

В. Е. Павловский, д-р физ.-мат. наук, проф., гл. науч. сотр., vlpavl@mail.ru,
Институт прикладной математики им. М. В. Келдыша РАН, Москва,
В. В. Павловский, ст. преподаватель, vlpavl2000@mail.ru,
Российский экономический университет имени Г. В. Плеханова, Москва

Технологии SLAM для подвижных роботов: состояние и перспективы¹

Обсуждается технология SLAM одновременных локализации робота в среде передвижения и картирования окружающего пространства, которая стала одной из основных при решении задачи навигации мобильных роботов. Дан краткий обзор популярных эффективных методов реализации этой технологии, основанных на различных алгоритмах и использующих разную аппаратную базу.

Ключевые слова: навигация, локализация, картирование, робот, SLAM

Введение

Одной из основных задач систем управления (СУ) автономных роботов является задача навигации и ее подзадача — локализация робота в пространстве.

В автоматизированной навигации подвижных агентов (роботов) технология одновременной локализации и картирования (SLAM, Simultaneous Localization And Mapping) — это вычислительная схема построения или обновления карты неизвестной окружающей среды с одновременным отслеживанием местоположения агента в ее пределах. Эта технология стала основной в навигации мобильных агентов (роботов) [1–3]. К настоящему времени уже существует большой ряд алгоритмов реализации технологии SLAM с известным, по крайней мере приближенным, решением. Популярные приближенные методы решения включают специальную фильтрацию — расширенный фильтр Калмана или фильтр частиц — и статистическое оценивание данных. Эти алгоритмы SLAM существенно ориентированы на имеющиеся аппаратные ресурсы, особенно в бортовых вариантах, следовательно, они нацелены на эксплуатационное соблюдение ограничений этих ресурсов и соответствующую вычислительную эффективность.

Существующие подходы используются в автоматизированных автомобилях, беспилотных воздушных транспортных средствах, автономных подводных транспортных средствах, марсоходах, недавно появившихся роботах типа сервисных для работ внутри помещений. В обзоре будем использовать работы [1–15].

¹ Работа выполнена при существенной поддержке гранта РФФИ 15-07-07483-а.

Постановка проблемы

Рассматривается следующая задача SLAM: на основе ряда наблюдений (измерений), полученных от датчика o_t в дискретные моменты времени t , требуется вычислить оценку местоположения агента (робота) x_t и построить карту окружающей его среды m_t . Все данные считаются вероятностными, что является ключевым положением всех рассматриваемых алгоритмов SLAM.

Таким образом, цель состоит в том, чтобы вычислить условную вероятность карты m_t и положения агента x_t при условии имеющегося измерения от датчика:

$$P(m_t, x_t | o_{1:t}). \quad (1)$$

Применение правила Байеса дает схему для того, чтобы последовательно обновить последующее значение местоположения, учитывая карту и функцию перехода $P(x_t | x_{t-1})$:

$$P(x_t | o_{1:t}, m_t) = \\ = P(o_t | x_t, m_t) \sum_{x_{t-1}} P(x_t | x_{t-1}) P(x_{t-1} | m_t, o_{1:t-1}) / Z. \quad (2)$$

Здесь x_t — положение робота в текущий момент времени t ; x_{t-1} — положение в предыдущий момент времени; m_t — построенная карта; $o_{1:t}$ — массив показаний датчиков в течение всего движения от начального до текущего момента. В знаменателе стоит соответствующий нормирующий множитель. Таким образом, формула описывает условное распределение для оценки текущего положения робота на основании построенной карты, показаний датчиков и оценки в предыдущий момент. Следующая формула аналогично описывает пересчет условного распределения для карты:

$$P(m_t | x_t, o_{1:t}) = \\ = \sum_{m_{t-1}} P(m_t | x_t, m_{t-1}, o_t) P(m_{t-1} | o_{1:t-1}, x_t). \quad (3)$$

Отметим, что как и многие проблемы статистического оценивания, совместное определение этих двух переменных может быть найдено на основе некоторого оптимального решения, основанного на алгоритмах типа максимального правдоподобия.

Здесь и далее будем положение агента (аппарата) называть его позой, это принятый в теории SLAM термин.

Особенности алгоритмов технологии SLAM

Согласно приведенной выше схеме алгоритмы, реализующие технологию SLAM, используют различные вычислительные схемы для решения задачи локализации и картирования на основе данных, полученных от различных датчиков.

Базовые (первоначальные) алгоритмы SLAM основаны на уравнениях (1)–(3), включают фильтры Калмана, фильтры частиц (иначе, методы Монте-Карло) и оценки, использующие различные интервальные вычисления. Они обеспечивают оценку указанной выше функции вероятности для позы робота и для параметров карты.

Популярным методом в технологии SLAM является уравнивание, основанное на использовании изображений от видеосенсоров. Уравнивание проводится для устранения невязок, обусловленных наличием ошибок в избыточно измеренных величинах, и для определения наиболее вероятных значений искомым неизвестных или их значений, близких к этим вероятнейшим. В процессе уравнивания это достигается путем определения поправок к измеренным величинам (углам, направлениям, длинам линий или превышениям). Нередко уравнивание выполняется методом наименьших квадратов для минимизации невязок.

Уравнивание в технологии SLAM позволяет совместно оценивать позы и характерные положения ориентиров, увеличивая точность карты, и используется во многих новых коммерциализированных системах, включающих технологию SLAM, таких как например, проект Google Танго [9] (проект специального гаджета — планшета с 3D-информацией об окружающем пространстве). Здесь используется ряд изображений (видеокадров) серии 3D-пунктов с различных точек наблюдения, а с помощью уравнения решается задача одновременного уточнения 3D-координат, описывающих геометрию сцены, параметров относительного движения и оптических особенностей камеры, используемой для получения изображения. Эти вычисления выполняются согласно специализированному критерию оптимальности, включающему соответствующие параметры проектирования изображения всех наблюдаемых пунктов.

Заметим, что уравнивание почти всегда используется в качестве последнего шага каждого основанного на 3D-особенностях среды алгоритма реконструкции сцены. Уравнивание было первоначально применено в области фотограмметрии в

1950-х гг. и все более часто используется исследователями компьютерного видения в течение последних лет.

Новые алгоритмы для технологии SLAM остаются до сих пор активной областью исследований, которые ведутся согласно различным требованиям и предположениям о типах карт, датчиков и моделей обработки. Многие из современных систем, реализующих технологию SLAM, основаны на комбинации различных подходов.

Сенсоры. Измерения в технологии SLAM

В технологии SLAM часто используются несколько различных типов датчиков, так как только в этом случае возможно последовательное уточнение данных о позе робота и о карте. Возможности и пределы (физические ограничения) различных типов датчиков были основной причиной разработки новых алгоритмов. Статистическая независимость измерений — обязательное требование, чтобы справиться с метрическими отклонениями и с шумом в измерениях.

Различным типам датчиков соответствуют различные алгоритмы SLAM.

Лазерные измерители. При использовании в качестве основных сенсоров лазерных измерителей получаемые визуальные данные предоставляют подробную информацию о многих объектах (измеряемых пунктах, точках) в области, окружающей робот, создавая своеобразные облака измерений. В этом случае иногда может быть сделан вывод о том, что не требуется применение технологии SLAM, поскольку формы в этих облаках данных могут быть легко и однозначно выровнены (методом уравнивания) в каждом шаге через соответствующую регистрацию изображений.

Сонары. Хорошими датчиками измерения дальностей могут служить сонары. Они могут работать в условиях пониженной видимости, например, в тумане, что в таких случаях делает их предпочтительными.

Тактильные датчики. Тактильные (осязательные) датчики используются чрезвычайно редко, поскольку они позволяют получить только информацию об объектах, расположенных очень близко к агенту-роботу. Поэтому необходима их интеграция с другими сенсорными системами.

Осязательные датчики и радиодатчики. Некоторые технологии SLAM предусматривают использование осязательных датчиков (измерение выполняется только локальным прикосновением), радаров и технологии Wi-Fi.

В современных технологиях SLAM применяются для ориентирования также квазиоптические радиодатчики, реализуемые в трилатеральном (определение положения по трем базовым станциям (БС)) или в мультитлатеральном вариантах с множественными БС (в системах RTLS (Real-time Locating

Systems), работающих, например, по радиометкам), или как мультиугломерная конструкция.

Оптические датчики могут быть одномерными (с единственным лучом) или двумерными (широкий луч, или двойной луч). В качестве оптических датчиков могут использоваться лазерные дальнометры, 3D-сенсоры высокого разрешения (например LiDAR), 3D-оптические локаторы. Эти датчики могут также включать специальные сенсоры — вспышки, одну или несколько 2D-камер. В алгоритмах, специализированных для работы под водой, в качестве датчиков могут использоваться 2D- или 3D-гидролокационные датчики.

С 2005 г. было выполнено интенсивное исследование технологии vSLAM (визуальной технологии SLAM), использующей, прежде всего, визуальные датчики (камеры) из-за увеличивающейся интенсивности применения таких датчиков, например камер в мобильных устройствах. Визуальные датчики и датчики типа оптических локаторов достаточно информативны, чтобы допускать существенное (значашее) получение измерений во многих случаях.

Ориентиры. Информация, получаемая от датчиков, может быть основана на ориентирах и на прямой обработке исходных данных. Ориентиры — уникально идентифицируемые объекты в мире, местоположение которых может быть оценено датчиком (например, точки доступа Wi-Fi или радиомаяки). Часто при прямой обработке исходных данных не делается предположение о том, что ориентиры могут быть точно определены, и вместо этого в них вероятностная модель $P(o_i|x_t)$ (вероятность данного измерения при данном местоположении объекта) рассматривается непосредственно как функция местоположения.

Носимый SLAM. Применяют также технологию SLAM для людей-пешеходов (операторов), в которой используется специальная обувь с установленными инерциальными датчиками в предположении, что пешеходы в состоянии избежать стен и двигаться вдоль них. Такая система автоматически строит план стен и может построить планы здания для дальнейшей обработки внутренней системой позиционирования. Оператору нужно в этом случае обойти помещения, и система создаст их карту.

SLAM во внешней среде. Для некоторых применений в естественной среде (так называемых outdoor-приложений) необходимость в технологии SLAM была почти полностью устранена из-за высоких точностей датчиков GPS, особенно в их дифференциальных версиях с повышенной точностью. С точки зрения технологии SLAM такие датчики могут быть рассмотрены как датчики местоположения, полностью доминирующие над прочими измерителями. Однако датчики GPS могут снизить качество работы полностью или в определенных случаях, например, в случаях плохой видимости спутников или технических проблем, такие примеры могут быть особенно интересны для некоторых приложений робототехники (движение в лесу, горах).

Ресурсы

- а) На странице в Интернет <http://openslam.org/> собран богатый набор различных методов SLAM. Там же имеются примеры реализаций различных алгоритмов, и они находятся в открытом доступе.
- б) В известной системе для роботов ROS (*Robot Operating System*) имеется пакет, поддерживающий реализацию алгоритма Mono-SLAM (SLAM, работающий с одной монокулярной камерой) [7, 8].
- в) В MATLAB имеются реализации нескольких версий SLAM — модели EKF-SLAM, FastSLAM 1.0, FastSLAM 2.0 и UKF-SLAM (метод, позволяющий избавиться от процедуры линеаризации в нелинейном стохастическом оценивании). Они реализованы как пакеты MATLAB-файлов и готовы к применению.

Алгоритмы SLAM

В общем случае алгоритм SLAM можно описать [14] как повторяющуюся последовательность шагов:

- 1) сканирование окружающего пространства,
- 2) определение смещения на основе сравнения текущего кадра с предыдущим,
- 3) выделение на текущем кадре особенностей-меток,
- 4) сопоставление меток текущего кадра с метками, полученными за всю историю наблюдений,
 - 5.1) обновление информации о положении робота за всю историю наблюдений,
 - 5.2) проверка на петли — не проходит ли робот повторно по одной и той же части местности,
 - 5.3) выравнивание общей карты мира (отталкиваясь от положения меток и робота за всю историю наблюдений).

На каждом шаге алгоритма робот располагает предположениями о структуре мира (у него есть текущая карта) и историей движения в нем (траектория движения относительно карты с разбивкой по времени).

Опишем кратко наиболее распространенные методы SLAM.

Базисный SLAM

Базисным SLAM будем называть первоначальные версии и схемы алгоритмов SLAM. Популярным методом для решения задачи SLAM был расширенный фильтр Калмана (*EKF, Extended Kalman Filter*). На каждом шаге у робота есть набор ранее полученных ориентиров (landmark) и поступившие данные (дальномерные и фото, пусть RGB). На основании новых и предыдущих кадров можно определить смещение робота (используя методы визуальной одометрии) и предсказать новую позицию робота. В месте с тем, из нового кадра можно выделить местоположение особенностей (ориентиров) и вычислить положение робота относительно них. На основе разницы между двумя этими оценками позиций робота обновляются вероятности/веса

для всех ориентиров и корректируются позы—траектория движения робота. В качестве структуры для хранения информации о мире и траектории движения робота используется разрастающаяся со временем матрица ковариации, содержащая на каждом шаге исчерпывающую информацию о текущем представлении мира. Заметим, что в случае большого числа поз лучше подойдет фильтр частиц. Типичная схема SLAM приведена на рис. 1 [12]. Здесь показаны реальная траектория и реальные положения ориентиров вокруг робота, и те же характеристики, полученные как оценки по доступ-

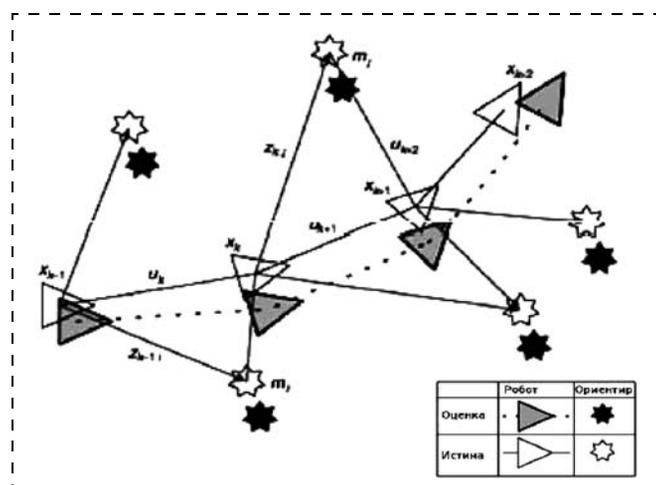


Рис. 1. Схема концепции SLAM



Рис. 2. Персональный робот ER-1 (Evolution Robotics, USA)

ным измерениям сенсоров робота. Цель алгоритмов навигации, очевидно, состоит в том, чтобы эти данные не сильно расходились.

Аппаратные средства робота достаточно важны. В качестве мобильных роботов будем рассматривать колесные роботы для применений внутри помещений. Далее будем главным образом рассматривать программное обеспечение SLAM и не будем рассматривать роботы со сложными моделями движения, такие как гуманоидные роботы, автономные подводные транспортные средства, автономные самолеты, роботы со нестандартными конфигурациями колеса и т. д.

Робот. Важным параметром качества работы робота является такой параметр, как оценка своего собственного положения только от одометрических датчиков, измеряющих вращения колес. У робота не должно быть ошибки больше, чем 2 см за 1 м перемещения и 2° за поворот робота на 45° . Типичные датчики положения робота должны сообщать роботу его координаты (x, y) в некоторой декартовой системе координат и также давать информацию о текущем смещении робота.

Для примера можно указать известную и давно выпускаемую конструкцию — мобильный робот ER-1 фирмы Evolution Robotics, USA. Робот поставляется с камерой и пресобранной системой управления нижнего уровня (рис. 2).

Рис. 2 дает определенное представление о типе робота, на который рассчитан базисный SLAM.

Устройство измерения дальностей (дальномеры). Используемые в настоящее время устройства измерения дальностей являются обычно лазерными сканерами. Они очень точны, эффективны и не требуют сложных вычислений при обработке измерений. Однако, надо отметить, они весьма дороги, цены могут превышать несколько тысяч USD. Есть с лазерными сканерами некоторые проблемы — они могут давать большие ошибки при попытке измерений от определенных поверхностей включая стекло, где они могут дать очень плохие данные. Также лазерные сканеры не могут использоваться под водой, так как вода препятствует распространению света, и измеряемый диапазон существенно уменьшается по сравнению с измерениями на воздухе.

Некоторое время назад весьма широко применялись такие сонары (ультразвуковые датчики). Они очень дешевы по сравнению с лазерными сканерами. Однако их измерения не очень хороши по сравнению с лазерными сканерами, и они часто дают некачественные данные. Если лазерные сканеры дают узкие лучи измерения, не превышающие обычно $0,25^\circ$, у сонара могут легко получаться лучи шириной до 30° .

Следует отметить также использование систем видения. Традиционно такая реализация сенсорной системы была весьма сложна в вычислительном плане и также была существенно подвержена ошибкам из-за изменений в освещенности (отметим здесь такой крайний случай, как комнату без

света, в ней система видения вообще работать не будет). В последние годы, тем не менее, в этой области были некоторые интересные достижения, например, инфракрасное зрение. Часто системы видения используют стереосистемы, чтобы измерить расстояние. Использование такого видения в последние годы более типично, нежели применение лазера или гидролокатора, что упрощает конструирование алгоритмов обработки. Также следует заметить, что картина, получаемая от системы видения, содержит существенно больший объем данных по сравнению со сканерами или локаторами. Некоторое время назад это было проблемой вычислительной обработки, но сейчас эта проблема снята в связи с резко возрастающей вычислительной мощностью бортовых процессоров. Например, у ER-1 имеется система видения, она будет использоваться в SLAM.

Данные одометрии. Важный аспект рассматриваемого алгоритма SLAM — данные одометрии. Цель получения одометрических данных — найти приблизительное положение робота, как оно измерено (и пересчитано) датчиками вращения колес робота. Эти данные служат начальным условием того, где робот мог бы находиться, для дальнейшего использования процедур EKF.

Сложная проблема в работе с данными одометрии и лазерными данными состоит в строгом контроле времени измерений и их синхронизации. Лазерные данные в некоторое время устаревают, если одометрические данные получены существенно позже. Чтобы удостовериться, что они действительны, можно экстраполировать данные.

Ориентиры. Ориентиры — особенности в среде, которые можно легко повторно наблюдать и отличить от окружающей среды.

Они используются роботом, чтобы узнать, где он находится (чтобы локализовать себя). Существенно, что ориентиры распознаются либо дальномерами, либо системой видения, либо датчиками прикосновения.

Можно сказать, что тип ориентиров робота и их использование будет зависеть от окружающей среды, в которой работает робот. Ориентиры должны быть заметными с различных точек наблюдения, чтобы позволять рассмотреть их (обнаружить) из различных положений и с различных углов наблюдения. Кроме того, ориентиры должны быть достаточно уникальными, чтобы они могли быть легко определены при переходе от одного временного шага к другому. Другими словами, если робот повторно наблюдает два ориентира, должна быть обеспечена возможность легко определить, какой из ориентиров был виден ранее. Если два ориентира находятся очень близко друг к другу, это может привести к ошибке и, тем самым, вызвать проблему.

Разработчик системы навигации должен выбрать ориентиры, постоянные на участке движения робота или, по крайней мере, изменяющиеся весьма редко. Наконец, есть очевидное условие, что время

поиска и обработки ориентиров роботом не должно быть слишком большим, иначе это сделает общую задачу робота плохо выполнимой.

Итак, ключевые условия о подходящих ориентирах следующие:

- ориентиры должны быть заметными с разных точек наблюдения;
- отдельные ориентиры должны быть отличимыми друг от друга;
- ориентиры должны быть достаточно многочисленными в окружающей среде;
- ориентиры должны быть постоянными.

Извлечение (детектирование) ориентиров. Как только определено, какие ориентиры робот должен использовать, необходимо определить надежный способ извлечения данных о них из сенсорных измерений. Как упомянуто выше, это зависит в основном от того, какие типы ориентиров назначены роботу, а также какие датчики используются в SLAM-системе.

Для лазерного сканирующего дальномера (и других дальномеров) хорошо работает способ определения скачков дальностей в измерениях на местности и наблюдения "поведения" нормалей к поверхности, построенных по триангуляционной сетке, в свою очередь, построенной по измеренным дальномером точкам поверхности. Здесь совместное наблюдение нормалей и скачков дальностей позволяет надежно выделять крупные препятствия на поверхности передвижения робота, которые могут служить ориентирами. Данный подход является разработкой авторов настоящей статьи, он проверен в серии экспериментов, моделирующих процесс опознавания ориентиров.

Схема алгоритма *базисного SLAM* для рассмотренного аппарата показана на рис. 3. Блоки этой схемы соответствуют описанным выше этапам вычислений с добавленными к ним вычислениям оценок по фильтрам Калмана или расширенного EKF.

Общая вычислительная схема фильтра Калмана (одно из представлений) может быть традиционно описана формулами (следуя, например, [15]):

$$\begin{aligned}
 q_{i+1}^* &= \hat{q}_{i+1} + K_{i+1}[R_{i(i+1)} - R_{0(i+1)}]; \\
 K_{q(i+1)} &= [I - K_{i+1}H_{i+1}]K_{q0(i+1)} \times \\
 &\times [I - K_{i+1}H_{i+1}]^T + K_{i+1}W_{i+1}K_{i+1}^T; \\
 K_{i+1} &= K_{q0(i+1)}H_{i+1}^T \times \\
 &\times [H_{i+1}K_{q0(i+1)}H_{i+1}^T + W_{i+1}]^{-1}; \\
 K_{q0(i+1)} &= \Phi_i K_{qi} \Phi_i^T + V_i,
 \end{aligned} \tag{4}$$

где q_{i+1}^* — апостериорный вектор состояния объекта-агента размерностью 6×1 , здесь полагается, что определению подлежат три проекции радиус-вектора агента в некоторой (пространственной) системе координат и три проекции его скорости, при плоском движении размерность векторов и матриц



Рис. 3. Схема базисного SLAM

может быть уменьшена; \hat{q}_{i+1} — априорный вектор состояния размерностью 6×1 ; K_{i+1} — матрица (коэффициентов) усиления фильтра размерностью $6 \times n$; $R_{i(i+1)}$ — "истинный" вектор измерений размерностью $n \times 1$; $R_{0(i+1)}$ — навигационный (расчетный) вектор измерений размерностью $n \times 1$; $K_{q(i+1)}$ — априорная матрица ковариаций среднеквадратических отклонений, характеризующих точность оценки вектора состояния, размерностью 6×6 ; H_{i+1} — матрица наблюдаемости размерностью $n \times 6$; I — единичная матрица размерностью 6×6 ; W_{i+1} — матрица ошибок измерения размерностью $n \times n$; $K_{q0(i+1)}$ — апостериорная матрица ковариаций среднеквадратических отклонений, характеризующих точность оценки вектора состояния, размерностью 6×6 ; Φ_i — фундаментальная матрица размерностью 6×6 ; V_i — матрица влияния внешних возмущений на вектор состояния; n — число измерений, вводимых в обработку.

В этой технологии понимается, что матрица наблюдаемости H — это матрица перехода, связывающая измерения и вектор состояния, получаемый от сенсоров. Фундаментальная матрица Φ — это матрица, описывающая движение связанной системы координат объекта (в проекциях на оси инерциальной) относительно инерциальной в заданном интервале времени.

Формула (4) является ядром системы вычислений и показывает вычислительную сложность SLAM. Заметим также, в этой процедуре имеется нетривиальная функция обращения матрицы, т. е. вычисления обратной матрицы.

MonoSLAM

Это алгоритм (алгоритмы), использующий данные от одной монокулярной камеры на роботе.

В этом состоит отличие данного метода от методов, использующих лазерные сканеры. Вычислительная схема весьма похожа на описанную в предыдущем разделе. Подробно эта система введена в работе [7].

Как в классических подходах SLAM, основанных на лазерных сканерах, предполагается, что поза робота описана как стохастическая переменная с гауссовским распределением, и карта окружающей среды не слишком насыщена. Окружающая среда описана ограниченным набором геометрических особенностей, т. е. имеет геометрически измеримые характеристики. Особенности также описаны как гауссовские переменные. Метод строит вероятностную 3D-карту среды робота и оценивает положение робота в ней.

Системное состояние, определенное роботом, определяет позу робота и наносится на карту, представленную в любое время $t = kt$, где t — время, начиная с предыдущего шага, оно также задается как стохастическая переменная с гауссовским распределением.

Информация относительно движения камеры в любой момент закодирована в векторе x_k , построенном как траектория робота.

Все статистические вычисления также проводятся как статистические фильтрации, могут использоваться различные фильтры.

В заключительной части работы [7] описано применение метода для андроидного робота. Таким роботом был выбран HRP-2. Авторы утверждают, что это один из самых совершенных роботов к моменту проведения экспериментов.

Робот оснащен тринокулярной видеокамерой, смотрящей вперед, позволяющей получать точные трехмерные измерения пространства перед роботом. Эта камера служила видеосенсором алгоритма SLAM. Помимо этого робот имеет и другие сенсоры, среди которых авторы выделяют трехосевой гироскоп, установленный в груди аппарата, позволяющий измерять угловые скорости тела на частоте до 200 Гц. В экспериментах этот датчик опрашивался на частоте 30 Гц, расширяя возможности стандартного monoSLAM.

Задача движения была поставлена следующим образом. Робот должен был двигаться в достаточно загроможденной лабораторной среде по окружности, причем из соображений безопасности траектория была разделена на пять частей, между которыми робот делал короткие остановки. Система SLAM работала все время не прерываясь. Рассмотренный в работе [7] алгоритм monoSLAM подтвердил свою достаточно эффективную работоспособность.

FastSLAM

Имеется эффективный алгоритм SLAM под названием FastSLAM [5]. Использование этого фильтра вычислительно более эффективно, чем использование классического фильтра Калмана.

В основе этого метода лежит идея разбиения фильтра большой размерности на множество фильтров малой размерности, что позволяет уменьшить вычислительную сложность задачи и соответственно ускорить вычисления. Рассмотрим для простоты движение робота на плоскости; тогда поза робота описывается тремя параметрами (координаты x и y и направление оси робота), а все ориентиры — двумя координатами x и y . Таким образом, если карта содержит K ориентиров, общее число параметров, требующих определения в задаче SLAM, составляет $2K + 3$.

Таким образом, в случае большого K размерность задачи возрастает пропорционально, что затрудняет применение традиционных методов. Фильтр Калмана требует хранения ковариационной матрицы всех параметров, которая содержит уже $(2K + 3)^2$ элементов, и полного пересчета ее при каждом измерении. Фильтры частиц (Particle Filter) также эффективны в случае небольших размерностей, но могут испытывать сложности с отслеживанием сложных многомерных областей, если не увеличивать число частиц. Практически непосредственная реализация этих методов позволяет эффективно обрабатывать карты с несколькими сотнями ориентиров, но тысячи ориентиров уже не позволяют обработку в реальном времени.

Метод FastSLAM основан на том, что отдельные измерения положений разных ориентиров независимы друг от друга (в строгом смысле вероятностной независимости случайных погрешностей этих измерений). Если бы мы точно знали положение робота, оценки координат каждого из K ориентиров можно было бы делать только на основании наблюдений за ним — наблюдения за другими ориентирами на них не влияют. Таким образом, $2K$ -мерная задача оценки карты распадается на K независимых двумерных задач оценки положений отдельных ориентиров, и при каждом измерении нужно обновлять данные только по тем ориентирам, которые наблюдались в этот раз.

В задаче SLAM, однако, положение робота также неизвестно и определяется по тем же наблюдениям за ориентирами. В этом случае между измерениями разных ориентиров возникает вероятностная зависимость через параметры позы робота — погрешности измерений одних ориентиров вносят погрешности в определяемую позу и через них влияют на погрешности измерения других ориентиров. Однако, если мы рассмотрим *условные распределения* оценок положений ориентиров относительно заданного распределения измеряемой позы робота, эти оценки уже будут *условно независимы*. Таким образом, общее апостериорное распределение $2K + 3$ параметров, измеряемых в методе SLAM, можно следующим образом разбить в трехмерное распределение параметров позы и K двумерных распределений координат ориентиров, условных

по распределению позы, но независимых при этом условии друг с другом:

$$p(s^t, \theta|z^t, u^t, n^t) = p(s^t|z^t, u^t, n^t) \prod_{k=1}^K p(\theta_k|s^t, z^t, u^t, n^t), \quad (5)$$

где s — поза робота, u — управление, z — измерения положений ориентиров (показания датчиков), а n — идентификация ориентиров, сопоставление наблюдений ориентирам; об этом будет подробнее сказано ниже. Верхний индекс t обозначает массив значений для всех моментов времени — распределения определяются не только текущими наблюдениями, но и всей прошлой историей. Заметим, однако, что в силу марковской природы рассматриваемых процессов можно хранить только текущую оценку распределений, последовательно обновляя ее на основании наблюдений. Наконец, θ — множество положений ориентиров, которые и измеряются в методе вместе с позой s , они не зависят от времени. Таким образом, формула (5) в явном виде отражает разложение совместного распределения s и θ в произведение безусловного распределения для позы s и условных по позе распределений положений всех ориентиров θ_k по отдельности.

Это разложение — строго верное и применимое в любых ситуациях SLAM — и лежит в основе метода FastSLAM. Практическая предложенная авторами реализация включает модифицированный фильтр частиц с M частицами для распределения позы и привязанными к каждой из этих частиц фильтра K двумерными фильтрами Калмана, измеряющими условные распределения параметров ориентиров. Всего получается KM фильтров, но эффективная организация фильтров Калмана в виде бинарного дерева вместо линейного списка позволяет обновлять всю структуру за $O(M \log k)$ операций для каждого наблюдения. Таким образом, метод становится намного эффективнее непосредственного многомерного фильтра для всех параметров, вычислительная сложность которого имеет порядок K^2 операций для каждого наблюдения.

Проведенные авторами эксперименты (как на вычислительных моделях, так и с физической моделью на полигоне NASA для исследований Mars Rover) показали, что оптимальное число частиц M фильтра составляет порядка 100 — при дальнейшем увеличении M точность измерений уже заметно не улучшается независимо от сложности карты. При этом алгоритм справляется в реальном времени с картами, содержащими тысячи и десятки тысяч ориентиров (моделирование проводилось для карт с 50 000 ориентиров) — величины, лежащие далеко за пределами возможностей обычных многомерных фильтров.

Кроме того, как дополнительное достоинство данного метода следует отметить большую устойчивость к ошибкам идентификации. Под идентификацией в задачах SLAM понимается отождествление ориентиров между измерениями — "понимание", что данные два наблюдения в разные

моменты времени относятся к одному ориентиру. В классическом методе ошибка идентификации может приводить к "катастрофическим последствиям". В случае метода FastSLAM вместо единого фильтра используются M фильтров, привязанных к отдельным частицам, и поэтому есть возможность по-разному обновлять эти блоки фильтров на основании разных гипотез об идентификации ориентиров. В дальнейшем частицы с привязанными к ним картами, основанными на неверной идентификации, потеряют в точности измерений и будут отсеяны фильтром частиц, что позволит методу "выправиться" и восстановить точность после ошибочной идентификации.

vSLAM

В определенном смысле визуальный SLAM обобщает методы работы monoSLAM.

vSLAM использует изображения от камеры, чтобы планировать положение робота в новой окружающей среде. Этот метод работает, отслеживая особенности изображений между кадрами камеры и определяя позу робота и положение особенностей в мире, основанные на их относительном движении. Это фактически обеспечивает дополнительный одометрический источник, который полезен для определения местоположения робота. Поскольку датчики камеры обычно более дешевые, чем лазерные датчики, это может быть полезной альтернативой или дополнением к основным на лазере методам локализации.

В этом разделе решение проблемы SLAM описывается с помощью ее графоориентированной формулировки. В этой структуре позы робота описаны узлами графа. Ребра между этими узлами представляют пространственные связи или ограничения между позами. Они, как правило, строятся из наблюдения датчиками или от одометрии. При такой формулировке решением проблемы SLAM будет такая конфигурация узлов, которая минимизирует ошибки введения ограничений.

Переход к оптимизации ориентирован на ряд поз и ограничений между ними. Одним из ограничений является карта. В графовом представлении каждая поза камеры соответствует одному узлу. Кроме того, для каждого узла можно строить стек наблюдаемых особенностей, а также их 3D-положения относительно узла. Ограничения между узлами вычислены на основе особенностей, связанных с узлами.

Значительное число методов SLAM требует использования точных, а потому дорогих и крупногабаритных сенсоров, таких как лазерные дальнометры, которые не всегда можно разместить на небольшом летающем аппарате. Здесь предлагается простой и надежный алгоритм, работающий с техническим зрением на базе одной или двух недорогих видеокамер и дающий устойчивые результаты даже при низком качестве (в экспериментах использовалось разрешение 320×200) и значительном сма-

зывании изображений. При этом алгоритм ориентирован на летающие роботы, и предполагается, что камеры установлены снизу робота и смотрят вертикально вниз для картографирования ландшафта под роботом. В случае стереозрения при этом возможно построение трехмерного ландшафта с определением также и высоты элементов рельефа; при наличии одной камеры алгоритм строит карту в предположении, что рельеф плоский. Кроме камер, предполагается оснащение робота датчиком наклона аппарата, определяющим углы тангажа и крена. Опыт показывает, что даже недорогие датчики определяют эти углы с достаточно высокой точностью.

Обработка информации с камер состоит в нахождении на них выделяющихся деталей — ориентиров, и анализе смещения этих ориентиров между разными снятыми камерой кадрами. В общем случае положение камеры описывается шестью параметрами (три координаты и три угла), и для определения всех шести требуется сопоставление трех ориентиров. Однако определение двух углов по датчику наклона дает возможность определять остальные параметры только по двум ориентирам. Для этого сначала изображения поворачиваются для коррекции данных углов тангажа и крена и построения виртуального горизонтального изображения. Затем по изменению масштаба — расстояния между ориентирами — определяется изменение высоты робота над рельефом, а по повороту соединяющей их линии — изменение угла рыскания. Наконец, после нормализации кадра коррекцией масштаба и поворота определяется смещение по горизонтальным осям.

Для отождествления ориентиров в разных кадрах используются 64-битные дескрипторы SURF (Speeded Up Robust Features) — это целочисленные характеристики особенностей изображения, инвариантные относительно поворотов и масштабирования; для каждого кадра хранится множество дескрипторов найденных особенностей и их координаты. Для определения смещения достаточно сопоставления двух ориентиров, поэтому применяется следующая процедура. Алгоритм сортирует все возможные сопоставления ориентиров по степени сходства (Евклидову расстоянию в пространстве дескрипторов). Затем берется пара наиболее похожих ориентиров, для них вычисляется соответствующее смещение камеры, после чего вычисляются проекции остальных ориентиров — как они должны были бы выглядеть при таком смещении камеры. Далее вычисляется суммарная оценка соответствия этих вычисленных проекций с ориентирами на реальном изображении (оценивая как близость координат на изображении, так и близость SURF дескрипторов). Если эта оценка превышает заданный порог — алгоритм полагает, что нужная трансформация найдена; иначе считается, что отождествление ориентиров было ложным, и берется следующая пара по степени сходства. Отметим, что отождествляемые пары берутся

именно так — сортировкой общего массива по степени сходства, это эффективнее, чем сравнивать сначала первый ориентир со всеми, потом второй, и так далее. В результате алгоритм останавливается либо после нахождения приемлемого по оценке соответствия, либо по превышении разрешенного времени работы. В последнем случае считается, что соответствие кадров не найдено.

Этот алгоритм применяется в двух ситуациях. Прежде всего, сопоставляются соседние кадры для "визуальной одометрии" — определяется последовательность смещений аппарата и интегрированием ее получают итоговые координаты. Однако такой метод подвержен накоплению ошибок, поэтому в случае возвращения в область, где робот уже был, проводится коррекция для "замыкания петли" — совмещения начала и конца пути.

Алгоритм, дающий возможность таких коррекций, составляет основу рассматриваемого метода vSLAM. Для этого строится граф, вершинами которого являются все когда-либо полученные кадры, а ребрами — ограничения на смещения между этими кадрами, полученные сопоставлением ориентиров. Изначально граф строится как цепь кадров, связанных ограничениями "визуальной одометрии", каждый раз рассчитываются только координаты позы робота для очередного кадра. Однако при этом каждый раз текущие координаты робота сопоставляются с координатами, в которых были сняты прошлые кадры; и если одно из прошлых положений оказывается близким к текущему (в пределах 3σ , где σ — оценка дисперсии координат), этот прошлый кадр также сопоставляется с текущим. Далее, если визуальное сопоставление успешно найдено, соответствующее ребро добавляется к графу, после чего координаты всех вершин пересчитываются так, чтобы минимизировать (в смысле метода наименьших квадратов) суммарные отклонения перемещений между прошлыми позами робота (вершинами графа) от визуально определенных смещений. Этот подход с использованием графов позволяет эффективно корректировать накопленные ошибки, "замыкая петли" траектории, при этом корректно обрабатывая ситуации многократных "возвращений в знакомые места" для сложных траекторий со множеством самопересечений.

Для тестирования этого метода использовались ручные эксперименты: две камеры и датчик наклона были закреплены на длинном шесте, который нес перед собой экспериментатор. С этим снаряжением он обошел вокруг здания, пройдя около 190 м, замкнув в конце траекторию. При этом накопленная ошибка оказалась довольно велика (порядка 20...30 м). Однако после возвращения к началу траектории метод корректно обнаружил замыкание петли и скорректировал построенный граф, исправив ошибки локализации. При этом на территории отсутствовали бросающиеся в глаза детали (часть пути со-

ставляла трава, часть — покрытая плитами дорога), но это не помешало успешно найти соответствие.

Второй эксперимент проводился также вручную внутри здания. На пути были разложены разные предметы, и определялась точность картографирования этих предметов. В результате получилась относительная погрешность менее 8 %, в основном порядка 5 %.

Далее были проведены эксперименты с реальными роботами. Робот — мини-дирижабль был оснащен камерами без датчика наклона (считая, что наклон всегда равен нулю — для дирижабля это близко к истине), после чего он проведен по области эксперимента для картографирования. Затем ему была поставлена задача удержания позиции; экспериментаторы несколько раз отталкивали робота от удерживаемой позиции, но всякий раз он успешно определял свое перемещение и возвращался.

Наконец, был проведен эксперимент с мини-вертолетом. На вертолете стояла одна камера без датчика наклона. Опять же, наклон считался равным нулю — что является уже намного более грубым допущением в случае вертолета. Тем не менее, картографирование области с помощью вертолета определило координаты ключевых ориентиров с ошибкой 20...60 см при общих размерах области 35 м.

Таким образом, эксперименты показывают надежную и устойчивую работу алгоритма в реальных условиях даже при оснащении роботов минимальным набором дешевых сенсоров.

INS SLAM, INS-GPS-SLAM

Как показано выше, в технологии SLAM редко используется единственный сенсор, который связан, главным образом, со счислением пути (dead reckoning, DR, одометрическое счисление) или инерциальной навигационной системой (INS).

Аббревиатура ИНС СЛАМ (в англоязычной литературе INS/SLAM или INS/VSLAM) обозначает систему, объединяющую обработку данных от сенсоров инерциальной навигационной системы (ИНС, INS) и от системы VSLAM.

В работе [6] предлагается именно объединение с инерциальной навигационной системой. В целом построение системы SLAM требует радара, лазерного сканера, датчика видения или любых других

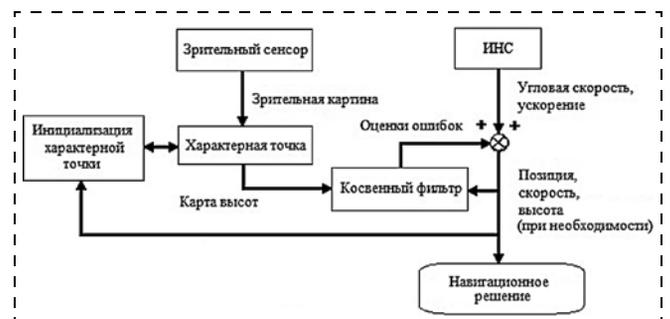


Рис. 4. Общая схема ИНС-SLAM

типов датчиков, это определяет характеристики данных (рис. 4). Предлагаемый подход использует единственный датчик видения (в варианте vSLAM), который является относительно дешевым вариантом, но полностью одна камера здесь не дает полного решения проблемы.

Из-за нелинейной модели наблюдения-видения зрительного датчика описанная выше интеграция с INS выполнена на основе вариантов: расширенного фильтра Калмана (ЕКФ), сигма-точечного фильтра Калмана (UKF) или распределенного фильтра частиц.

Результаты моделирования показывают, что предложенный распределенный фильтр имеет лучшие характеристики работы с точки зрения определения вектора изменений состояния и числа вычислений. Есть соответствующие оценки работы по сравнению с централизованным фильтром частиц, известным оптимальным фильтром. Вообще, известно, что ЕКФ обеспечивает большую скорость вычисления; однако у получаемой оценки возникает самая большая ошибка из-за особенностей фильтра Калмана. Централизованному фильтру частиц нужно самое большое время вычисления даже при том, что его результат оценки близок к распределенному. Таким образом, предложенный фильтр может получить более высокое качество оценки работы и уменьшить время вычисления, используя распределенную структуру фильтра.

В отличие от фильтра Калмана, созданный фильтр частиц может быть применен к нелинейным системным моделям с негауссовским шумом распределения; может быть способен к оценке состояния даже в том случае, когда трудно формировать модель измерения или когда распределение не гауссовское. Датчик видения дает базу для оценки, и она может быть улучшена при использовании фильтра частицы, даже в нелинейных случаях.

Одна из самых больших проблем в формировании фильтра INS/vSLAM состоит в том, что типы и число заметных характерных точек, а также комбинация доступных измерений датчиков изменяются, что приводит к изменению конфигурации фильтра.

Результаты моделирования показывают, что и аппаратно и программно рассмотренный фильтр является наиболее эффективным.

Заключение

Разработки методов SLAM ведутся и в России [10–13]. Особый интерес представляют работы [10, 11], где описана система типа monoSLAM, работающая лишь от единственного лазерного дальномера.

Важные результаты приведены в работе [13], где представлено сравнение работы трех алгоритмов типа SLAM, работающих по видеопотоку от единственной камеры. Рассмотрены monoSLAM, PTAM и LSD-SLAM. Эти алгоритмы фактически идентичны по используемой сенсорной аппаратуре (единствен-

ная видеочамера), но различаются методами определения особенностей на изображении. PTAM использует детектор углов на изображении, LSD-SLAM — области изображений для построения карты и пройденной траектории, что позволяет ему составлять высокодетализированные карты в режиме реального времени. Однако в целом метод monoSLAM позволяет строить наиболее точную траекторию по наименее детализированной карте.

Приведенное выше описание методов SLAM и результаты работы [13] приводят к выводу, что создание алгоритма SLAM, обеспечивающего универсальное решение задач одновременной локализации и картирования для мобильных агентов, еще далеко от завершения.

Список литературы

1. URL: https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping.
2. **Seren Riisgaard, Morten Rufus Bias.** SLAM for Dummies. A Tutorial Approach to Simultaneous Localization and Mapping. 127 p. URL: http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/laslam_blas_repo.pdf.
3. **Leonard J. J., Durrant-Whyte H. F.** Simultaneous map building and localization for an autonomous mobile robot // Intelligent Robots and Systems '91. Intelligence for Mechanical Systems. Proc. IROS'91. IEEE/RSJ International Workshop. 1991. P. 1442–1447. DOI: 10.1109/IROS.1991.174711
4. **Steder B., Grisetti G., Stachniss C., Burgard W.** Visual SLAM for Flying Vehicles. 5 p. URL: <http://ais.inforntatik.uni-freiburg.de/publications/papers/stder08tro.pdf>.
5. **Montemero M., Thrun S., Koller D., Wegbreit B.** FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem // Carnegie-Mellon University, Stanford University, Proc. of American Association for Artificial Intelligence, AAAI-02. 2002. P. 593–598.
6. **Dae Hee Won, Sebum Chun, Sangkyung Sung, Young Jae Lee, Jeongho Cho, Jungmin Joo, Jungkeun Park.** INS/vSLAM System Using Distributed Particle Filter // International Journal of Control, Automation, and Systems. 2010. 8 (6). P. 1232–1240. DOI: 10.1007/s12555-010-0608-7.
7. **Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, Olivier Stasse.** MonoSLAM: Real-Time Single Camera SLAM // IEEE Transactions On Pattern Analysis And Machine Intelligence. 2007. Vol. 29, N. 6. P. 1–16.
8. **Russo L., Rosa S., Bona B., Matteucci M.** A ROS Implementation of the Mono-SLAM Algorithm. 13 p. URL: <http://www.slideshare.net/csandit/a-ros-implementation-of-the-monoslam-algorithm>.
9. URL: <https://www.google.com/atap/project-tango/>
10. **Герасимов В. Н., Михайлов Б. Б.** Решение задачи управления движением мобильного робота при наличии динамических препятствий // Вестник МГТУ им. Н. Э. Баумана. Сер. "Приборостроение". 2012. Спец. выпуск № 6 "Робототехнические системы". С. 83–92.
11. **Герасимов В. Н.** Алгоритм SLAM на основе корреляционной функции // Экстремальная робототехника. Сб. докл. в Всеросс. науч.-техн. конф. 2015. С. 101–106.
12. **Кузнецов П. М.** Система навигации робота на базе визуальной одометрии. СПб: Изд. СПбГУ, 2012. 15 с. URL: <http://www.apmath.spbu.ru/ru/education/master/2012/kuznecov-prez.pdf>
13. **Боковой А. В.** Исследование методов одновременного картирования и локализации беспилотных летательных аппаратов по видеопотоку, полученному с единственной камеры // Тр. Междунар. науч.-техн. конф. "Экстремальная робототехника". Тр. II Всеросс. семинара "Беспилотные транспортные средства с элементами искусственного интеллекта (БТС ИИ)" (2015). 2015. С. 26–33.
14. URL: <http://my-it-notes.com/2013/01/slam-basis-and-links-at-open-source/>.
15. URL: https://en.wikipedia.org/wiki/Kalman_filter

SLAM Technologies for the Mobile Robots: State and Prospects

V. E. Pavlovsky, vlpavl@mail.ru✉,

Keldysh Institute of Applied Mathematics of RAS, Moscow, 125047, Russian Federation,

V. V. Pavlovsky, vlpavl2000@mail.ru,

Plekhanov Russian University of Economics, Moscow, 117997, Russian Federation

Corresponding author: **Pavlovsky Vladimir E.**, D. Sc., Leading Resercher, Keldysh Institute of Applied Mathematics of RAS, Moscow, 125047, Russian Federation, e-mail: vlpavl@mail.ru

Received on 29.12.2015

Accepted on 10.02.2016

SLAM is a modern technology of a simultaneous localization of a mobile robot in the environment of motion and mapping of the surrounding space. By using SLAM a robot determines its position (solves the problem of localization) in space and simultaneously builds a map of the environment. This is a major technology for approaching the problem of navigation of the mobile robots. Now certain algorithms are already available with the known solution for the problem of SLAM. SLAM algorithms are applied widely for control of the mobile devices moving on the roads or off roads, for control of the flying robots (autonomous robot planes, helicopters, dirigibles), and for control of the android walking robots. Of course, for each class of the mobile robots there are specific features of a concrete SLAM technology, but the main ideas remain the same. In the specified technology a robot orientates itself in relation to the known reference points, using the sensors providing information on the reference points. In the first SLAM algorithms video cameras and/or range finders were often used, and later satellite navigation sensors were used.

The article presents a brief review of the popular effective solutions to the SLAM problem based on various sensor processing algorithms and using different hardware. Among them the basic SLAM methods, such as MonoSLAM, FastSLAM, VisualSLAM, INS-SLAM, and INS-GPS-SLAM are considered. Their description is presented in the same order as information about these methods appeared in literature, and in the specified row, so the subsequent methods are more improved, than the preceding ones.

Now the above methods ensure a reliable navigation for the mobile robots, however, new methods appear, and research is going on, in particular, with the aim to increase the speed of the SLAM methods and develop new sensors providing information on the environment.

Keywords: navigation, localization, mapping, robot, SLAM

For citation:

Pavlovsky V. E., Pavlovsky V. V. SLAM Technologies for the Mobile Robots: State and Prospects, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2016, vol. 17, no. 6, pp. 384–394.

DOI: 10.17587/mau.17.384-394

References

1. **Available at:** https://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping.
2. **Soren Riisgaard, Morten Rufus Bias.** SLAM for Dummies. A Tutorial Approach to Simultaneous Localization and Mapping, 127 p., Available at: http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/laslam_blas_repo.pdf.
3. **Leonard J. J., Durrant-Whyte H. F.** Simultaneous map building and localization for an autonomous mobile robot, Intelligent Robots and Systems' 91. *Intelligence for Mechanical Systems. Proc. IROS'91. IEEE/RSJ International Workshop*, 1991, pp. 1442–1447, DOI: 10.1109/IROS.1991.174711.
4. **Steder B., Grisetti G., Stachniss C., Burgard W.** Visual SLAM for Flying Vehicles, 5 p., available at: <http://ais.informatik.uni-freiburg.de/publications/papers/stder08tro.pdf>.
5. **Montemerlo M., Thrun S., Koller D., Wegbreit B.** FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem, Carnegie-Mellon University, Stanford University, *Proc. of American Association for Artificial Intelligence, AAAI-02*, 2002, pp. 593–598.
6. **Dae Hee Won, Sebum Chun, Sangkyung Sung, Young Jae Lee, Jeongho Cho, Jungmin Joo, Jungkeun Park.** INS/vSLAM System Using Distributed Particle Filter, *International Journal of Control, Automation, and Systems*, 2010, vol. 8 (6), pp. 1232–1240, DOI: 10.1007/s12555-010-0608-7.
7. **Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, Olivier Stasse.** MonoSLAM: Real-Time Single Camera SLAM, *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 2007, vol. 29, no. 6, pp. 1–16.
8. **Russo L., Rosa S., Bona B., Matteucci M.** A ROS Implementation of the Mono-SLAM Algorithm, 13 p., available at: <http://www.slideshare.net/csandit/a-ros-implementation-of-the-monoslam-algorithm>.
9. **Available at:** <https://www.google.com/atap/project-tango/>
10. **Gerasimov V. N., Mikhailov B. B.** Reshenie zadachi upravleniya dvizheniem mobil'nogo robota pri nalichii dinamicheskikh prepyatstviy (Solution motion control tasks the mobile robot in the presence of dynamic obstacles), *Vestnik MGTU im. N. E. Bauman. Ser. "Priborostroenie"*. 2012, Spet. iss. 6 "Robototekhnicheskie sistemy", pp. 83–92 (in Russian).
11. **Gerasimov V. N.** Algoritm SLAM na osnove korrelyatsionnoi funktsii (), *Proc. of the All-Russian Scientific and Engineering. Conf. EkstremVnaya robototekhnika*, 2015, pp. 101–106 (in Russian).
12. **Kuznetsov P. M.** Sistema navigatsii robota na baze vizual'noi odometrii (Robot navigation system based on visual odometry), SPb., Publishing house of SPbGU, 2012, 15 p., available at: <http://www.apmath.spbu.ru/ru/education/master/2012/kuznecov-prez.pdf>
13. **Bokovoi A. V.** Issledovanie metodov odnoremennogo kartirovaniya i lokalizatsii bespilotnykh letatel'nykh apparatov po videopotoku poluchennomu s edinstvennoi kamery (Research of methods of one-dimensional mapping and localization of unmanned aerial vehicles for videopotokupoluchennomu with a single camera), *Proc. of the All-Russian Scientific and Engineering. Conf. EkstremVnaya robototekhnika*, Works Proc. Seminar "Unmanned Vehicles with Elements of Artificial Intelligence (AI BPS)", 2015, pp. 26–33 (in Russian).
14. **Available at:** <http://my-it-notes.com/2013/01/slam-basis-and-links-at-open-source/>.
15. **Available at:** https://en.wikipedia.org/wiki/Kalman_filter