

Правительство Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Санкт-Петербургский государственный университет»

Кафедра системного программирования

Вьюгинов Николай Юрьевич

Метод одновременной навигации и составления карты (SLAM)

Курсовая работа

Научный руководитель:
ст.преп. кафедры СП Брыксин Т. А.

Санкт-Петербург
2016

Содержание

Введение.....	3
Обзор.....	5
Реализация.....	8
Заключение.....	13
Список литературы.....	14

Введение

Одной из самых активно развивающихся областей робототехники являются автономные мобильные системы. Важной чертой таких роботов является возможность самостоятельно осуществлять навигацию в пространстве.

Чаще всего достаточно точная и подробная карта сцены отсутствует. Поэтому активно развивающейся областью прикладной науки является построение роботом карты неизвестной местности самостоятельно. При этом роботу необходимо в некотором хорошо структурированном виде сохранить всю требуемую информацию об объектах сцены. Впоследствии эта информация может быть переиспользована для решения роботом всевозможных задач на исследуемой области пространства [1].

Большинство современных реализаций делит поставленную задачу на две основные подзадачи: задача последовательного уточнения карты(mapping) и задача пересчёта своего положения в пространстве(localization).

Решение поставленной задачи ложится на плечи автономного робота, который получает информацию о внешнем мире посредством датчиков и строит гипотезу о своём местоположении на основе обработки полученных данных. Вследствие этого решение обеих поставленных подзадач заметно усложняется из-за того, что датчики имеют некоторую модель ошибки, точно вычислить и нивелировать которую в общем случае не представляется возможным. Поэтому роботу необходимо постоянно корректировать данные о своём положении(localisation). Уточнение собственных координат осуществляется за счёт вычисления своего положения относительно некоторых выделенных объектов(landmarks). Каждому выделенному объекту сопоставляется описание, позволяющее идентифицировать его на каждом кадре, на котором он присутствует.

Методики и технологии для решения поставленной задачи получили название SLAM¹ (Simultaneous Localization And Mapping). На данный момент существует довольно большое количество реализаций и подходов, опирающиеся на аппаратные и программные возможности платформ.

Компания ООО «КиберТех» разрабатывает робототехнический контроллер ТРИК². ТРИК — это миникомпьютер, содержащий все необходимое оборудование для создания на его базе автономных робототехнических систем. Контроллер совместим с широким спектром периферийных устройств, имеет в своем составе все необходимое оборудование для управления двигателями постоянного тока и сервоприводами, а также для приема и обработки информации

¹ SLAM, https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping

² Сайт компании ООО «КиберТех», <http://www.trikset.com/>

от цифровых и аналоговых датчиков, микрофонов, видеомодулей. Контроллер снабжён цветным сенсорным дисплеем, программируемыми кнопками, есть поддержка WiFi, Bluetooth 4.0 (включая LE) и ANT.

На центральном ARM-процессоре контроллера ТРИК работает Linux, моторы и датчики программно доступны даже из shell скриптов. Поэтому автономные модели можно программировать не только на C или C++/Qt, но и на JavaScript, C#/F# (.NET), Python и Java.

Целью данной курсовой работы является реализация рабочего прототипа, решающего задачу “Одновременной навигации и составления карты” (SLAM).

Для реализации поставленной задачи предстоит решить следующие задачи:

- изучить существующие реализации;
- выбрать набор технологий, подходящий для реализации на базе контроллера ТРИК;
- реализовать рабочий прототип.

1.Обзор

Основные подходы к решению задачи локализации и создания карты делятся на классы по типу датчиков, благодаря которым робот узнаёт о состоянии окружения. Большим классом алгоритмов, решающих задачу SLAM, являются реализации, использующие алгоритмы компьютерного зрения. Ниже будет рассмотрена одна из самых универсальных и активно используемых реализаций – ORB-SLAM.

ORB-SLAM

ORB-SLAM³ является универсальным и довольно точным решением задачи Monocular SLAM [2]. Система способна в реальном времени вычислять траекторию камеры и строить разреженную трёхмерную реконструкцию сцен самых разных размеров: от рабочего стола до нескольких городских кварталов. Система отлично справляется с восстановлением довольно беспорядочных траекторий, в ней также реализованы некоторые дополнительные эвристические методы для замыкания объёмных циклов. Предусмотрена борьба со скачками (relocalisation), то есть с резкими изменениями положения наблюдателя или резким изменением сцены. В рамках ORB-SLAM существует большое количество реализаций отдельных частей алгоритма и реализована система выбора метода, который больше подходит к данной конкретной ситуации. Это приводит к отличным практическим результатам.

На рис. 1 приведён пример работы ORB-SLAM. Слева приведён снимок камеры, на котором выделены landmark-и. Справа – искомая 3D карта с отмеченными на ней положениями камеры.

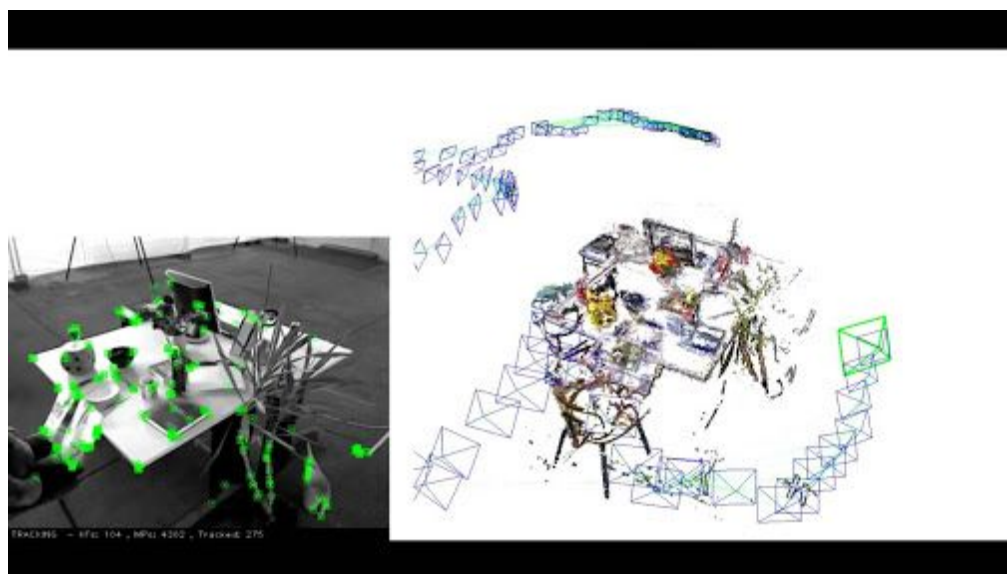


Рис. 1. Пример работы ORB-SLAM

³ ORB-SLAM, <http://webdiis.unizar.es/~raulmur/orbslam/>

DP SLAM

DP-SLAM – реализация, использующая показания лазерного дальномера и фильтр частиц⁴ для хранения гипотез о положении робота и конфигурации окружающей его сцены [4]. Фильтр частиц (последовательный метод Монте-Карло) — рекурсивный алгоритм для численного решения проблем оценивания (фильтрации, сглаживания), особенно для нелинейных случаев. Метод заключается в том, что в фильтре поддерживается множество гипотез о текущем положении робота [3]. В начальный момент генерируется случайный набор гипотез. В ходе работы алгоритма некоторые из них будут отсеиваться из-за несоответствия вторичным признакам системы, которые могут быть измерены более точно. После завершения цикла сканирования выбирается наиболее вероятная из оставшихся в фильтре гипотез – она будет искомым картой, на которой к тому же будет отмечена траектория движения исполнителя.

Для хранения и структурирования большого числа промежуточных карт авторами была реализована древовидная структура (рис. 2). Вся информация, необходимая для решения обеих подзадач: локализации (localization) и корректировка карты (mapping) помещается в одном фильтр частиц.

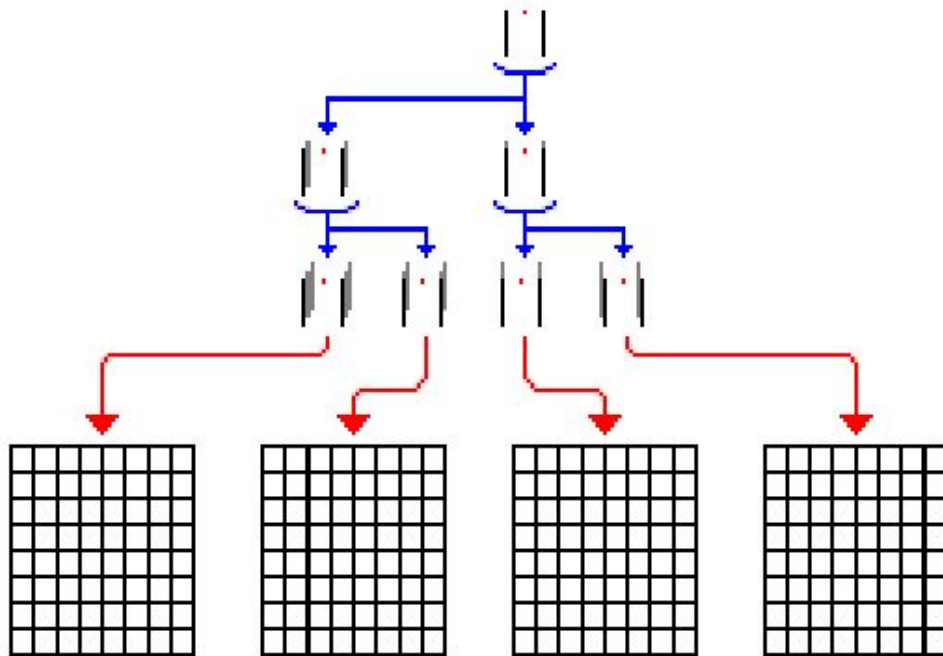


Рис. 2. Древовидная система хранения “гипотетических” карт

⁴ Particle filter, https://en.wikipedia.org/wiki/Particle_filter

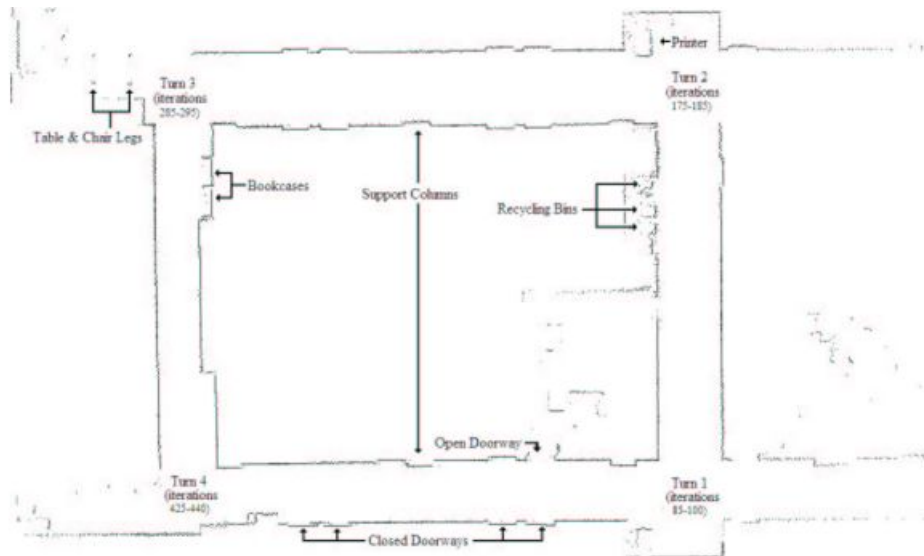


Рис. 3. Результат работы алгоритма DP-SLAM на 9000 частиц

Рассмотрим две реализации алгоритма DP-SLAM. Первая реализация одновременно поддерживает до 9000 возможных гипотез, и отсеивает только те из них, которые в силу невыполнения каких-либо условий абсолютно точно не могут соответствовать реальному состоянию системы [5]. Второй подход на каждом шаге алгоритма выбирает только одну “самую вероятную” гипотезу и обновляет карту исходя из неё. При сравнении рисунков 3 и 4 становятся очевидными преимущества реализации, которая одновременно поддерживает большое количество гипотез, ведь довольно вероятна ситуация, при которой в силу шумов самой разной природы наиболее вероятной станет гипотеза, довольно сильно отличающаяся от реального состояния системы.

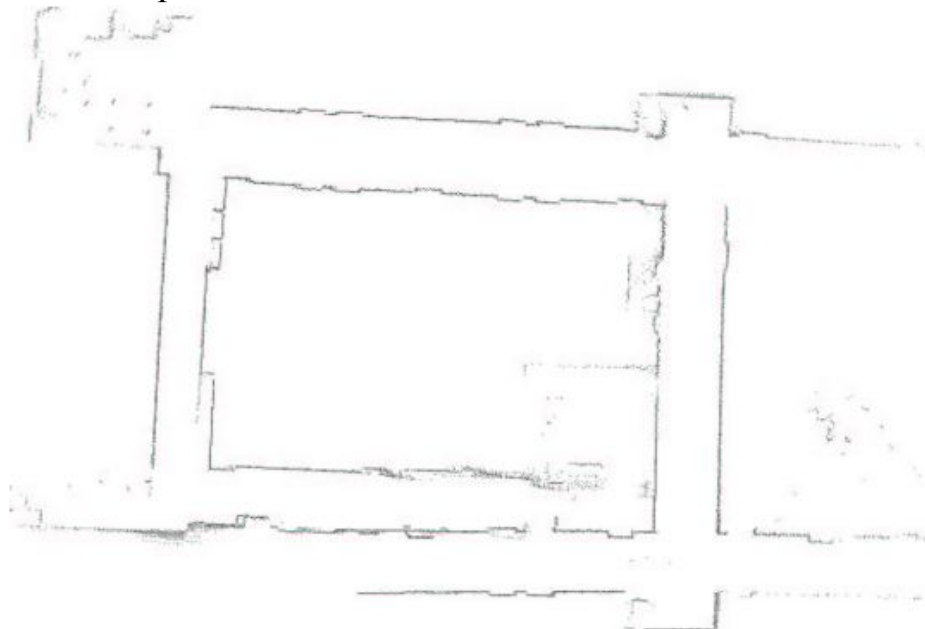


Рис. 4. Результат работы SLAM с выбором самого вероятного кандидата

2.Реализация

Алгоритм

Один шаг алгоритма, решающего задачу SLAM, состоит из нескольких подзадач. Первая состоит в вычислении точного положения исполнителя в пространстве, используя информацию, поступающую на датчики. Для этого необходимо выделить на сцене некоторые ориентиры (landmarks, на рисунке 5 обозначены звёздочками) и определить их описания, необходимые для установления соответствия. На основе своего расположения относительно этих ориентиров робот и корректирует показания о своём положении в пространстве.

После чего он может уточнить карту, на основе новой информации об окружающей сцене.⁵

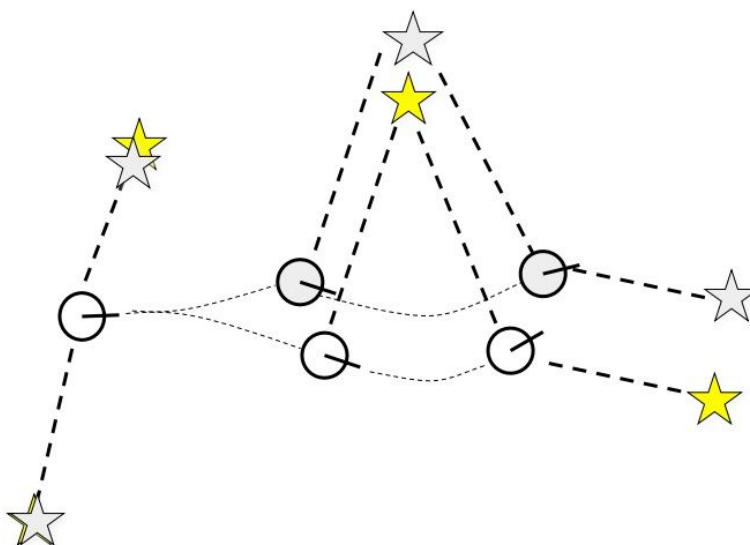


Рис. 5. Схема работы алгоритма SLAM

Выбор средства измерений

Большинство реализаций SLAM используют дорогостоящие лазерные дальномеры (лидары), такие как Hokuyo URG-04LX-UG01 Scanning Laser Rangefinder⁶, цена которых достигает 1115\$.

В данной работе было решено использовать недорогой инфракрасный дальномер Sharp GP2Y0A21YK0F, который включён в стандартную поставку Кибернетического конструктора ТРИК.

⁵ SLAM for Dummies,

http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslamb_las_repo.pdf

⁶ https://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx_ug01.html

Модификации алгоритма

Вследствие выбора датчиков пришлось отказаться от модели DP-SLAM. Из-за довольно высокого уровня шума и погрешности показания датчика получаются неточными и большое число частиц, соответствующих положению робота, “склеивается”, то есть становится неразличимо для алгоритма. Поэтому в поставленных условиях алгоритм работает довольно плохо. Эти проблемы и пути их решения подробно описаны в лекции Cyrill Stachniss “Grid-based FastSLAM”⁷.

Таким образом, необходимо найти подход к решению поставленной задачи, который будет не так чувствителен к погрешностям измерений, как DP-SLAM. Отлично подойдет Grid Maps⁸. Grid Maps – подход, описывающий средства хранения и обновления информации о сцене, окружающей робота [6]. Эта методика имеет ряд полезных свойств, благодаря которым она отлично подходит для решения поставленной задачи.

1. Сцена представляется как набор клеток, каждая из которых соответствует небольшому участку пространства. Это и обеспечивает относительную невосприимчивость алгоритма к небольшим погрешностям замеров.
2. Каждая клетка либо полностью занята, либо полностью свободна.
3. Реализация не очень затратна по памяти.
4. Не важно, откуда поступает информация об окружающем мире и каким образом она была получена, что обеспечивает модульность системы.

Также стоит отметить, что на ARM926 отсутствует модуль для операций с плавающей запятой. Но подход Grid-Maps позволяет реализовать алгоритм, не использующий операции деления нецелых чисел. Для этого необходимо предварительно подсчитать значения тригонометрических функций для нужных углов ($\pi \cdot k / 200$, $k = 0..400$).

⁷ C. Stachniss. Grid-based FastSLAM

<http://ais.informatik.uni-freiburg.de/teaching/ws12/mapping/pdf/slam12-gridfastslam-4.pdf>

⁸ Grid-based SLAM, <http://ais.informatik.uni-freiburg.de/teaching/ws12/mapping/pdf/slam11-gridmaps.pdf>

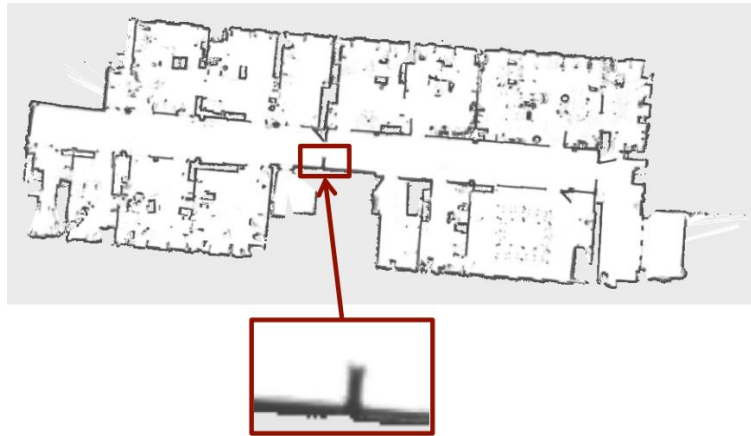


Рис. 6. Пример карты, построенной с использованием Grid Maps

На рисунке 6 изображена карта, построенная с использованием Grid Maps, на увеличенном фрагменте видно небольшое размытие краёв стен, которое появилось вследствие дискретизация пространства.

На рисунке 7 показана разбитая на клетки сцена, и представление этой сцены в памяти. В каждой ячейке матрицы хранится вся необходимая информация об участке пространства, соответствующего этой клетке.

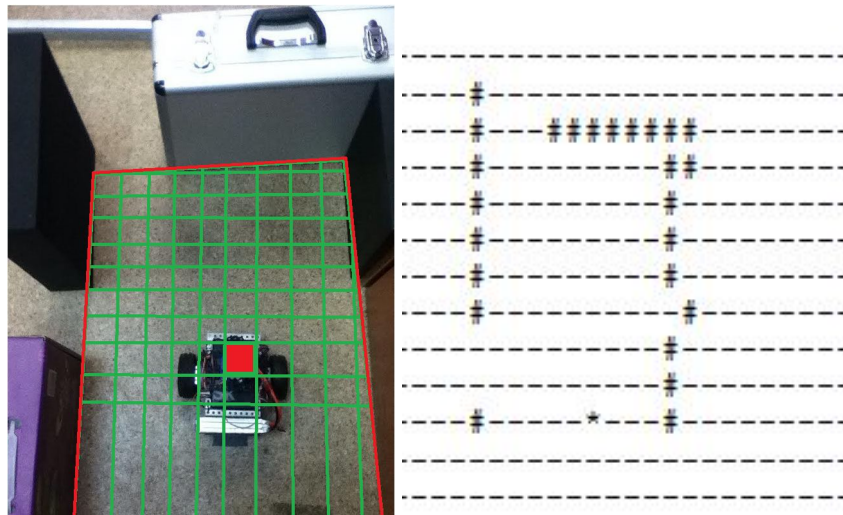


Рис. 7. Сцена и её представление в памяти

Следующей задачей алгоритма является уточнение положения исполнителя в пространстве. Наиболее простой реализацией является перебор тройки $\{x, y, \theta\}$, характеризующей положения робота и выбор наиболее подходящей гипотезы на основе значения функции, численно оценивающей, насколько хорошо текущие показания сенсора могут быть встроены в карту с гипотетического положения исполнителя.

Перебор может быть заменён на метод Монте-Карло⁹. Новый алгоритм меняет положение робота за счёт прибавления случайных величин к каждой компоненте состояния робота $\{x, y, \theta\}$, затем применяется описанная выше функция, чтобы выбрать гипотезу, наиболее подходящую для встраивания в карту.

На рисунке 8 показан результат работы алгоритма, реализованного в рамках данной курсовой работы. На рис. 8а показано, как выглядела карта до фазы уточнения положения робота и встраивания в неё текущих показаний датчика. На Рис. 8b изображён снимок пространства, сделанный роботом на данной итерации работы алгоритма. Для этого снимка была подобрана тройка $\{x, y, \theta\}$, обеспечивающая наилучшее встраивание. На рисунке 8с показано, как выглядит карта после осуществления одной итерации работы алгоритма SLAM. Видно, что все шумы, которые были в верхней части карты, исчезли после изменения положения исполнителя.

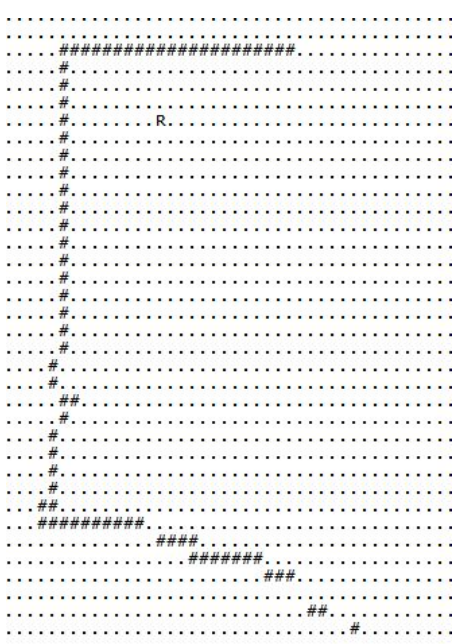
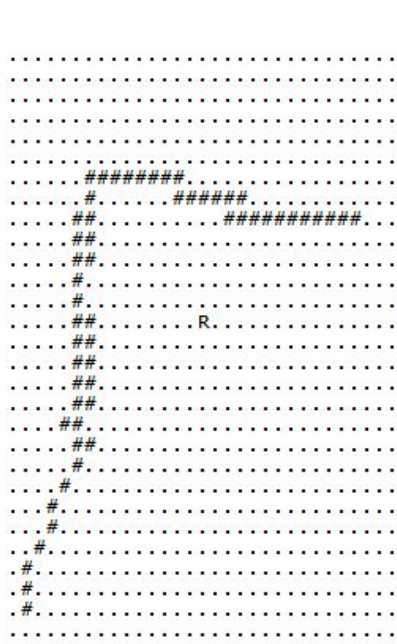
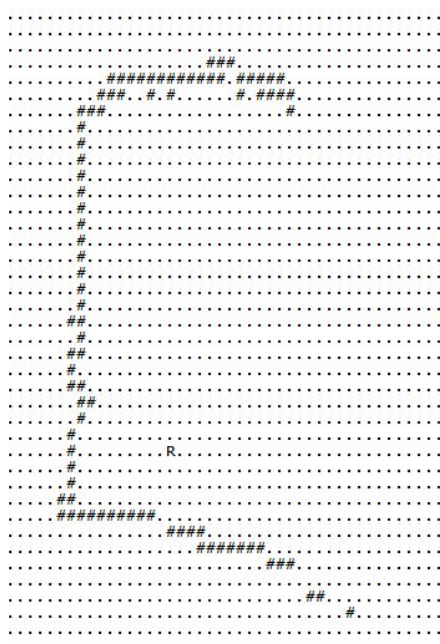


Рис. 8а Карта сцены

Рис. 8б Показания датчика

Рис. 8с Карта после встраивания

Направление дальнейшей работы

В ряде работ отмечается, что метод Монте-Карло может быть заменён на Метод отжига¹⁰. Метод отжига (метод симуляции отжига, simulated annealing) – это техника оптимизации, использующая упорядоченный случайный поиск.

В своей работе “Simulated Annealing: Practice versus theory” [7] Л. Ингбер пишет, что модификации метода отжига являются одним из самых эффективных методов поиска решений для широкого класса оптимизационных задач. Одним из

⁹ Метод Монте-Карло и его точность, <https://habrahabr.ru/post/274975/>

¹⁰ Метод отжига, <http://www.math.spbu.ru/user/gran/sb1/lopatin.pdf>

главных преимуществ этой методики является то, что в условиях нехватки вычислительных ресурсов метод отжига и его модификации, выдает решение, очень близкое к одному из локальных экстремумов.

В одной из более ранних работ “Genetic Algorithms and Very Fast Simulated Reannealing: A Comparison” [8] Л. Ингбер сравнил адаптивный метода отжига (Adaptive Simulated Annealing) с рядом генетических алгоритмов и показал, что на довольно большом классе задач метод отжига не уступает генетическим алгоритмам.

Переход от “метода Монте-Карло” к “методу Отжига” не только может ускорить работу алгоритма, но и поможет избежать “застревания” в локальных максимумах оптимизируемой функции, и продолжить поиск глобального максимума.

Заключение

В рамках данной курсовой работы были получены следующие результаты:

- изучены различные реализации и методики решения задачи одновременной навигации и составления карты;
- выбран набор технологий, подходящих для переноса на платформу ТРИК;
- предложен ряд оптимизаций, как алгоритмических, так и на платформо-зависимом уровне;
- реализован рабочий прототип и продемонстрирована состоятельность выбранного подхода.

Литература

- [1] Заметки о робототехнике. URL: <http://robotics.osll.ru/2014/12/slam.html>
(дата обращения: 23.05.2016)
- [2] *Mur-Artal Raul, Montiel J. M. M., Tardos Juan D.*
ORB-SLAM: a Versatile and Accurate Monocular SLAM System.
// IEEE Transactions on Robotics. 2015. Т. 31. № 5. pp. 1147 - 1163.
- [3] Простыми словами о фильтре частиц. URL: <https://habrahabr.ru/post/276801>
(дата обращения 23.05.2016)
- [4] *A. Eliazar, R. Parr.* DP-SLAM: Fast, Robust Simultaneous Localization and Mapping Without Predetermined Landmarks. // IJCAI'03. 2003. pp. 1135-1142.
- [5] *Kai M. Wurm, C. Stachniss G. Grisetti.* Bridging the Gap Between Feature and Grid-based SLAM. // Robotics and Autonomous Systems. 2010 Т. 58. № 2, pp 140–148
- [6] *C. Stachniss.* Grid-based FastSLAM
<http://ais.informatik.uni-freiburg.de/teaching/ws12/mapping/pdf/slam12-gridfastslam-4.pdf>
- [7] Ingber L. Simulated Annealing: Practice versus theory // Mathematical and Computer Modelling. 18(11). 1993. P. 29-57.
- [8] Ingber L., Rosen B. Genetic Algorithms and Very Fast Simulated Reannealing: A Comparison // Mathematical and Computer Modelling. 16(11). 1992. P. 87-100.