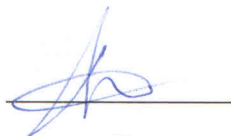


МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по научно-исследовательской деятельности
аспиранта за пятый семестр

Аспирант

направление 09.06.01
специальность 05.13.11



Филатов Антон Юрьевич

Руководитель д.т.н., проф



Экало Александр Владимирович

Санкт-Петербург

2018

Введение

В настоящее время всё больше задач передаётся мобильным движущимся роботам. Это могут быть марсоходы, автопилотируемые автомобили, изучающие местность дроны, даже роботы-пылесосы. Перед всеми перечисленными агентами может встать задача одновременного определения собственного положения и построения карты неизвестного окружения (Simultaneous Localization And Mapping — SLAM).

Актуальность

Очевидно, что эффективность работы, в том числе точность построенной в результате карты, можно увеличить, если привлечь к выполнению работы нескольких агентов. Однако, в этом случае возникают новые вопросы и задачи, в том числе вопрос о количестве информации, которую они должны передавать друг другу, способ связи между ними и прочее. С другой стороны агенты, имея собственное представление об окружающей среде, могут являться арбитрами друг для друга.

Разработка алгоритма, решающего задачу многоагентного SLAM для малопроизводительных компьютеров (которыми могут оснащаться дешёвые мобильные роботы) является актуальной краеугольной задачей, где необходимо соблюсти баланс между точностью и быстродействием.

Целью данной работы является разработка алгоритма многоагентного SLAM, подходящего для маломощных процессоров, который позволит повысить точность результирующей карты и траектории по сравнению с одноагентным алгоритмом.

Задачи данной работы:

- исследование существующих решений задачи одноагентного SLAM и определение масштабируемости сущностей, формируемых в процессе решения этой задачи
- исследование существующих решений задачи многоагентного SLAM и их анализ производительности и точности
- формирование требований к разрабатываемому алгоритму
- разработка архитектуры компонентов алгоритма решения задачи многоагентного SLAM и программная реализация
- тестирование алгоритма, сравнение точности и производительности с существующими алгоритмами, анализ качества работы разработанного алгоритма

Объектом исследования является задача одновременного определения положения и построения карты стай роботов.

Предметом исследования является быстродействие и точность алгоритма, решающего задачу одновременного определения положения и построения карты стай роботов.

Научная новизна работы: разработанный подход описывает решение задачи многоагентного SLAM не только с точки зрения точности, но и с точки зрения производительности. Приведены правила, по которым в зависимости от вычислительных возможностей можно прогнозировать точность работы, сохраняя высокую скорость.

Практическая значимость работы: разработанный программный продукт можно применять для роботов, управляемых малопроизводительными процессорами, что позволяет удешевить аппаратную составляющую системы. Разработанный алгоритм не накладывает ограничений на области применения.

Апробация результатов.

Разработанный и реализованный алгоритм применяется для исследовательских целей изучения различных решений задачи SLAM. Алгоритм был протестирован в рамках лаборатории мобильных роботов JetBrains Research.

Глава 1. Решение задачи SLAM одним агентом

1.1 Постановка и формальное описание задачи SLAM

Объём 2 страницы

Прежде, чем говорить о многоагентных системах необходимо проклассифицировать существующие одноагентные подходы и определить, возможно ли масштабирование этих подходов и сколько это будет стоить.

Задача SLAM построить карту неизвестной местности и определить на ней собственное положение при условии, что агент может наблюдать окружающую среду используя только такие датчики как видеокамера, лазерный дальномер, сонар и тому подобное.

1.2 Классификация алгоритмов решения задачи SLAM

Объём 2 страницы

По размерности наблюдений:

- 3D

- 2D

По типу используемых сенсоров

- визуальный

- лазерный

По способу обработки входных измерений

- основанный на выделении особых точек

- использующий «сырые» измерения

По способу представления карты

- использующий сетку занятости

- графовый

По характеру изменения окружающей среды

- статический

- динамический

1.3 SLAM, основанные на выделении особых точек.

1.3.1 Общий подход

Объём: 2 страницы

Исторически первыми появились именно такие алгоритмы. Идея очень близка к человеческому восприятию мира. Ориентируясь в незнакомом городе, человек отмечает для себя, например, высокие башни и определяет приблизительное расстояние между ними, а дальше использует их, как ориентир.

Для автономных вычислительных агентов можно реализовать подобный алгоритм: необходимо выделять на снятых измерениях «особые» точки, из которых и будет в будущем состоять карта. Чем больше таких точек будет определено в процессе работы, тем точнее будет результирующая карта и позиция агента. Также на точность влияет количество позиций, с которых удалось различить «особую» точку. Поэтому для таких алгоритмов вопрос о робастном выделении особых точек стоит очень остро.

Работа алгоритма разделена на несколько этапов, в числе которых можно выделить снятие измерений (скан, кадр и пр.), скан матчинг (сопоставление измерения и карты) и обновление карты.

1.3.2 EKF SLAM

Объём 3 страницы

Первая идея решения заадчи SLAM, предложенная [R.Smith,M.Self,andP. Cheeseman] в 1990 году, опиралась на расширенный фильтр калмана. Идея в том, что все препятствия должны коррелировать между собой, то есть (если агент работает в неподвижном окружении) взаимное расположение препятствий не должно меняться. А это значит, что, пронаблюдав пару препятствий под разными углами, можно повысить точность оценки их расположения.

Однако этот алгоритм непригоден для масштабирования. Мало того, что собственная матрица ковариаций препятствий увеличивается в процессе работы алгоритма при нахождении новых препятствий, так ещё и сопоставлять две или несколько матриц становится ёмкой вычислительной задачей.

1.3.3 FastSLAM

Объём: 1 страница

Предложенная [Thrun и компания] идея продолжает концепцию EKF SLAM, но увеличивает производительность за счёт отсутствия корелляции между препятствиями. Тесты в исходной статье показывают, что точность такого подхода остаётся достаточно высокой.

1.4 Лазерные и использующие сетку занятости SLAM.

1.4.1 Байесовская теория

Объём 3 страницы

Альтернативным подходом для хранения карты в памяти является использование сетки занятости. То есть всё пространство разделяется на двумерные или трёхмерные клеточки, каждая из которых содержит число от 0 до 1 — вероятность быть занятой. Лучше всего такой подход проявляет себя для лазерных SLAM-ов, то есть для тех, которые запускаются на агентах, использующих лазерные дальномеры, сонары и прочее.

Такое представление карты позволяет использовать Байесовскую теорию для определения собственного положения, поскольку для каждой ячейки можно вычислить вероятность нахождения в ней агента при условии, что в данный момент времени получены определённые наблюдения.

1.4.2. TinySLAM

Объём 3 страницы

Наивным решением, работающем для лазерного SLAM можно считать [slam in 200 lines of code]. Здесь предлагается использовать простейшие ячейки, тривиальным образом вычислять распределение вероятностей по всем клеточкам и прочее. Данный подход показывает наименьшую вычислительную сложность, но и является далеко не самым точным.

В данном разделе также необходимо продемонстрировать полную схему работы SLAM от скан матчера до встраивания скана в карту.

1.4.3. Теория Демстера-Шафера

Объём 4 страницы

Альтернативой для байесовской модели вычисления вероятности является теория Демстера-Шафера и основанная на ней модель вычисления вероятности TBM — Transferable Beleife

Model. Исторически она применялась в динамических SLAM, однако она также хорошо зарекомендовала себя в условиях, когда измерения содержат заметный шум.

Идея подхода в том, что вместо распределения вероятности от состояния 0 — ячейка свободна до 1 — ячейка занята можно использовать 4-х мерный вектор, где размерности представляют собой вероятности ячейки быть свободной, быть занятой, быть неизвестной и находиться в состоянии конфликта. Такой объект требует новых правил для сложения и умножения вероятностей.

SLAM на основе ТВМ, например [credebilist] может применяться не только в динамическом окружении, но также и в статическом применение этой модели к TinySLAM увеличивает точность его работы.

1.5 Фильтр частиц. Gmapping

Объём 2 страницы

Основным недостатком подхода, описанного в предыдущем пункте является невозможность проверки консистентности выходных данных «на лету». Если в процессе работы алгоритма что-то пошло не так, сам алгоритм не сможет распознать эту ситуацию и впоследствии получит неверные данные.

Для решения этой проблемы применяется фильтр частиц. Назовём частицей состояние системы в определённый момент (позиция агента и построенная им карта). Если во время работы алгоритма происходит ситуация, когда измерения не могут быть сопоставлены с картой единственным образом, необходимо создать новую частицу и отслеживать сразу обе, вычисляя правдоподобность каждой из них. Примером алгоритма, использующего такой подход является gmapping, который в данный момент считается де-факто стандартным алгоритмом решения задачи SLAM. Однако для масштабирования такой алгоритм подходит плохо, поскольку не решёнными являются вопросы сопоставления набора частиц различных агентов. [некоторые] предлагают усреднение частиц перед процессом обмена информацией.

1.6 Графовый SLAM. Cartographer

Объём 3-4 страницы

Наиболее популярным в данный момент является графовый подход к построению карты. При таком подходе сырые измерения объединяются в узлы графа, а рёбрами служит трансформация между позициями, откуда можно эти измерения снять. Сложность такого подхода состоит в т. н. Замыкании циклов: когда происходит снятие измерения с позиции, которую агент уже посещал. В этом случае необходимо каким-то образом найти вершину графа, содержащую такое же наблюдение и «замкнуть» цикл в графе. В данный момент именно этот вопрос делает графовый подход вычислительно сложным.

Одним из графовых алгоритмов является google Cartographer, который комбинирует онлайн и офлайн оптимизацию с целью нахождения баланса между точностью и скоростью работы.

Идея алгоритма очень подходит для масштабирования, поскольку инструмент «поиск» некоторого наблюдения в графе уже присутствует и наблюдения других агентов можно сопоставлять с собственными аналогичным образом. Проблемой является вычислительная сложность подхода.

1.7 SLAM Constructor framework

Объём 1 страница

Некоторые описанные выше подходы реализованы во фреймворке, на базе которого и реализован предлагаемый алгоритм.

Глава 2. Решение задачи SLAM стаей агентов

2.1 Высокоуровневая схема многоагентного SLAM

2.1.1 Архитектура многоагентного SLAM

Объём 4 страницы

Высокоуровневая схема многоагентного SLAM обязательно включает в себя одноагентный SLAM, однако не обязательно, чтобы каждый агент одинаково выполнял одноагентный алгоритм. Можно выделить различные архитектурные подходы, описывающие иерархию агентов в многоагентном SLAM.

- звезда с сервером
- звезда с маршрутизатором
- распределённая сеть
- связь каждый с каждым.

Дополнительные задачи:

- протокол связи между агентами
- map merging
- консенсус

2.1.2 Предположения и допущения

Объём 2-3 страницы

Идеальный многоагентный SLAM — это алгоритм, где заранее неизвестно взаимное расположение агентов, где нет узкого горлышка в виде сервера и где агенты связываются друг с другом только по мере возможности или необходимости для обмена информацией. Однако большинство предлагаемых сейчас решений не соответствуют всем этим требованиям. Например, наиболее распространены алгоритмы, требующие заранее известного расположения агентов. Существуют алгоритмы, где агенты помечаются яркими ярлыками, которые распознаются видеокамерами, установленными на других агентах, и с их помощью определяется взаимное расположение. Однако такой подход требует использования аппарата распознавания изображений, что является роскошью для низкопроизводительных устройств. Кроме того обеспечить яркими ярлыками реальных роботов — сложновыполнимая задача в индустрии.

2.2 Графовый многоагентный SLAM, `cg_mrslam`

Объём 2-3 страницы

Как говорилось выше, наиболее подходящим для «наивного» решения задачи многоагентного SLAM является графовый подход. Он был реализован, например, [`cg_mrslam`]. В данной конкретной работе для обеспечения производительности было выдвинуто допущение об известном взаимном расположении агентов.

Но если представить себе алгоритм без такого допущения, схема графового многоагентного лазерного SLAM требует доработки только в области определения, что агенты находятся недалеко друг от друга. В [icra18] эту проблему решили разбиением окружения на кластеры и определением агента, в каком кластере он сейчас находится. За поиск измерения в графе отвечает нейронная сеть.

2.3 Одногипотезные неграфовые многоагентные SLAM

Объём 2 страницы

В данном разделе описаны подходы, стоящие на заре многоагентных систем, такие как Multi-FastSLAM или vision-based Centralized SLAM. Общая проблема всех этих алгоритмов в том, что они либо строят итоговую карту на некотором сервере, либо что каждый агент на выходе получает различную карту. Причина этого в том, что для однопоточного алгоритма невозможно создать судью, который вовремя бы определил, что в процессе определения положения произошла ошибка, и алгоритм начал работать неправильно.

2.4 Вопросы многоагентного SLAM

Объём 2 страницы

- Как организовать отказоустойчивую систему без сервера
- Когда начинать взаимодействие между агентами
- Кто и как часто будет объединять карты
- Что делать в случае конфликта карт
- Что считать выходным результатом
- Когда целесообразно применять многоагентный SLAM по сравнению с одноагентным

пока 44 страницы.

Глава 3. Требования к разрабатываемому алгоритму

3.1 Область применения алгоритма

3.1.1 Ограничения и допущения, накладываемые на окружающую среду

Объём 2-3 страницы

Агенты работают в indoor окружении, которое может быть произвольно по размеру. Алгоритм должен хорошо работать и для помещений размером 5 на 5 и 100 на 100. Это означает, что его сложность должна расти не более, чем линейно в зависимости от размера помещений. Кроме того, будем считать, что первичное взаимное расположение агентов неизвестно. Поэтому алгоритм должен вычислять это расположение самостоятельно.

3.1.2 Ограничения и допущения, накладываемые на аппаратную составляющую агентов

Объём 2 страницы

Агенты должны быть оснащены лазерным дальномером, чтобы снизить нагрузку на вычислительный процессор (обрабатывать изображения сложнее). Кроме того, агенты должны быть оснащены датчиком одометрии, чтобы упростить процесс скан матчинга.

3.2 Оценка сложности алгоритма

Объём 2-3 страницы

Связь между агентами происходит считанное количество раз, но сложность обмена информацией должна быть минимальна. Чтобы снизить количество транзакций, необходимо уменьшить процесс «разговора» до минимума: запрос-ответ. Объём передаваемой

информации должен расти не более чем линейно в зависимости от роста размеров окружения.

3.3 Расчёт точности алгоритма исходя из заданных аппаратных ограничений

Объём 4-5 страниц

Большинство современных алгоритмов рассчитаны либо на офлайн выполнение, либо на высокопроизводительную аппаратную составляющую.

Предлагаемый алгоритм должен работать на любом ЦПУ, а значит он должен предоставлять ответ в любой момент времени выполнения. То есть чем больше времени даётся алгоритму, тем точнее будет результат. Однако хоть какой-то результат должен быть получен после любого произвольного времени, данного на работу.

Для этой концепции отлично подходит алгоритм монте-Карло.

Причём его следует использовать не только в качестве основного алгоритма скан матчинга, но и в процессе map merging-a. Скорость сходимости такого алгоритма $1/\sqrt{n}$, что говорит о необходимости увеличения выборки. Однако, существует зависимость между количеством итераций и дисперсией погрешности.

Глава 4. Архитектура компонентов разработанного алгоритма многоагентного SLAM

4.1 Высокоуровневая схема алгоритма

Объём 2 страницы

Каждый агент выполняет одноагентный SLAM алгоритм, затем когда агенты находятся в окрестности друг друга выполняется процесс обмена информацией. Окрестность можно определить, например, при помощи радиолокации. Как только агент получает чужую карту и скан, он встраивает их в собственную карту и выполняет объединение карт. Одновременно с этим такой же алгоритм происходит и на другом агенте.

4.2 Выбор алгоритма на каждом агенте

Объём 1 страница

Поскольку необходимо заботиться о скорости работы алгоритма, нельзя использовать графовые алгоритмы или фильтры частиц. Существует работа, в которой многоагентный SLAM реализован с использованием фильтра частиц, однако при передаче частиц другому агенту они все усредняются, что приводит порой к неверному ответу.

Поэтому остаётся использовать только однопотенциальные неграфовые SLAM в качестве базового алгоритма. Может подойти *tinySLAM*, однако, как говорилось выше, существует его модификация с использованием более сильной теории Демстера-Шафера — *vinuSLAM*.

4.3. Объединение карт

4.3.1. Вопрос о подходящем моменте для обмена информацией

Объём 2-3 страницы.

Для того, чтобы объединить информацию, накопленную несколькими агентами, необходимо знать их взаимное расположение. Идеальной была бы ситуация, когда обмен информацией происходит в тот момент, когда роботы находятся в одной точке. Также подойдёт случай, если агенты «видят» друг друга и могут оценить расстояние и угол поворота. В

разработанном алгоритме предлагается третий подход. Связь устанавливается, когда агенты находятся в радиусе 0,5-1 метра друг от друга.

4.3.2. Определение взаимного расположения

Объём 2-3 страницы

Поскольку агенты не оснащены видеокамерой, отследить другого агента по лазерному дальномеру невозможно. Предполагается, что связь началась, когда агенты находятся недалеко, а значит они снимают примерно одинаковые наблюдения в момент связи. Агент передаёт другому агенту собственные наблюдения, а тот в свою очередь применяет к ним процесс скан матчинга и таким образом определяет взаимное расположение.

4.3.3. Консенсус в определении финальной карты

Объём 2 страницы

Поскольку каждый агент выполняет однопоточный алгоритм, который не предполагает изменения карты в процессе работы, может произойти такая ситуация, когда агенты побывали в одном и том же месте, но в результате ошибки, построили разную карту местности. В этом случае необходимо прийти к консенсусу, какую из построенных карт принимать, как правильную

4.4. Программная реализация

Объём 1 страница

Реализация происходит в рамках slam-constructor на языке C++