


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

ОТЧЕТ
по научно-квалификационной работе
аспиранта 5-го семестра

Студент гр. 6935



Филатов Ан. Ю.

Руководитель д.т.н., профф.



Экало А. В.

Санкт-Петербург

2019

Цель работы

Целью работы является анализ работы алгоритмов поиска взаимного расположения агентов в задаче многоагентной локализации и построения карты. Особенностью такой работы является существования различных структур таких алгоритмов, что приводит к зависимости метода определения взаимного расположения от архитектуры алгоритма

Введение

Для решения задачи SLAM ключевым вопросом является модель представления карты. Одним из вариантов карт являются карты препятствий (Occupancy grid map). Использование таких карт целесообразно, когда с датчиков поступают зашумленные данные, либо робот нет априорной информации об окружающей среде. Такие карты используют автопилоты, роботы-спасатели и другие роботы, которые используют для перемещения информацию о местности вокруг себя. Построение карты при помощи данных только с одного робота происходит медленно и с ошибками, исправить которые затруднительно. Преодолеть эти недостатки можно, если использовать более одного робота, а карты, которые они строят, соединять в одну. Сложность заключается в том, чтобы исправлять конфликты, которые возникают при слиянии карт.

Карты препятствий, полученные с помощью алгоритмов SLAM, в одном и том же окружении могут различаться. Появляется необходимость построения объединённой карты, полученной путем слияния нескольких исходных карт препятствий, обнаружение и устранение конфликтов на которой, является важной частью этой задачи.

Архитектура решения

В [1] Алгоритм разбит на два этапа: стохастический поиск взаимного расположения, а затем применение функции оценки качества наложения

Оценка качества наложения состоит из нескольких этапов.

Сначала авторы вводят алгоритм вычисления похожести изображений (стр.5 вычисление ϕ). Надо заметить, что ϕ - это расстояние между изображениями. Чем оно меньше, тем более изображения похожи. Затем идёт демонстрация, как можно

вычислять это быстро, используя d-map. Авторы признают, что карты могут быть похожими только в небольшой области, поэтому они дополняют такую функцию оценки похожести таким образом, что прибавляют к её значению количество различающихся ячеек и отнимают количество совпадающих. Мотивация такая: самое лучшее наложение должно иметь наибольшее значение функции ϕ , при этом совпадающих ячеек оно имеет больше всех, а различающихся - меньше всех, поэтому разница dis-agr у самого лучшего наложения минимальна. Следовательно речь идёт о поиске такого наложения, чтобы сумма ϕ и dis-agr была наименьшая. Если убрать слагаемое dis-agr получим в чистом виде метод градиентного спуска. Если добавить это слагаемое с некоторым коэффициентом, то получим метод градиентного спуска с дополнительным возмущением.

Таким образом, идея авторов состоит в поиске наилучшей трансформации между картами, а не поисках и разрешении конфликтов в картах.

Скорость работы: предлагаемые авторами реализация мержит карты размером 400×400 за 1-2 минуты (в это время включён также поиск наилучшей трансформации между картами)

Сложность реализации в рамках SLAM конструктора:

Если не реализовывать поиск взаимного расположения между картами, то сложность низкая.

Новизна идеи и заявленная точность:

Статья датируется 2006-м годом, а, следовательно, идея не нова. По точности: авторы предлагают методику для оценивания качества слияния карт. На одной из карт по этой метрике точность составила 98%

Ограничения на окружающую среду и агентов

Предлагаемый алгоритм будет использоваться агентами, находящимися в помещении, что означает, что размер исследуемого пространства не превосходит нескольких сотен квадратных метров. Во-первых это означает, что невозможно использовать GPS, поскольку погрешность определения местоположения с помощью GPS составляет несколько метров, и такая погрешность недопустима в рассматриваемом случае. Это означает, что в качестве датчика можно использовать лазерный дальномер, эффективный радиус которого измеряется десятками метров. Выбор лазерного

дальномера в качестве сенсора позволяет строить карту в виде сетки занятости. Таким образом карта, построенная в результате работы алгоритма, будет представлять собой план здания, по которому перемещаются агенты.

Другим важным замечанием является тот факт, что алгоритм не решает задачу построения траектории и управления мобильной платформой для исследования окружения. Задачей разработанного алгоритма является только обработка измерений, полученных от лазерного дальномера а также датчика одометрии, и построение карты.

Помещение, в котором перемещаются агенты не должно содержать динамически изменяющихся областей. Появление редких и небольших шумов возможно, например, перемещение людей или других мобильных платформ. Однако не допускается перемещение мебели или колонн в помещении, особенно в моменты, когда агент не наблюдает за соответствующей частью окружения.

Следующим допущением является отсутствие перепада высоты, чтобы все наблюдения производились в одной плоскости. Кроме того для достижения этого требования необходимо плотно зафиксировать лазерный дальномер на агенте.

Обязательным также является условие, что все агенты, выполняющие алгоритм SLAM, оборудованы одинаково: они не только имеют одинаковые вычислительные возможности, но и также должны обладать одинаковой мобильной платформой, а также обладать одинаковыми лидарами. Это требование обусловлено тем, что каждый агент выполняет одинаковый алгоритм, поэтому структура выходных данных у агентов должна совпадать. Если, например, вычислительные мощности одного из агентов окажутся больше, чем у другого, то качество построенной карты у “слабого” агента окажется хуже вследствие нехватки времени на обработку данных. В этом случае карта, полученная в результате слияния будет не только содержать области с разной степенью чёткости, но также в случае пересечения некоторых областей менее чёткая карта может внести значительную погрешность в результат.

Область применения предлагаемого алгоритма - это малобюджетные, а следовательно и низкопроизводительные, агенты, поэтому алгоритм не должен содержать шагов, на работу которых существует ограничение по времени (например обработку новых измерений). Для того, чтобы алгоритм мог быть запущен на любых устройствах, он должен следовать такому условию, что чем больше времени алгоритм получает на

работу, тем более точный результат возможно получить. Такому принципу работы следует, например, стохастический скан матчер Монте-Карло. Его идея такова, что они ищет наиболее позицию агента, выбирая её случайным образом из области поиска. Понятно, что чем больше времени получит такой скан матчер на работу, тем больше вероятность выбрать позицию, наиболее близкую к оптимальной.

Поскольку разработанный алгоритм будет применяться для мобильных роботов, необходимо реализовать его таким образом, чтобы он мог быть запущен на максимальном количестве различных аппаратных устройствах. В настоящее время существует огромное количество разнообразных платформ, которые могут выполнять алгоритм SLAM, начиная от Turtlebot, Duckiebot - тележки, оснащённые простым микропроцессором, и заканчивая pr2-bot, которые представляют собой сложную систему взаимодействующих модулей.

Чтобы иметь возможность запустить высокоуровневый алгоритм на разных конфигурациях роботов существует большое количество различных фреймворков. Такие фреймворки представляют собой прослойку между аппаратной и программной составляющими любого робота. Степень интеграции с аппаратной частью также может варьироваться. Так, например, есть целый класс различных фреймворков, называемых DDS (Data distribution service)[2], задача которых предоставить обмен сообщениями между различными узлами программы. Следует заметить, что шаблон читатель-писатель, который реализован в таких фреймворках, - это широко используемый подход при программировании роботов, поскольку он позволяет всем узлам агента выполнять свои функции децентрализованно, обмениваясь данными с “соседними” узлами.

Выводы

В ходе выполнения работы были рассмотрены существующие методы определения взаимного расположения агентов в задаче многоагентной локализации и построения карты. Большинство работ связаны с анализом непосредственно карт, при этом позиции агентов во внимание не принимаются. Однако подход, использующий априорное знание о позиции агента на карте может дать некоторые преимущества. Если предположить, что обмен информацией между агентами происходит в момент,

когда они находятся в непосредственной близости, то наблюдения каждого агента могут быть сопоставлены с картой другого агента, что позволит вычислять взаимное расположение напрямую, а также агентам быть «судьями» друг для друга.

Список используемой литературы:

- 1) A. Birk and S. Carpin, "Merging Occupancy Grid Maps From Multiple Robots," in *Proceedings of the IEEE*, vol. 94, no. 7, С. 1384-1397, July 2006.
- 2) G. Pardo-Castellote, "OMG Data-Distribution Service: architectural overview," *23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings.*, Providence, Rhode Island, USA, 2003, С. 200-206.