

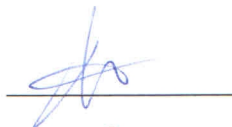
МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по научно-исследовательской деятельности
аспиранта за 4 семестр

Аспирант

направление 09.06.01

специальность 05.13.11



Филатов Антон Юрьевич

Руководитель д.т.н., проф



Экало Александр Владимирович

Санкт-Петербург

2018

Цель работы

Происследовать существующие решения для многоагентного SLAM, выработать общую модель и предложить собственный вариант такого алгоритма.

Введение

Задачу одноагентного SLAM (Simultaneous Localization And Mapping) невозможно считать целиком решенной, не смотря на то, что существуют подходы, работающие с допустимой точностью на некоторых входных данных. Проблема всегда в том, что если во время работы алгоритма происходит сбой, его практически невозможно идентифицировать, не имея некоторого арбитра извне. Решить эту проблему призван многоагентный SLAM, где каждый агент является арбитром для других агентов.

Описание проделанной работы

Среди авторов, занимающихся задачей SLAM, поставленной перед одним роботом (в будущем для краткости будем обозначать такую задачу single SLAM или одиночный SLAM), можно выделить «классика», Себастьяна Трона. Его книга «Simultaneous Localization and Mapping» не является самой первой работой, в которой упоминается эта задача, однако в ней он подробно описывает постановку задачи, а также указывает существующие на тот момент (2008 год) методы решения этой задачи.

В 2005 году этот автор выпускает статью [1], где представляет своё видение решения задачи SLAM для нескольких роботов, работающих в команде. Эта работа не является самой первой, которая расширяет задачу одиночного SLAM на нескольких роботов, однако её по праву можно считать классической.

Роботы в данном подходе могут обмениваются информацией относительно редко (возможно даже всего один раз по завершению работы), и цель такого обмена — слияние локальных карт, построенных каждым роботом. Идея в том, что начальное расположение роботов неизвестно, и выполнить слияние карт необходимо, не обладая этой информацией. Поэтому задача декомпозируется на две части: выяснить, пересекаются ли локальные карты вообще; если пересекаются, то по какой области и какова трансформация между картами.

Карты представляют собой набор особых точек (которые выделяются на кадре или скане) и их координат. При слиянии необходимо выделить общие особые точки, а также дополнить карту новыми точками, полученными от другого робота. Для достижения этой цели особые точки на одной из карт разбиваются на триплеты и производится анализ, присутствует ли такой триплет на другой карте с точностью до поворота и смещения. Этот процесс осуществляется с помощью SB-tree поиска и имеет сложность $O(N * \log N)$, где N — это количество особых точек.

Тестирование алгоритма проводилось на улицах города, каждый робот проезжал значительное расстояние. О характеристиках роботов авторы не уточняют, известно лишь, что им удавалось в режиме реального времени обрабатывать изображения с видеокамер и выделять на них особые точки. Из этого можно сделать вывод, что технические характеристики роботов были сравнительны с настольными компьютерами, но не слабее.

Одной из первых работ, в которых рассматривается задача многоагентного исследования окружения является [2], 1997. Идея предложенного алгоритма с высоты современной постановки задачи выглядит чересчур просто: в каждый момент времени движется, только по прямой, только один робот, цель которого построить «полоску» карты, которую он сможет пронаблюдать с помощью своих сенсоров. Остальные роботы не двигаются, пока первый робот не остановится. Затем работу начинает второй робот, проделывает то же самое, а остальные ожидают завершения его работы. После этого все измерения накладываются друг на друга и производится оценка точности одометрии путём сопоставления общих частей в измеренных «полосах». Очевидно, что такой

подход не имеет интереса в контексте данной работы, поскольку цель текущего исследования в ускорении решения задачи SLAM за счёт увеличения количества роботов, а не наоборот. Поэтому перейдём к рассмотрению более современных алгоритмов.

В более современных работах, однако, по-прежнему можно встретить довольно сильные ограничения, в которых авторы предлагают свои решения. Например, существует работа [3], где предполагается, что известно начальное взаимное расположение роботов. Это, безусловно, очень существенное требование, которое, однако, позволяет пользоваться некоторыми преимуществами. Так, например, в описываемой работе, когда роботы начинают процесс обмена информацией, они обмениваются не построенными картами, а необработанными наблюдениями, которые они могут встраивать в собственную карту, поскольку знают начальное относительное расположение, а также могут с некоторой точностью определить перемещение друг друга в течение работы программы.

Особенностью этой работы также является способ представления карты. Авторы предпочли использовать графовое представление, где каждая вершина — это позиция робота, а также набор сканов, которые робот может пронаблюдать с этой позиции, а ребро графа — это трансформация (перемещение и поворот) между вершинами. Такой подход был выбран потому, что в этом случае относительно легко встраивать наблюдения другого робота в собственную карту. Дело в том, что трансформация между роботами оценивается во время связи между ними, а этого, вкупе с сырыми данными робота, достаточно, чтобы построить новую вершину графа. Останется лишь оптимизировать граф, чтобы определить, не строятся ли в таком случае дублирующиеся вершины, а также уточнить состояние графа, если пришедшие от другого агента наблюдения сильно расходятся с собственными.

Если вспоминать эволюцию развития одноагентного SLAM, то первым шагом к ускорению его работы был переход от EKF к алгоритму, получившему название FastSLAM. В этом случае стали предполагать, что все особые точки, из которых состоит карта независимы, что позволило существенно уменьшить размерность матрицы ковариаций, которую необходимо было хранить и обращаться в классическом подходе EKF. Вследствие такого упрощения появляется возможность обрабатывать новые наблюдения чрезвычайно быстро, а освобоившиеся вычислительные ресурсы потратить на увеличение точности работы. FastSLAM часто используется в совокупности с так называемым фильтром частиц. Частица — это гипотеза о состоянии мира и о положении агента на ней. Задача фильтра — определять наиболее вероятную гипотезу, отсеивать гипотезы, вероятность которых крайне мала и держать в памяти и обновлять все правдоподобные гипотезы.

В качестве расширения такого подхода на несколько агентов в [4] предлагается следующий подход. Каждый робот выполняет одноагентный fastSLAM, самостоятельно выделяя ориентиры на карте. В момент, когда происходит

обмен информацией, появляется задача слияния карт. Сложность в том, что карты для каждого робота хранятся в нескольких гипотезах, из которых возможно выделить наиболее вероятную, однако нельзя исключать из рассмотрения остальные гипотезы. В данном подходе между агентами передаётся математическое ожидание карты - среднее арифметическое с весами каждой частицы. После этого включается алгоритм ближайшего соседа для определения совпадающих ориентиров. В общем алгоритм занимает $O(mn^2)$, где m — число гипотез, а n — количество ориентиров.

Очевидным недостатком такого подхода является примитивный способ передачи гипотез: вместо группы гипотез передаётся одна «средняя по больнице». Такой подход приводит к значительному ухудшению результирующей карты. Необходимость для каждого робота искать ориентиры самостоятельно также влечёт за собой ухудшение карты, поскольку разные роботы могли выявить разные ориентиры, либо пропустить некоторые из них, либо распознать ориентир в том месте, где его нет. Этим всем приходится платить за скорость работы алгоритма.

В большинстве работ предполагается наличие знания о взаимном расположении роботов. Это знание может быть получено разными путями, самый простой из которых — это изначально указать каждому агенту, где располагаются остальные. При этом каждый агент сохраняет свою траекторию движения, а при обмене информацией с другими агентами делится ей, так чтобы у них была возможность определить текущее взаимное расположение роботов. Этот подход имеет недостаток в том, что если один агент ошибается во время собственной работы и оценивает свою траекторию неверно, то это приводит к ошибкам у всех роботов. Существует похожий, но отличающийся, подход для определения взаимного расположения. Он подразумевает наличие некоторого сервера, который хранит знание обо всех агентах, возможно, даже их картах и траекториях. Это позволяет выделять агентов, у которых возникают ошибки во время выполнения, поскольку сервер имеет возможность оперативно сравнить карты, построенные всеми агентами. Такая клиент-серверная архитектура была, например, реализована в [5] или в [6]. Недостаток этой архитектуры также легко заметить: как только появляются помехи или перебои в связи между клиентами и сервером, как работа всей системы нарушается, поскольку агенты не могут взять на себя роль сервера.

Для того, чтобы получить информацию о взаимном расположении агентов, можно пойти совершенно другим путём. Вместо априорной оценки расположения роботов можно извлечь его из данных, которые роботы получают с помощью сенсоров. Например, существует подход, когда агенты начинают обмен информацией, когда они физически сталкиваются друг с другом, то есть почти находятся в одной точке в пространстве. В этом случае процесс обмена наблюдаемыми данными или построенными картами упрощается, поскольку нет необходимости вычислять преобразование между

картами или прямыми наблюдениями, а остаётся лишь задача обновлять собственную карту.

Недостатком такого решения является ограничение в свободе перемещения роботов. Поскольку одна из главных задач робота исследовать неизвестное пространство, роботу необходимо планировать свой маршрут максимально эффективно. Однако для того, чтобы роботы посетили одну точку необходимо либо продумывать траекторию заранее и рассчитывать время, когда они смогут оказаться на одном месте, либо инициировать у роботов в какой-то момент алгоритм перемещения в одну точку, а это влияет на эффективность маршрута. Столкновение роботов может произойти и случайно, но в этом случае это событие, на которое невозможно рассчитывать, оно может не произойти вовсе за всё время выполнения роботами исследования.

В статье [7] предлагается альтернативное решение задачи о нахождении взаимного расположения агентов. Авторы предлагают снабдить роботов панорманой камерой, с помощью которой агенты будут иметь возможность находить друг друга. Для того, чтобы на кадре можно было легко отделить робота от других объектов, предлагается также снабдить роботов яркой светящейся меткой. В статье описывается алгоритм по которому можно идентифицировать на кадре эту метку, вычислить угол, на который необходимо повернуться, чтобы «повернуться лицом» к другому роботу. У авторов также есть наработки, как по полученному кадру определить расстояние до робота, однако они сами признаются, что это лучше сделать при помощи лазерного дальномера.

Таким образом можно получить расстояние и угол поворота между двумя агентами, когда они находятся на некотором расстоянии друг от друга, что позволяет не нарушать порядок следования по маршруту лишь для того, чтобы несколько роботов могли обмениваться информацией.

Однако вне лабораторных условий такой подход слабо применим, поскольку сложно себе представить индустриального робота, на котором бросается в глаза хорошо различимая метка. Кроме того при таком подходе роботы должны взаимодействовать при очень хорошем освещении, а иначе задача распознавания метки становится чересчур трудоёмкой.

Существует также подход к реализации многоагентного алгоритма, который не учитывает взаимное расположение робота. Понятно, что в таком подходе не возможно пользоваться сырыми данными других агентов, однако есть возможность обмениваться построенными локальными картами и объединять их, не имея знаний о трансформации между ними. Такой подход описывался в начале этого раздела, он использовался в статье [1]. Особенность работы Трона в том, что карта представлена в виде особых точек, но также существуют и решения для другого представления карты. Так, например, в работе [8] показано, что можно определить области, в которых пересекаются локальные карты, используя преобразование карты в область Хафа. То есть на карте определяются прямые линии, вычисляются их углы наклона. Далее на двух

картах ищутся места, где содержится одинаковое количество прямых линий, имеющих одинаковый угол наклона. Это позволяет вычислить поворот между двумя картами.

С помощью похожего подхода можно также вычислить и смещение по оси x и y . Однако такой подход работает только в том случае, когда две карты имеют большую область пересечения. Поэтому подход Трона с поиском триплетов из особых точек выглядит более устойчивым к ложноположительному нахождению пересечения карт.

Заключение

В ходе исследования были изучены существующие алгоритмы, решающие задачу многоагентного SLAM. В качестве алгоритма, который может быть реализован и который совмещает в себе достоинства всех алгоритмов был выбран подход, реализующий несколько одноагентных алгоритмов, работающих автономно, с последующим слиянием карт, построенных агентами. Взаимодействие агентов происходит по каналу один-к-одному, где каждый агент обновляет собственную карту. Траектории движения всех роботов также хранятся на каждом агенте.

Список литературы

- [1] Thrun S., Liu Y. Multi-robot SLAM with sparse extended information filters //Robotics Research. The Eleventh International Symposium. – Springer, Berlin, Heidelberg, 2005. – С. 254-266.
- [2] Rekleitis I. M., Dudek G., Milios E. E. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error //IJCAI. – 1997. – Т. 2. – С. 1340-1345.
- [3] Lazaro M. T. et al. Multi-robot SLAM using condensed measurements //Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. – IEEE, 2013. – С. 1069-1076.
- [4] Özkucur N. E., Akın H. L. Cooperative multi-robot map merging using fast-SLAM //Robot Soccer World Cup. – Springer, Berlin, Heidelberg, 2009. – С. 449-460.
- [5] Forster C. et al. Collaborative monocular slam with multiple micro aerial vehicles //Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. – IEEE, 2013. – С. 3962-3970
- [6] Castro D. A., Morales C. F., De la Rosa R. F. Multi-robot SLAM on Client-Server Architecture //Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), 2012 Brazilian. – IEEE, 2012. – С. 196-201.
- [7] Zhou X. S., Roumeliotis S. I. Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case //Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on. – IEEE, 2006. – С. 1785-1792.
- [8] Carpin S. Fast and accurate map merging for multi-robot systems //Autonomous Robots. – 2008. – Т. 25. – №. 3. – С. 305-316.