with Unknown Initial Correspondences and Limited Communication // URL: <https://pdfs.semanticscholar.org/0699/c2e386b73369091373aa1bd837f26bdcbe79.pdf>
27. Multi-Robot SLAM With Sparse Extended Information Filers // URL: <http://robots.stanford.edu/papers/Thrun03e.pdf>
28. Multi-Robot SLAM using Condensed Measurements // URL: <http://webdiis.unizar.es/~mtlazaro/papers/Lazaro-IROS13.pdf>
29. Multi-Robot SLAM with Topological/Metric Maps // URL: <https://pdfs.semanticscholar.org/3a68/7395f5008e551039fad80451c7380e0edab3.pdf>
30. Decentralized Cooperative SLAM for Sparsely-Communicating Robot Networks: A Centralized-Equivalent Approach // URL:

- <http://pdfs.semanticscholar.org/685f/299d4f635fb237af7f764be77c1dd7e207ee.pdf>
31. Multi-robot visual SLAM using a Rao-Blackwellized Particle Filter // URL: [http://arvc.umh.es/documentos/articulos/2009\\_RAS\\_multirobot.pdf](http://arvc.umh.es/documentos/articulos/2009_RAS_multirobot.pdf)
  32. Multi-Robot Marginal-SLAM // URL: <http://www.cs.ubc.ca/~nando/papers/marginalslamijcai.pdf>
  33. Multi-Robot Range-Only SLAM by Active Sensor Nodes for Urban Search and Rescue // URL: [http://gki.informatik.uni-freiburg.de/papers/sun\\_et\\_al\\_rocup08.pdf](http://gki.informatik.uni-freiburg.de/papers/sun_et_al_rocup08.pdf)
  34. Multi-Robot Pose Graph Localization and Data Association from Unknown Initial Relative Poses via Expectation Maximization // URL: <https://pdfs.semanticscholar.org/3827/1b1d70a71ca577b482c6f25a518ef8d68863.pdf>
  35. A Neural Network-based Multiple Robot Simultaneous Localization and Mapping // URL: [http://www.sajad-saeedi.ca/uploads/3/8/5/9/38597021/saeedi\\_iros\\_2011\\_b.pdf](http://www.sajad-saeedi.ca/uploads/3/8/5/9/38597021/saeedi_iros_2011_b.pdf)
  36. Unsupervised 3D Object Recognition and Reconstruction in Unordered Datasets // URL: <http://matthewalunbrown.com/papers/3dim05.pdf>
  37. Multiple-robot Simultaneous Localization and Mapping - A Review // URL: [http://www.sajad-saeedi.ca/uploads/3/8/5/9/38597021/saeedi\\_jfr\\_2015.pdf](http://www.sajad-saeedi.ca/uploads/3/8/5/9/38597021/saeedi_jfr_2015.pdf)
  38. Scale-invariant feature transform // URL: [https://en.wikipedia.org/wiki/Scale-invariant\\_feature\\_transform](https://en.wikipedia.org/wiki/Scale-invariant_feature_transform)