А.Ю. Филатов

**Общий подход к разработке графового алгоритма SLAM.**

*SLAM (Simultaneous Localization and Mapping) – задача одновременного построения карты и определения на ней собственного положения, которая ставится перед мобильными роботами. Такие роботы оснащаются различными датчиками наблюдения за окружением, сенсорами. Алгоритмы, решающие задачу SLAM различаются по многим параметрам, таким как тип сенсоров, способ представления данных, полученных от них, способ представления карты. Одним из наиболее робастных (устойчивых к изменению входных данных) алгоритмов является графовый SLAM. В работе описана общая концепция к разработке такого алгоритма и сделаны первые шаги по его реализации.*

**Графовый SLAM, мобильные роботы, оптимизация графа**

**I. Введение**

Одна из наиболее часто встречающихся задач для мобильных роботов – это задача перемещения от точки А до точки В. Хорошо, если движение происходит по прямой, на пути нет препятствий, тогда эта задача вырождается до тривиальной. Но как только требуется объехать препятствие или даже группу препятствий, то роботу становятся необходимы сенсоры, а также алгоритм обработки данных, полученных от них. Если же задача робота не просто объехать препятствие, а ещё и проложить маршрут или запомнить окружение, то в таком случае используются алгоритмы, решающие задачу SLAM [1].

Задачу SLAM можно декомпозировать на несколько задач:

* *Выбор вида используемых сенсоров.* Обычно используются видеокамеры или лазерные дальномеры. С помощью сенсоров первого типа можно получить трёхмерную карту окружающего пространства, и алгоритмы SLAM, использующие такие сенсоры входят в класс задач 3D SLAM. С помощью лазерных дальномеров получается как будто план здания, такие алгоритмы, соответственно, входят в класс 2D SLAM. Кроме того роботы всегда оснащаются датчиками одометрии, которые предоставляют априорную оценку перемещения робота. Однако эти датчики накапливают погрешность, поэтому данные, полученные от них, необходимо уточнять.
* *Представление карты.* Очевидно, что представление карты изменяется в зависимости от вида сенсоров, но всё же для обоих видов сенсоров можно найти общие черты при классификации способов представления карты. Так, карта может быть основана на выделении особых точек на ней; в таком случае вместо непосредственно карты, можно получить совокупность хорошо различимых объектов и их взаимное расположение. Также карта может быть представлена, как двумерный или трёхмерный план здания. Но одним из наиболее перспективных способов представления карты является граф [2]. Вершинами графа являются сканы – данные с датчиков в один момент времени, а рёбрами графа – трансформация между этими сканами, то есть перемещение, которое должен проделать робот, чтобы попасть из одной вершины в другую
* *Скан матчер.* Когда снимается новое измерение, то есть появляется новый скан, его необходимо сопоставить с имеющейся картой так, чтобы выяснить перемещение робота в тот момент, когда был получен этот скан, а также обновить карту. Эту работу выполняет компонент, называемый скан матчером.

Цель данной работы – реализовать простой алгоритм SLAM, в котором карта представляется в виде графа. В качестве сенсоров был выбран лазерный дальномер, поскольку данные, полученные с него, обрабатываются быстрее, что является преимуществом для мобильных маломощных роботов.

**II. Концептуальное описание графового алгоритма SLAM**

Ключевой особенностью графового алгоритма SLAM является внутреннее представление карты в виде графа. Граф является ориентированным и может содержать циклы. В качестве вершин графа выступают т. н. ноды, в которых содержится скан или несколько сканов, полученных в некоторый момент времени; а рёбрами графа являются перемещения робота, то есть, на какое расстояние роботу необходимо проехать и на какой угол повернуться между двумя нодами, чтобы пронаблюдать скан, содержащийся во второй ноде.

Очевидным достоинством такого подхода является сохранение всех сканов, полученных в результате работы в необработанном виде, что означает, что к ним можно будет вернуться в любой момент по мере необходимости. В не графовых алгоритмах SLAM (таких как EKF-SLAM, gmapping[3] и других) сканы встраиваются в карту и более не сохраняются, таким образом, информацию об эволюции карты проще сохранять именно в графовом алгоритме SLAM.

Общая схема работы графового алгоритма SLAM показана на рис. 1. Как упоминалось выше, новые измерения, полученные от сенсора, добавляются в граф. Очевидно, это возможно сделать двумя способами: добавить новую ноду с соответствующей трансформацией или учесть сразу эту трансформацию, чтобы добавить скан в уже существующую ноду. Первый подход позволяет использовать каждый скан в будущем в чистом виде, но при этом увеличивается общее количество нод, что усложняет будущую обработку графа. Второй подход позволяет сократить количество нод, при этом каждая нода хранит в себе подкарту, а не скан. Для того, чтобы подкарта была консистентна, необходимо использовать алгоритм скан матчинга, что увеличивает время обработки одного скана, но уменьшает время оптимизации графа.

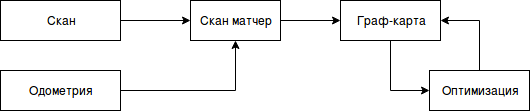
****

Рисунок 1. Схема работы графового алгоритма SLAM.

Оптимизация графа — это процесс, в ходе которого изменяются веса рёбер графа, удаляются ноды и т. д. Оптимизация обычно запускается в то время, когда робот посещает место, в котором он уже был. При этом может так произойти, что перемещения робота были сохранены с ошибками, и измерения, полученные в посещённом месте попадут не в уже созданную ноду, а в новую. Исправить эту ситуацию должен процесс замыкания циклов и оптимизация графа. На рис. 2 показан пример описанной выше ситуации. Для того чтобы исправить возникшую ошибку, необходимо в данном случае проконтролировать, что начало и конец движения совпадают.

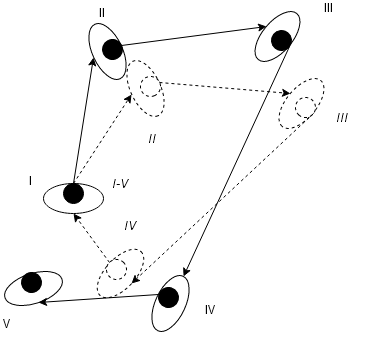


Рисунок 2. Пример перемещения робота (штрихами) и графового представления этого перемещения до запуска алгоритма оптимизации графа (непрерывные линии).

Наиболее перспективным из существующих алгоритмов графового SLAM является Google Cartographer [4]. В ходе его работы в нодах графа несколько сканов объединяются в подкарты, которые являются неделимыми единицами, и оптимизация происходит над такими нодами, не изменяя их содержимое. Недостатком этого алгоритма является невозможность его работы в режиме реального времени – процесс оптимизации графа занимает слишком много времени, хоть он и запускается не на каждой итерации алгоритма.

Разрабатываемый в представленной работе алгоритм работает таким образом, что каждый новый скан добавляется в новую ноду. Проверка, что новая нода находится в том месте, которое робот ещё не посещал, то есть что она действительно новая, производится на каждом шаге алгоритма, причём сравнение производится со всеми сохранёнными в памяти нодами.

Этот момент является предметом для улучшения алгоритма, можно производить сравнение не со всеми нодами, а с теми, которые находятся в некоторой окрестности от текущего положения. Имея приблизительные данные о перемещении робота можно оценить траекторию движения робота и область, в которой он находится. Тогда можно сравнивать каждую ноду лишь с некоторыми из этой области и таким образом уменьшить время обработки каждого нового скана. Оптимизация графа в момент замыкания цикла на данный момент производится примитивно: исправляется вес только последнего ребра в графе. Можно использовать библиотеку g2o для выполнения оптимизации графа.

Разрабатываемый алгоритм тестируется в операционной системе ROS на наборах данных, записанных в университете MIT [5]. ROS был выбран как одна из самых популярных операционных систем для мобильных роботов, в ней реализовано множество утилит, позволяющих эмулировать реальное поведение роботов. Наборы данных MIT имеют преимущество в том, что они сопровождаются вместе с groundtruth — истинным перемещением робота во время записи данных, это позволяет точно оценить ошибку работы алгоритма.

**III. Заключение**

Алгоритмы, решающие задачу SLAM являются крайне востребованными в условиях развивающейся области мобильной робототехники, поэтому крайне важно иметь возможность запускать на них робастный и точный алгоритм, решающий задачу SLAM в режиме реального времени. В этой работе описано общее устройство графового алгоритма SLAM, а также рассказано о прогрессе и перспективах разработке собственного графового алгоритма. Разрабатываемый алгоритм ещё требует исследования в некоторых алгоритмических шагах. Например, пока без ответа остаётся вопрос о способе оптимизации графа, а возможности встраивания сканов в существующие ноды, о способе замыкания циклов.

**IV. Список литературы**

1. C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on Robotics, 32(6):1309–1332, Dec 2016.

W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kmmerle, C. Dorn-hege, M. Ruhnke, A. 2. Kleiner, and J. D. Tards. A comparison of slam algorithms based on a graph of relations. In 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems , pages 2089–2095, Oct 2009.

3. B. Gerkey. Ros slam gmapping. http://wiki.ros.org/slam gmapping, 2010. [Accessed 15-Jan-2018].

4. Google. 2d cartographer backpack deutsches museum, 2016. [Accessed 15-Jan-2018].

5. M. Fallon, H. Johannsson, M. Kaess, and J. J. Leonard. The mit stata center dataset. The International Journal of Robotics Research, 32(14):1695–1699, 2013.