

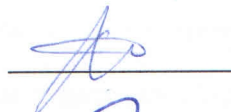
МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по научно-квалификационной работе
аспиранта за четвёртый семестр

Аспирант

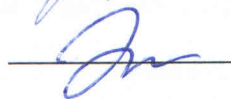
направление 09.06.01

специальность 05.13.11



Филатов Антон Юрьевич

Руководитель д.т.н., проф



Экало Александр Владимирович

Санкт-Петербург

2018

Цель работы

Целью научно-квалификационной работы является анализ схем многоагентного алгоритма одновременной локализации и построения карты, определение критериев оценки качества таких алгоритмов, а также выделение методик определения качества.

Введение

Прежде, чем подробно описывать различные архитектуры, следует остановиться на вопросе об определении взаимного расположения агентов. В простом случае агенты начинают работу, находясь в одной точке или, хотя бы, на известном расстоянии друг от друга (и с известным изначальным направлением). В этом случае неважно, когда и как данные от нескольких агентов будут объединены, поскольку в любой момент известна траектория движения агента и начальная точка. Следовательно для любого агента всегда известно его положение в глобальной системе координат.

Когда известно положение каждого агента, пусть даже с некоторой погрешностью, задача объединения наблюдений каждого агента становится тривиальной, с ней может справиться скан матчер, используемый для решения задачи одноагентного SLAM.

Следует заметить, однако, что невозможно расположить агентов абсолютно в конкретной точке или на абсолютно точно измеренном расстоянии между ними, если речь не идёт о симуляции. Поэтому взаимное расположение агентов обязательно содержит погрешность. Величина этой погрешности зависит от множества условий, но само наличие такой погрешности наводит на мысль, что алгоритм, который может обработать входные данные, содержащие случайную погрешность, может также справиться с задачей, где эта погрешность введена намеренно.

Другими словами, если известно только приблизительное расположение агентов, например, известно, что они находятся в радиусе 10 сантиметров друг относительно друга, возможно совместить данные наблюдений, которые они предоставляют.

Таким образом в задаче многоагентного SLAM появляется дополнительная задача о определении взаимного расположения между агентами. И для решения этой задачи различные авторы используют решение, основанные на разных допущениях.

Так, например, одно из решений - это оснастить мобильных агентов камерой с углом обзора в 360°, чтобы они не имели слепых зон, а также яркими хорошо различимыми метками, например, светодиодами. Тогда при определении на камере таких светодиодов будет возможно определить направление, в котором агентам необходимо двигаться для того, чтобы встретиться в одной точке. Кроме того при помощи дальномера или любого другого инструмента можно определить расстояние между агентами, и этих двух параметров (взаимный угол и расстояние) достаточно, чтобы определить координаты одного агента в координатах другого и, следовательно, приступить к задаче объединения данных.

К недостаткам такого подхода можно отнести как раз сложность установки камеры и ярких меток на агента. В лаборатории возможно соблюсти все условия, чтобы метки всегда были различимы и камера могла их распознавать, однако на практике такой метод обнаружения зависит от освещения, заряда питающей батареи и прочих других факторов.

Другим способом определения взаимного расположения является разметка окружения. Она может быть выполнена различными способами: специальные “зоны коммуникации”, QR-коды, нанесённые на пол и определяющие координаты, разметка стен или потолка - все способы разметки окружения позволяют агентам определить свои координаты в глобальной системе. Как только несколько агентов определяют собственные координаты, появляется возможность объединять их данные.

Описанные выше способы требуют некоторой подготовки агентов или окружения, однако существуют подходы, которые не требуют дополнительных данных, а основываются на данных сенсоров, используемых для решения задачи SLAM.

Схема многоагентного SLAM обязательно включает в себя одноагентный SLAM, однако не обязательно, чтобы каждый агент одинаково выполнял одноагентный алгоритм. Более того агенты могут быть технически оснащены совершенно разным оборудованием. Можно выделить различные архитектурные подходы, описывающие иерархию агентов в многоагентном SLAM.

- звезда с сервером
- звезда с маршрутизатором
- распределённая сеть

Архитектура многоагентного SLAM

Звезда с сервером.

Исторически первая архитектура многоагентного алгоритма состояла из сервера, который собирал данные от агентов и самостоятельно обрабатывал их. Другими словами единственную роль, которую играли агенты - это роль сенсоров. Кроме того, сервер мог вовсе не предоставлять обратной связи агентам. Достоинства такой структуры очевидны: даже если начальное взаимное расположение агентов неизвестно, сервер обладает знаниями о том, что “видит” каждый из наблюдателей в каждый момент времени. То есть сервер решает задачу одиночного SLAM, имея гораздо больше сенсоров. Недостаток такой структуры невозможно игнорировать: система очень зависима от качества связи с сервером. Кроме того, в случае, если сервер по каким-то причинам выйдет из строя, мобильные агенты будут не в состоянии выполнять хоть какие-то шаги алгоритма SLAM, помимо наблюдения за окружением.

В предыдущем разделе поднимался вопрос о способе вычисления взаимного расположения агентов. В том числе рассматривался подход, когда агенты снабжаются дополнительной камерой и яркими метками. В рамках рассматриваемой архитектуры необходимо, чтобы агенты передавали данные с этой камеры на сервер для выполнения вычисления взаимного расположения.

Другими словами сервер выполняет роль единственного вычислительного узла. Такая архитектура значительно удешевляет агентов - они играют роль сенсоров и не должны выполнять никаких вычислительных операций. Таким образом система свободно масштабируется: для увеличения количества агентов в сети достаточно приобрести дешёвую платформу, оснастить датчиками и средством передачи информации.

Звезда с маршрутизатором.

Звезда с маршрутизатором является модификацией архитектуры, описанной выше. Отличительной особенностью является распределение вычислений между агентами.

Другими словами каждый агент решает собственную задачу SLAM, но связь с другими агентами осуществляется через центральный узел.

При таком подходе недостаток звезды с сервером немного видоизменяется: с одной стороны “узкое горлышко” остаётся в виде маршрутизатора, однако застраховаться от выхода его из строя можно, установив резервный маршрутизатор. В системе с сервером в центре звезды ставить дублирующий центральный узел часто не рентабельно, а в данной архитектуре установка резервного оборудования целесообразна.

Необходимо отметить, что в этом случае каждый агент должен обладать достаточными вычислительными мощностями, чтобы снимать наблюдения, решать задачу SLAM и обмениваться данными с маршрутизатором. Плюсом такой системы является децентрализация вычислений, и, следовательно, упрощение масштабирования.

Распределённая сеть.

Следующим этапом децентрализации вычислений является отказ от маршрутизатора. Система тогда выглядит следующим образом: агенты выполняют связь друг с другом только тогда, когда они находятся в непосредственной близости друг от друга. Это может достигаться, например, разбиением пространства на участки, где каждый агент становится ответственным за определённый участок пространства, и когда другой агент заходит на “чужую” территорию, он знает, с каким агентом ему следует связаться.

Другим способом организовать взаимодействие между агентами является общение при столкновении агентов. Как только один агент замечает другого посредством собственных датчиков, агенты могут сблизиться, если необходимо, и начать процесс идентификации друг друга и обмена информацией.

В любом случае система становится абсолютно стабильной к потере любого узла, поскольку связь между агентами лежит в области задач самих агентов. Если, например, один из них физически застрял, или повредил датчик, или израсходовал заряд батареи, остальные агенты смогут продолжать работу, даже без вышедшего из строя. Безусловно, выход из строя одного из узлов может повлиять на качество построенного результата, однако он не повлечёт остановку работы всей системы.

Также необходимо упомянуть задачи, которые встают дополнительно, помимо задачи одноагентного SLAM.

Самым важным является вопрос протокола общения между агентами. Если система содержит сервер, то ответ на этот вопрос тривиальный. Однако при использовании других архитектур к протоколу общения следует отнестись внимательно. Самые популярные решения - это перед началом взаимодействия агентов либо переместиться в одну точку, либо оценить взаимное расположение друг друга. Затем каждый агент передаёт собственные накопленные данные другому агенту, и связь на этом заканчивается. Хотя возможна ситуация, когда агенты верифицируют объединённые карты.

Другим важным вопросом является способ объединения карт. Как только решён вопрос вычисления взаимного расположения агентов, необходимо соединить карты, построенные агентами. Сложность в этом вопросе состоит в том, что даже если карты построены без ошибок, они построены при помощи вероятностных подходов, а следовательно любая погрешность в измерении может привести к неполному соответствию построенных карт. Например, если рассмотреть многоагентный EKF SLAM, то можно представить ситуацию, когда оба агента наблюдали одно и то же препятствие, но один из них допустил погрешность в вычислении координат этого препятствия. Агенты не могут оценить, какая из карт является более точной и вынуждены либо выбирать одну из гипотез, либо строить среднее.

Из описанной выше проблемы возникает вопрос консенсуса карт. Если одна из карт содержит ошибку, или если каждый из агентов построил кардинально отличающиеся друг от друга карты, необходимо принимать решение, как будет выглядеть итоговая карта. Важно ответить на вопрос, будет ли она одна или каждый агент будет предоставлять его собственный вариант карты.

Выводы

В ходе выполнения работы были рассмотрены следующие схемы алгоритмов многовгентной локализации и построения карты:

- звезда с сервером
- звезда с маршрутизатором
- распределённая сеть

Вне зависимости от архитектуры существует два подхода к определению качества работы алгоритмов:

- тестирование в симуляторе. Такое тестирование позволяет сравнить построенную карту и траекторию с точно известным окружением, что может быть использовано для определения точности, но полученные результаты не позволяют учитывать все возможные шумы, возникающие в реальном мире.

- тестирование на данных, полученных с реальных датчиков. Недостатком такого тестирования является отсутствие знания об истинной карте и траектории, поскольку они могут быть измерены только с некоторой ненулевой погрешностью