

МЕТОД ФИЛЬТРА ЧАСТИЦ ДЛЯ ОПРЕДЕЛЕНИЯ ПЕРВОНАЧАЛЬНОГО ПОЛОЖЕНИЯ РОБОТА В ПОМЕЩЕНИИ

А.В. Бобков

alexander.bobkov@bmstu.ru

М.Н. Алхатиб

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Рассмотрена задача определения собственного положения робота по данным лидара с использованием модифицированного фильтра частиц. Задача актуальна для многих практических приложений, связанных с разработкой и эксплуатацией автономных мобильных платформ, например, для логистических роботов на автоматизированном производстве и в складском хозяйстве. Рассмотрен наиболее используемый в случае решения указанной задачи метод фильтра частиц. Метод заключается в генерации виртуальных частиц, отображающих вероятное положение робота, изменении состояния частиц по данным одометрии и последующей фильтрации в соответствии с совпадением наблюдаемых и предполагаемых данных лидара. Выявлен недостаток метода, к которому относится большое начальное число частиц, необходимое для быстрой сходимости метода. Предложена модифицированная схема на основе предварительного детектирования крупных элементов сцены в данных лидара. Для этого предложено применять преобразование данных в пространство Хафа, что позволяет выявить наиболее крупные прямолинейные отрезки в данных лидара и соотнести их с картой помещения. Для исследования свойств предложенного алгоритма проведено математическое моделирование в средах MATLAB, ROS, Gazebo. Численные эксперименты показали, что модифицированный метод позволяет существенно снизить начальное число частиц и работать в режиме реального времени даже для больших помещений

Ключевые слова

Робот, определение положения, пространство Хафа, фильтр частиц, автономная навигация, логистический робот

Поступила 09.02.2023

Принята 20.11.2023

© Автор(ы), 2024

Введение. Современное развитие промышленности невозможно представить без использования логистических роботов с той или иной степенью автономности. Автономные логистические роботы позволяют исключить человеческий фактор из вредных, опасных либо просто монотонных и неинтересных для человека видов деятельности. Изучению автономных мобильных роботов и автономной навигации посвящено множество научных исследований [1, 2]. В настоящей работе рассмотрена задача определения собственного положения робота (навигации) и наиболее важные алгоритмы ее решения.

Задача навигации мобильного робота заключается в определении его местоположения в среде и в нахождении пути, который приведет объект из текущего местоположения в заданное. Для навигации на поверхности Земли часто используется GPS, позволяющая определить текущее местоположение робота на карте и спланировать путь для обхода препятствий и достижения конечной цели. Однако при навигации в помещении использование GPS затруднено вследствие сильного ослабления сигналов элементами конструкций, а также многочисленных переотражений сигнала [3, 4]. Для помещений альтернативой GPS можно считать систему радиомаяков, однако их точное размещение представляет собой отдельную технически сложную задачу.

Из множества навигационных задач здесь ограничимся только определением первоначального положения робота. Такая задача решается при включении робота или после долгого перемещения робота в среде с числом ориентиров и их качеством, недостаточными для однозначного определения его положения.

Оценка положения робота с помощью фильтра частиц. Наиболее хорошо изученными алгоритмами определения положения робота по данным лидара являются различные варианты фильтра Калмана (расширенный фильтр Калмана (ЕКФ) [5], фильтр Ульмана [6, 7] (фильтр Калмана без запаха, UKF)) и фильтр частиц (PF), или локализация Монте-Карло (MCL). Фильтр частиц [8–11], похожий на фильтр Калмана, широко применяется в задачах локализации робота ввиду нелинейности характеристики оценки. Это метод оценки состояния динамической системы, который принимает и обновляет ее текущее положение на основе информации о движении команд управления и наблюдения. Фильтр частиц использует частицы или образцы, которые являются гипотезами для системы, находящейся в одном единственном состоянии. Например, если имеется 1000 частиц, то существует 1000 предположений о вероятном местоположении системы (робота). Алгоритм работы фильтра частиц состоит из двух шагов. Первый

шаг — прогнозирование, которое зависит от информации о командном движении, такой как скорость. Второй шаг — внесение поправки, которая учитывает совпадение между наблюдаемыми данными и местоположением частицы. Далее увеличивается вес той частицы, которая имеет большее значение согласования. Затем выполняется повторная выборка частиц, чтобы ограничить число частиц и исключить частицы с низким весом.

Метод фильтра частиц позволил решить множество проблем в задаче определения собственного положения. Первые применения фильтра частиц были ограничены малоразмерными помещениями с известной картой [12]. В настоящее время существует множество подходов, которые в основном ориентированы на локализацию роботов в различных средах [13, 14], например, за счет совместного использования фильтра частиц и системы радиомаяков. Работа фильтра частиц с использованием системы ультразвуковых датчиков вместо лидара описана в [15–17].

Существует большое число вариантов фильтра частиц, позволяющих улучшить такие характеристики метода, как время вычислений и точность оценки местоположения.

Адаптивный MCL (AMCL) предложен Д. Фоксом для сокращения времени вычислений MCL [18]. AMCL — алгоритм локализации робота для навигации в двумерной среде, в котором используется относительная энтропия для оценки состояния робота на известной карте. Это позволяет рассчитывать число частиц, необходимое для MCL, вместо применения их фиксированного числа [19, 20]. Кроме того, существует еще один вариант MCL, который представляет собой локализацию Монте-Карло с преобразованием нормального распределения (NDT-MCL). Это кусочно-непрерывное представление, в котором также используется фильтр частиц для оценки состояния робота на известной карте за счет представления пространства в виде набора нормальных распределений [21], что позволяет улучшить оценку положения.

Алгоритм адаптивной локализации Монте-Карло с преобразованием нормального распределения (ANDT-MCL), который обеспечивает большую точность и скорость вычислений, предложен в [22]. В алгоритме использованы дивергенция Кульбака — Лейблера, преобразование нормального распределения и систематическая повторная выборка.

Задачу определения собственного положения можно сформулировать следующим образом. Имеется перемещающийся внутри помещения робот. Робот имеет карту здания, поэтому он знает общую схему расположения стен и примерное расположение мебели, но не знает своего первоначального положения относительно карты. Он может быть где угод-

но, поэтому роботу необходимо использовать датчики и модель движения для оценки местоположения на карте. Пусть у робота есть лидар, возвращающий расстояние до объектов в поле зрения. Датчик возвращает облако точек, однако в реальности эти измерения зашумлены. Робот может сопоставить облако точек с картой окружающего пространства, чтобы определить свое местоположение.

В дополнение к зашумленным измерениям робот также может использовать данные одометрии. Точный расчет последующего положения вычисляется с использованием предыдущего положения и относительных измерений (линейной и угловой скорости). Ввиду наличия шума точность относительных измерений с течением времени снижается и им требуется корректировка.

Существует два возможных способа определения местоположения: 1) использование зашумленных данных лидара, которые определяют некоторые особенности окружающего пространства и которые можно сравнить с картой; 2) использованием зашумленных данных одометрии, которые оценивают пройденное роботом расстояние.

Для понимания того, как фильтр частиц решает проблему определения положения, рассмотрим следующую ситуацию. Будем полагать, что робот использует лидар для определения местоположения. Начальное местоположение робота на карте неизвестно. Можно утверждать, что робот находится в любой точке открытого пространства с равной вероятностью (рис. 1, *а*).

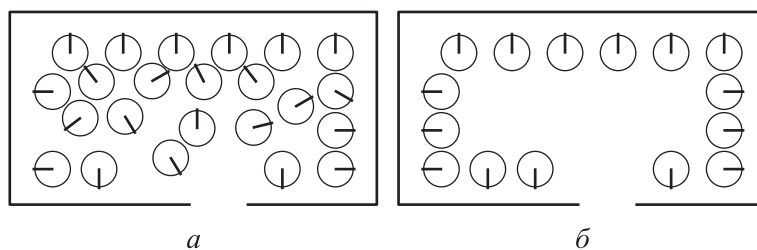


Рис. 1. Равномерное распределение положения и направления робота (*а*) и распределение вероятностей местонахождения робота после первого измерения (*б*)

Будем измерять сцену с использованием лидара и в результате робот определит, есть ли перед ним препятствие или нет, а также расстояние до препятствия. Однако он не может определить точное местоположение, так как стен много и робот может находиться перед любой из них. Робот может знать, что это, вероятно, не угол, и не дверь, и не середина

комнаты, потому что стена находится близко. Поскольку данные зашумлены, можно утверждать, что робот находится в одном из указанных мест с наибольшей вероятностью. Нельзя с полной уверенностью утверждать, что робот находится не в центре комнаты, так как он может смотреть на временное плоское препятствие, которого не было на карте. Такая вероятность мала, но она все же есть. Следовательно, после первого измерения распределение вероятностей местонахождения робота будет выглядеть так, как показано на рис. 1, б.

Пусть робот продолжает движение, чтобы увидеть другую деталь, например угол. Тогда можно снизить число возможных местоположений робота, удаляя те из них, которые не отвечают предыдущим измерениям. Если робот повернет налево и увидит угол, продолжит поворот и увидит другой угол, а затем увидит дверь, то он сможет использовать память, чтобы определить местоположение на карте. Основываясь на том, что робот видел и на оценках того, как он перемещался в окружающем пространстве и в каком направлении повернул, появляется возможность определить пройденный им путь и текущее местоположение.

Представим изложенное в виде формализованного алгоритма. Изначально у робота есть только карта комнаты и существует равная вероятность того, что он находится в любом ее месте и смотрит в любом направлении. Можно аппроксимировать распределение вероятностей, генерируя равномерное число случайных дискретных положений (см. рис. 1, а). Каждое положение или частица сохраняется как возможная оценка состояния робота.

Робот имеет информацию о перемещении (линейные и угловые скорости), которую можно использовать для каждой отдельной частицы, чтобы предсказать ее новое положение, если она движется так же, как и робот. Однако в этой информации существует неопределенность вследствие зашумленности данных одометрии и накапливающейся погрешности интегрирования, в силу чего реальное положение робота и положение частиц будут со временем расходиться. Чтобы избежать этого, необходимо использовать дополнительные навигационные данные для коррекции положения, например, получаемые с использованием лидара. В этом случае фильтр может последовательно просматривать все частицы и определять, какие данные вернет лидар, если положение частицы было бы реальным. Затем можно сравнить эти данные с данными лидара и определить, насколько вероятно, что данная частица отражает реальное состояние системы. Чем лучше совпадают данные, тем выше вероятность нахождения робота в состоянии текущей частицы (рис. 2).

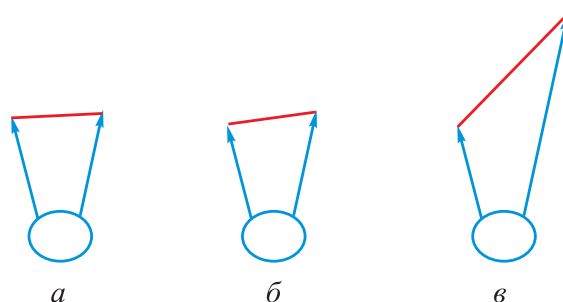


Рис. 2. Оценка вероятности по данным лидара:
a — более вероятное положение; *б* — данные лидара;
в — менее вероятное положение

Для каждой частицы развертка лидара (линия красного цвета, см. рис. 2) ориентируется относительно положения частицы, а затем вычисляется корреляция. Оценка корреляции:

$$S = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right) \left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}}, \quad (1)$$

где m, n — координаты пикселей; A_{mn} — точки карты; B_{mn} — данные лидара; \bar{A} , \bar{B} — средние значения пикселей карты и лидара (препятствия отмечены единицами, проходимое пространство — нулями). Расчет S по (1) позволяет получить число пикселей на карте и лидаре, которые перекрывают друг друга. Оценка корреляции S изменяется в диапазоне 0–1, она максимальна при полном совпадении данных и минимальна при полном несовпадении. Поэтому ее можно рассматривать в качестве вероятности совпадения состояний частицы и робота.

Процесс расчета корреляции повторяется для каждой частицы. Оценки корреляции для разных этапов сохраняются, образуя предысторию, которая применяется для расчета суммарной вероятности реализации точки. Она может быть найдена, например, как среднее по предыстории.

Далее будем перераспределять частицы на основе рассчитанных вероятностей так, чтобы число более вероятных частиц увеличивалось, а менее вероятных — уменьшалось. Для этого удалим частицы с наименьшей вероятностью и сгенерируем новые вблизи частиц с высокой вероятностью. Это приведет к тому, что частицы будут собираться в тесный кластер вблизи наиболее вероятного положения робота [9].

Алгоритм фильтра частиц

1. Задать первоначальное равномерное распределение частиц (x_i, y_i, φ_i) , $i = 1, \dots, N$.
2. Получить данные одометрии (линейные (v_x, v_y) и угловую (ω) скорости робота) и изменить положение каждой частицы: $x_i = x_i + v_x \Delta t$, $y_i = y_i + v_y \Delta t$, $\varphi_i = \varphi_i + \omega \Delta t$.
3. Получить данные лидара B .
4. Для каждой частицы построить симуляцию данных лидара A и рассчитать вероятность их совпадения с реальными данными B по (1).
5. Рассчитать вероятность реализуемости каждой частицы как среднюю вероятность совпадения данных по предыстории $p_i = \text{mean}(S_{ki})$.
6. Выполнить перераспределение частиц: выбранное число Q частиц с наихудшей вероятностью удаляется, а вместо них генерируется новое на основе числа Q частиц с наилучшей вероятностью и небольшого случайного отклонения.
7. Повторить шаги 2–6.

Чем больше частиц сгенерировано, тем выше точность определения местоположения, но тем больше времени требуется для расчета, поэтому существует компромисс между точностью и скоростью.

Модифицированный фильтр частиц. Основной недостаток фильтра частиц — большой объем вычислений, связанный с необходимостью генерации и проверки большого числа гипотез-частиц для точного вычисления местоположения [11]. Это особенно важно при глобальной локализации, когда местоположением может быть любая точка на карте, и число требуемых частиц очень велико. В этом случае необходимо применять более рациональные подходы к фильтрации частиц. Пример такого подхода — адаптивная локализация Монте-Карло [10], которая пересчитывает число частиц после каждого поколения.

Альтернативный подход заключается в том, что необходимое число частиц может быть уменьшено за счет устранения заведомо маловероятных положений при использовании некоторых эвристических допущений.

1. Наличие существенного числа линейных ориентиров. При навигации внутри помещений основной ориентир — стены, которые, как правило, являются ровными и плоскими. Отражение луча лидара дает прямолинейный отрезок, который легко найти и который существенно менее подвержен шуму, чем отдельные измерения лидара.

2. Наличие преобладающих направлений. Стены и другие ориентиры в помещении расположены не произвольно, а с доминированием опреде-

ленных направлений, кратных 90° . Это позволяет ограничить диапазон значений углов частиц, удалив из рассмотрения заведомо неверные направления.

3. Возможность задания траектории движения и обхода ориентиров так, чтобы неопределенность положения снималась наиболее быстро.

Детектирование прямолинейных участков в данных лидара. В настоящее время существует большое число детекторов прямых (например, на основе преобразования Хафа [23] или метода RANSAC), которые обеспечивают хорошее качество детектирования и высокую скорость. Здесь используем детектор на основе расширенного преобразования Хафа вследствие его крайне высокой скорости работы.

Переведем данные лидара из полярной системы азимут–дальность в декартову систему:

$$\begin{aligned}x(\alpha) &= R(\alpha) \cos \alpha; \\y(\alpha) &= R(\alpha) \sin \alpha; \\ \alpha &= 1 \dots 360^\circ,\end{aligned}$$

исключив данные с низкой мощностью и бесконечной дальностью.

Введем двумерное дискретное дуальное пространство (ρ, φ) параметров нормали к прямой и перенесем в него рассматриваемые точки в соответствии с преобразованием

$$\rho(\varphi) = x(\alpha) \cos \varphi + y(\alpha) \sin \varphi. \quad (2)$$

Выражение (2) представляет собой уравнение прямой в нормальной форме, где ρ и φ — длина и угол нормали к прямой, проходящей через точку (x, y) . Выражение (2) превращает точку (x, y) исходного пространства в синусоиду в дуальном пространстве. Соответственно, каждая точка (ρ, φ) дуального пространства задает прямую в исходном пространстве (рис. 3, а).

Точка пересечения наибольшего числа синусоид в дуальном пространстве (ρ^*, φ^*) даст нормаль к прямой, проходящей через наибольшее число точек в исходном пространстве.

Представим дуальное пространство в виде массива счетчиков H и массивов детекторов концов отрезков (X_A, Y_A, X_B, Y_B) . Для каждой точки (x, y) отсчета лидара будем перебирать углы $\varphi = 1 \dots 360^\circ$. Для каждого угла φ найдем $\rho(\varphi)$ и увеличим счетчик $H(\rho(\varphi), \varphi)$ на 1. Кроме того, изменим ячейки $X_A(\rho(\varphi), \varphi)$, $Y_A(\rho(\varphi), \varphi)$, $X_B(\rho(\varphi), \varphi)$,

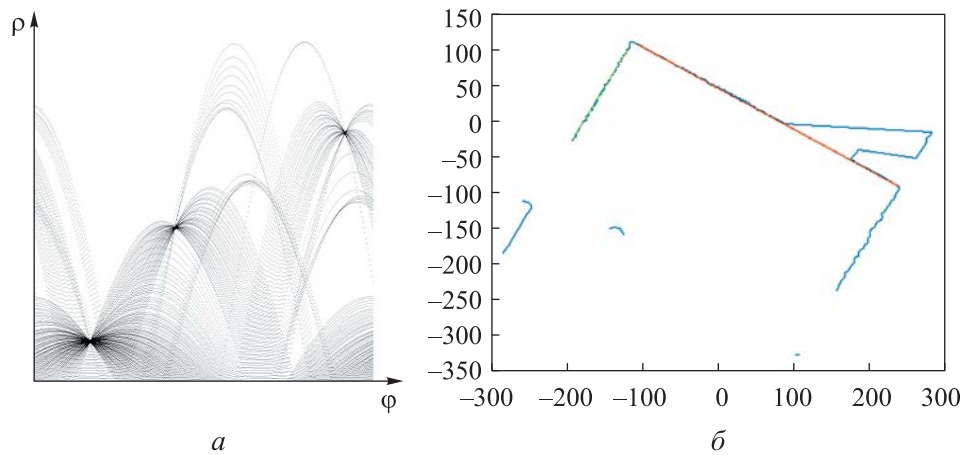


Рис. 3. Детектирование отрезков:

- a* — дуальное пространство параметров нормали: точки данных лидара в дуальном пространстве образуют синусоиды; точки, лежащие на одной прямой, формируют пучок синусоид, которые пересекаются в общей точке;
- б* — найденные отрезки (данные лидара (—), длинный (—) и короткий (—) отрезки)

$Y_B(\rho(\varphi), \varphi)$ так, чтобы они содержали минимальное и максимальное значения координат x и y соответственно.

По окончании построений найдем максимум в массиве H . Положение максимума $H(\rho^*, \varphi^*)$ даст параметры нормали к прямой, проходящей через максимальное число точек лидара, значения $X_A(\rho^*, \varphi^*)$, $Y_A(\rho^*, \varphi^*)$, $X_B(\rho^*, \varphi^*)$, $Y_B(\rho^*, \varphi^*)$ — координаты концов отрезка. Кроме того, величина ρ^* также задает расстояние до найденной стены.

Аналогично можно найти и другие отрезки, проанализировав оставшиеся локальные максимумы.

При наличии высокого уровня шума в данных лидара можно дополнительно выполнить верификацию найденных отрезков, пройдя по найденным линиям и собрав точки вблизи нее.

Алгоритм детектирования прямолинейных участков

1. Представить пространство поиска в виде двумерных массивов размером $R_{\max} \times 360$ и инициализировать их массив счетчиков $H = 0$; массивы координат левых нижних углов $X_A = Y_A = \infty$; массивы координат правых верхних углов $X_B = Y_B = 0$.

2. Определить данные лидара по дальности $R(\alpha)$ и мощности $P(\alpha)$, $\alpha = 1 \dots 360^\circ$.

3. Построить для каждого отсчета лидара $R(\alpha)$ с мощностью не ниже порога измерений $P(\alpha) > P_{\min}$, $\alpha = 1 \dots 360^\circ$, гармонический отклик в пространстве поиска.

3.1. Найти декартовы координаты α -й точки отражения луча лидара в исходном пространстве:

$$x(\alpha) = R(\alpha) \cos \alpha;$$

$$y(\alpha) = R(\alpha) \sin \alpha.$$

3.2. Построить отклик для каждого угла $\varphi = 1 \dots 360^\circ$.

3.2.1. Вычислить длину предполагаемой нормали к прямой

$$\rho = \text{int} [x(\alpha) \cos \varphi + y(\alpha) \sin \varphi].$$

3.2.2. Увеличить счетчик $H(\rho, \varphi) = H(\rho, \varphi) + 1$.

3.2.3. Скорректировать концы отрезка:

$$X_A(\rho, \varphi) = \min(X_A(\rho, \varphi), x(\alpha));$$

$$Y_A(\rho, \varphi) = \min(Y_A(\rho, \varphi), y(\alpha));$$

$$X_B(\rho, \varphi) = \max(X_B(\rho, \varphi), x(\alpha));$$

$$Y_B(\rho, \varphi) = \max(Y_B(\rho, \varphi), y(\alpha)).$$

4. Найти счетчик H с максимальным значением $h^* = \max(H)$ и его координаты ρ^* , φ^* .

5. Если значение h^* меньше порогового, то существенные линейные ориентиры в поле зрения отсутствуют и робот продолжает движение в поисках ориентира.

6. Если значение h^* оказалось выше порогового, то линейный ориентир полагается найденным и выполняется дополнительный поиск второго линейного ориентира с ортогональным направлением.

6.1. Найти максимум $h_1^* = \max(H)$ в области $\varphi = \varphi^* + 90 \pm \varepsilon$. Если значение h_1^* больше порогового, то дополнительный линейный ориентир полагается найденным.

7. Выполнить фильтрацию частиц в соответствии с найденными линейными ориентирами.

Пример работы алгоритма представлен на рис. 3, б.

Ограничения на генерацию частиц. Наличие длинного отрезка в данных лидара позволяет ввести ограничения на ориентацию генерируемых частиц: направление частицы должно лежать в диапазоне $\varphi^* + 90k \pm \varepsilon$,

$k = 0, 1, 2, 3$, ε — допуск на ошибку. Если в помещении есть неортогональные стены или другие протяженные объекты, то они могут быть учтены аналогично. Это позволяет существенно снизить число частиц с заведомо неверной ориентацией: например, при $\varepsilon = 5^\circ$ будет взят суммарный диапазон 40° вместо 360° , и алгоритм ускорится в 9 раз.

Наличие длинного отрезка в данных лидара задает ограничения и на положение, показывая, что камера находится именно на этом расстоянии от стен и только в такой ориентации (рис. 4, а).

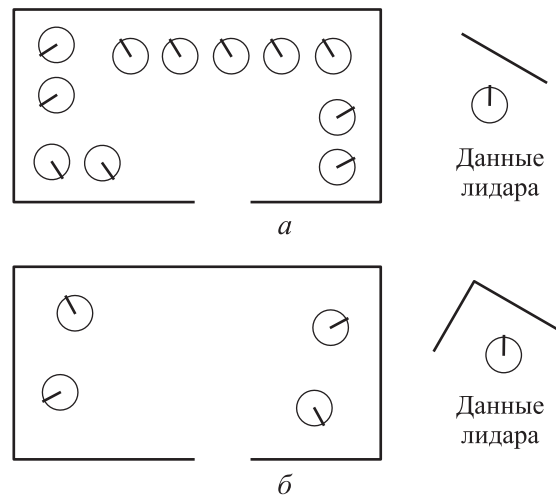


Рис. 4. Ограничения на положение камеры при наличии линейного (а) и углового (б) ориентиров

Наличие двух достаточно длинных и ортогональных отрезков в данных лидара позволяет задать точечный ориентир типа «угол» и еще более сократить область генерации точек (рис. 4, б). В этом случае существенная часть точек в заведомо неверных положениях просто исключается на ранних этапах. Однако здесь также появляется возможность того, что короткий отрезок создан не обозначенным на карте объектом, например, другим роботом.

Проверить соответствие расстояния от камеры до стены легко с использованием карты расстояний. В этом случае по заданной карте помещения для каждой точки заранее рассчитывается расстояние от нее до ближайшей точки стены. В процессе проверки данных лидара найденное значение ρ^* сравнивается со значением расстояния в анализируемой точке с учетом некоторого допуска.

Ускорить процесс ориентации можно, задав более удачный маршрут обхода ориентиров. Маршрут строится так, чтобы обеспечить близкий

к оптимальному обходу ориентиров, позволяющий как можно быстрее избавиться от неопределенности положения.

Ориентиры могут быть искусственными (например, фигурные маркеры) или естественными (например, стены, углы, проемы, а также специфические объекты, положение которых постоянно).

Вычислительный эксперимент. Для проверки предложенной модификации использовали MATLAB для реализации фильтра частиц и Gazebo для моделирования движения робота. Алгоритмы реализованы с использованием функции локализации Монте-Карло (MCL) на TurtleBot в среде Gazebo. Алгоритм требует известной карты, задача состоит в том, чтобы оценить положение и ориентацию робота на карте на основе движения робота и лазерного датчика.

Рассмотрим два варианта определения собственного положения робота: 1) глобальная локализация — робот может быть расположен в любом месте на карте (рис. 5, а); 2) локальная локализация — робот может находиться в определенной области на карте (рис. 5, б).

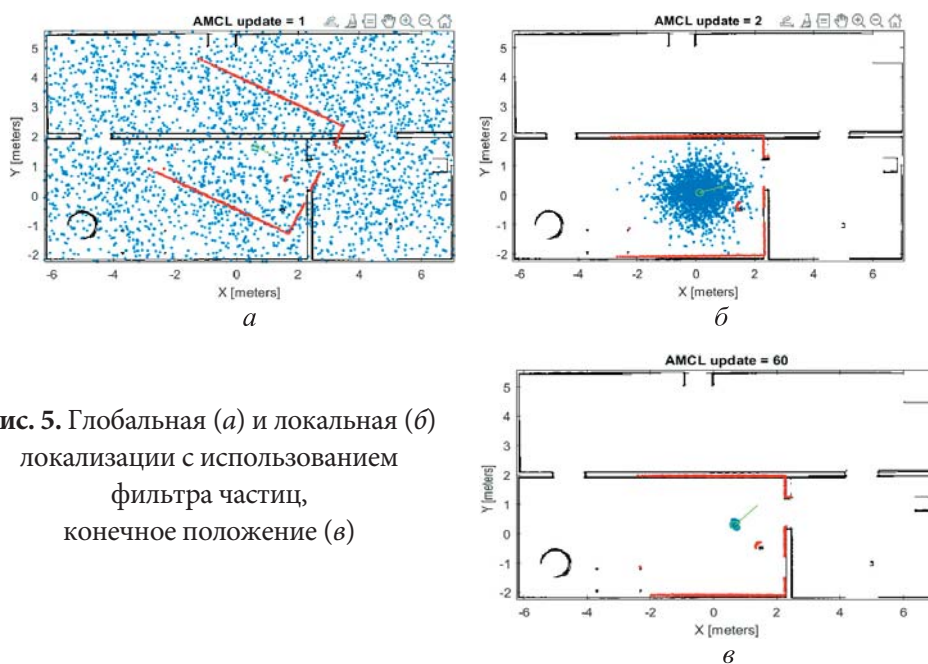


Рис. 5. Глобальная (а) и локальная (б) локализации с использованием фильтра частиц, конечное положение (в)

Робот перемещается по некоторой траектории, осматривая окружающую среду, и на каждом шаге вес частиц обновляется в зависимости от соответствия между данными лидара и картой. В конце пути робот может определить местоположение и ориентацию (рис. 5, в).

Выполним повторную выборку частиц, чтобы ограничить число частиц, исключить частицы с низким весом и сохранить частицы с большим

весом (веса зависят от совпадения). Необходимо получить хороший результат при оценке положения робота, когда используется большое число частиц. Однако это занимает больше времени, поэтому необходимо найти соотношение между числом частиц и временем выполнения. Для дальнейшей работы предложено использовать видеокамеру для повышения точности определения местоположения робота.

Для корректного определения начального положения робота в мире «Офис» среды Gazebo требуется первоначальная выборка примерно из 50 тыс. частиц (рис. 6, *а*). Для экспериментов использован компьютер с четырехъядерным процессором Intel® Core™ i7-4790 с тактовой частотой 3,6 ГГц. Время моделирования исходного алгоритма для 50 тыс. частиц составило 2,4 с на 1 итерацию. Из них 0,45 с составили постоянные затраты на моделирование, не связанные с расчетом положения частиц и не зависящие от их числа.

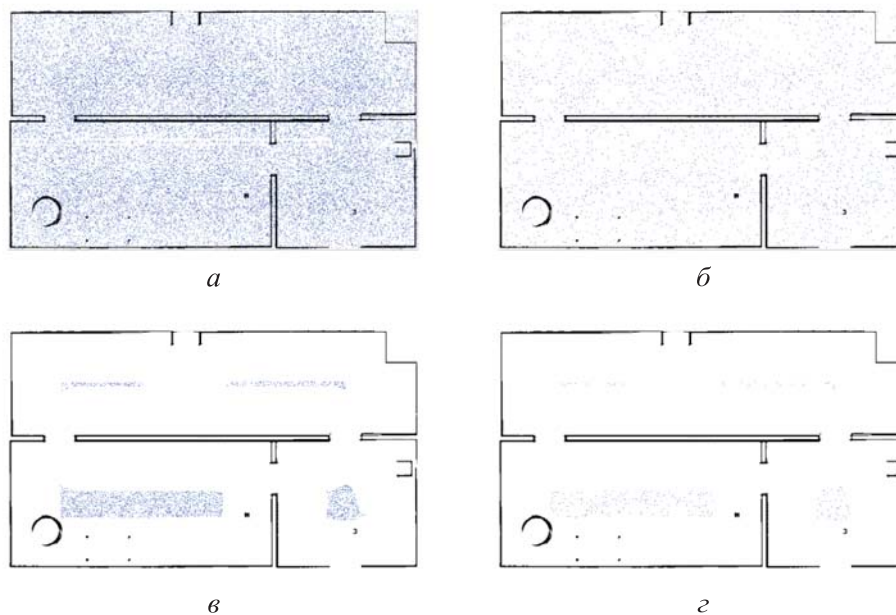


Рис. 6. Начальное распределение точек без учета данных лидара (*а*), с учетом ориентации (*б*), расстояния до стены (*в*) и ориентации и расстояния (*г*)

При наличии в поле зрения робота хотя бы одной стены можно ввести ограничения на ориентацию частиц. В этом случае существенная часть частиц может быть устранена как неправдоподобная по ориентации (рис. 6, *б*). Число частиц сокращается на порядок (без ухудшения качества работы алгоритма).

Наличие в поле зрения стены можно использовать и для задания ограничений на положение камеры. В этом случае существенная часть частиц может быть устранена как неправдоподобная по расстоянию до стены (рис. 6, в). Число частиц снова сокращается на порядок. Однако эффективность отсева будет сильно зависеть от положения робота относительно стены: число точек с малым расстоянием до стен больше, чем с большим.

Ограничения на угол и расстояние можно использовать одновременно. В этом случае получаем сокращение требуемого числа частиц на два порядка. Это позволяет алгоритму работать в режиме реального времени (рис. 6, г).

Среднее число частиц

Начальное распределение точек	49 684
Распределение с учетом:	
ориентации	5 497
расстояния до стены	4 544
ориентации и расстояния	520

Следовательно, предложенная модификация фильтра частиц позволяет существенно сократить начальное число частиц за счет использования априорной информации и анализа карты помещения и существенно ускорить работу алгоритма определения первоначального положения с помощью фильтра частиц, делая возможным его применение для навигации внутри помещения.

Заключение. Рассмотрена задача определения собственного положения робота по данным лидара с применением модифицированного алгоритма фильтра частиц. Показано, что использование априорной информации о геометрии помещения вместе с детектором линейных ориентиров позволяет существенно сгладить основной недостаток фильтра частиц — большой объем вычислений при отсутствии серьезных ограничений на начальное местоположение мобильного объекта. Установлено, что применение детектора Хафа для данных лидара позволяет надежно детектировать линейные ориентиры даже в зашумленных данных, а наличие успешно найденных ориентиров — сократить требуемое число виртуальных частиц на два порядка и соответственно повысить скорость вычислений, сделав возможным работу алгоритма в реальном времени.

В дальнейшей работе предполагается использовать данные видеокamеры вместо лидара в целях увеличения числа доступных ориентиров в поле зрения и ускорения работы алгоритма ориентации, а также повышения его точности.

ЛИТЕРАТУРА

- [1] Mohamed S.A.S., Haghbayan M.-H., Westerlund T., et al. A survey on odometry for autonomous navigation systems. *IEEE Access*, 2019, vol. 7, pp. 97466–97486. DOI: <https://doi.org/10.1109/ACCESS.2019.2929133>
- [2] Ullah I., Shen Y., Su X., et al. A localization based on unscented Kalman filter and particle filter localization algorithms. *IEEE Access*, 2019, vol. 8, pp. 2233–2246. DOI: <https://doi.org/10.1109/ACCESS.2019.2961740>
- [3] Miura S., Hisaka S., Kamijo S. GPS multipath detection and rectification using 3D maps. *IEEE ITSC*, 2013, pp. 1528–1534. DOI: <https://doi.org/10.1109/ITSC.2013.6728447>
- [4] Kos T., Markezic I., Pokrajcic J. Effects of multipath reception on GPS positioning performance. *Proc. ELMAR*, 2010, pp. 399–402.
- [5] Grewal M.S., Andrews A.P. Kalman filtering. Wiley, 2018.
- [6] Rekleitis I.A. Particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02. Centre for Intelligent Machines, McGill Univ., 2004.
- [7] Del Moral P. Nonlinear filtering: interacting particle solution. *Markov Process. Relat. Fields*, 1996, vol. 2, no. 4, pp. 555–580.
- [8] Zhang F., Li S., Yuan S., et al. Algorithms analysis of mobile robot SLAM based on Kalman and particle filter. *IEEE ICMIC*, 2017, pp. 1050–1055. DOI: <https://doi.org/10.1109/ICMIC.2017.8321612>
- [9] Thrun S., Burgard W., Fox D. Probabilistic robotics. Cambridge, MIT Press, 2005.
- [10] Heilig A., Mamaev I., Björn H., et al. Adaptive particle filter for localization problem in service robotics. *MATEC Web Conf.*, 2018, vol. 161, art. 01004. DOI: <https://doi.org/10.1051/mateconf/201816101004>
- [11] Marchetti L., Grisetti G., Iocchi L. A comparative analysis of particle filter based localization methods. In: Lakemeyer G., Sklar E., Sorrenti D.G., Takahashi T. (eds). *RoboCup 2006: Robot Soccer World Cup X. RoboCup 2006. Lecture Notes in Computer Science*, vol. 4434. Berlin, Heidelberg, Springer, 2006, pp. 442–449. DOI: https://doi.org/10.1007/978-3-540-74024-7_44
- [12] Thrun S. Particle filters in robotics. *Proc. UAI*, 2002, pp. 511–518. DOI: <https://doi.org/10.48550/arXiv.1301.0607>
- [13] Fox D., Thrun S., Burgard W., et al. Particle filters for mobile robot localization. In: Doucet A., de Freitas N., Gordon N. (eds). *Sequential Monte Carlo Methods in Practice. Statistics for Engineering and Information Science*. New York, Springer, 2001, pp. 401–428. DOI: https://doi.org/10.1007/978-1-4757-3437-9_19
- [14] Woo J., Kim Y.-J., Lee J., et al. Localization of mobile robot using particle filter. *SICE-ICASE Int. Joint Conf.*, 2006, pp. 3031–3034. DOI: <https://doi.org/10.1109/SICE.2006.315151>
- [15] Valin J.-M., Michaud F., Rouat J. Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering. *Rob. Auton. Syst.*, 2007, vol. 55, iss. 3, pp. 216–228. DOI: <https://doi.org/10.1016/j.robot.2006.08.004>

- [16] Arif S.A., Niaz M.H., Shabbir N., et al. RSSI based trilateration for outdoor localization in Zigbee based wireless sensor networks (WSNs). *IEEE CICN*, 2018.
DOI: <https://doi.org/10.1109/CICN.2018.8864943>
- [17] Wu K., Reju V.G., Khong A.W.H., et al. Swarm intelligence based particle filter for alternating talker localization and tracking using microphone arrays. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 2017, vol. 25, iss. 6, pp. 1384–1397.
DOI: <https://doi.org/10.1109/TASLP.2017.2693566>
- [18] Fox D. Adapting the sample size in particle filters through KLD-sampling. *Int. J. Rob. Res.*, 2003, vol. 22, iss. 12, pp. 985–1003.
DOI: <https://doi.org/10.1177/0278364903022012001>
- [19] György K., Kelemen A., Dávid L. Unscented Kalman filters and particle filter methods for nonlinear state estimation. *Procedia Technol.*, 2014, vol. 12, pp. 65–74.
DOI: <https://doi.org/10.1016/j.protcy.2013.12.457>
- [20] Montemerlo M., Thrun S., Koller D., et al. FastSLAM: a factored solution to the simultaneous localization and mapping problem. *Proc. AAAI Conf. on Artificial Intelligence*, 2002, pp. 593–598.
- [21] Saarinen J., Andreasson H., Stoyanov T., et al. Normal distributions transform Monte-Carlo localization (NDT-MCL). *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013, pp. 382–389. DOI: <https://doi.org/10.1109/IROS.2013.6696380>
- [22] Lim T.Y., Yeong C.F., Su E.L.M., et al. Enhanced localization with adaptive normal distribution transform Monte Carlo localization for map based navigation robot. *Electr. Eng.*, 2019, vol. 18, no. 3-2, pp. 17–24.
DOI: <https://doi.org/10.11113/elektrika.v18n3-2.193>
- [23] Duda R.O., Hart R.E. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM*, 1972, vol. 15, no. 1, pp. 11–15.
DOI: <https://doi.org/10.1145/361237.361242>

Бобков Александр Валентинович — канд. техн. наук, доцент кафедры «Системы автоматического управления» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1).

Алхатиб Мажди Насиб — аспирант кафедры «Системы автоматического управления» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1).

Просьба ссылаться на эту статью следующим образом:

Бобков А.В., Алхатиб М.Н. Метод фильтра частиц для определения первоначального положения робота в помещении. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2024, № 1 (146), с. 74–92. EDN: СТДРВЕ

INTRODUCING THE PARTICLE FILTER METHOD IN DETERMINATION OF THE ROBOT INITIAL POSITION IN PREMISES

A.V. Bobkov
M.N. Alkhatib

alexander.bobkov@bmstu.ru

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The paper considers the problem of determining the robot own position from the lidar data using a modified particle filter. The problem is relevant for many practical applications related to development and operation of the autonomous mobile platforms, for example, logistics robots in automated production and warehousing. The particle filter method most used in solving this problem is considered. The method involves generating the virtual particles that represent the robot probable position, changing the particles state according to the odometry data and subsequent filtering in accordance with matching in the observed and estimated lidar data. A drawback of the method was identified, i.e., high initial number of particles required in the method rapid convergence. The paper proposes a modified scheme based on preliminary detection of the large scene elements in the lidar data. It suggests introducing data transformation into Hough space for this purpose making it possible to identify the largest straight segments in the lidar data and correlate them with the premises map. Mathematical simulation was carried out in the MATLAB, ROS, and Gazebo environments to study the proposed algorithm properties. Experiments showed that the modified method could significantly reduce the particles initial number and make it possible to work in real time even in the large premises

Keywords

*Robot position determination,
Hough space, particle filter,
autonomous navigation,
logistics robot*

Received 09.02.2023

Accepted 20.11.2023

© Author(s), 2024

REFERENCES

- [1] Mohamed S.A.S., Haghbayan M.-H., Westerlund T., et al. A survey on odometry for autonomous navigation systems. *IEEE Access*, 2019, vol. 7, pp. 97466–97486. DOI: <https://doi.org/10.1109/ACCESS.2019.2929133>
- [2] Ullah I., Shen Y., Su X., et al. A localization based on unscented Kalman filter and particle filter localization algorithms. *IEEE Access*, 2019, vol. 8, pp. 2233–2246. DOI: <https://doi.org/10.1109/ACCESS.2019.2961740>

- [3] Miura S., Hisaka S., Kamijo S. GPS multipath detection and rectification using 3D maps. *IEEE ITSC*, 2013, pp. 1528–1534.
DOI: <https://doi.org/10.1109/ITSC.2013.6728447>
- [4] Kos T., Markezic I., Pokrajcic J. Effects of multipath reception on GPS positioning performance. *Proc. ELMAR*, 2010, pp. 399–402.
- [5] Grewal M.S., Andrews A.P. Kalman filtering. Wiley, 2018.
- [6] Rekleitis I.A. Particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02. Centre for Intelligent Machines, McGill Univ., 2004.
- [7] Del Moral P. Nonlinear filtering: interacting particle solution. *Markov Process. Relat. Fields*, 1996, vol. 2, no. 4, pp. 555–580.
- [8] Zhang F., Li S., Yuan S., et al. Algorithms analysis of mobile robot SLAM based on Kalman and particle filter. *IEEE ICMIC*, 2017, pp. 1050–1055.
DOI: <https://doi.org/10.1109/ICMIC.2017.8321612>
- [9] Thrun S., Burgard W., Fox D. Probabilistic robotics. Cambridge, MIT Press, 2005.
- [10] Heilig A., Mamaev I., Björn H., et al. Adaptive particle filter for localization problem in service robotics. *MATEC Web Conf.*, 2018, vol. 161, art. 01004.
DOI: <https://doi.org/10.1051/mateconf/201816101004>
- [11] Marchetti L., Grisetti G., Iocchi L. A comparative analysis of particle filter based localization methods. In: Lakemeyer G., Sklar E., Sorrenti D.G., Takahashi T. (eds). *RoboCup 2006: Robot Soccer World Cup X. RoboCup 2006. Lecture Notes in Computer Science*, vol. 4434. Berlin, Heidelberg, Springer, 2006, pp. 442–449.
DOI: https://doi.org/10.1007/978-3-540-74024-7_44
- [12] Thrun S. Particle filters in robotics. *Proc. UAI*, 2002, pp. 511–518.
DOI: <https://doi.org/10.48550/arXiv.1301.0607>
- [13] Fox D., Thrun S., Burgard W., et al. Particle filters for mobile robot localization. In: Doucet A., de Freitas N., Gordon N. (eds). *Sequential Monte Carlo Methods in Practice. Statistics for Engineering and Information Science*. New York, Springer, 2001, pp. 401–428. DOI: https://doi.org/10.1007/978-1-4757-3437-9_19
- [14] Woo J., Kim Y.-J., Lee J., et al. Localization of mobile robot using particle filter. *SICE-ICASE Int. Joint Conf.*, 2006, pp. 3031–3034.
DOI: <https://doi.org/10.1109/SICE.2006.315151>
- [15] Valin J.-M., Michaud F., Rouat J. Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering. *Rob. Auton. Syst.*, 2007, vol. 55, iss. 3, pp. 216–228. DOI: <https://doi.org/10.1016/j.robot.2006.08.004>
- [16] Arif S.A., Niaz M.H., Shabbir N., et al. RSSI based trilateration for outdoor localization in Zigbee based wireless sensor networks (WSNs). *IEEE CICN*, 2018.
DOI: <https://doi.org/10.1109/CICN.2018.8864943>
- [17] Wu K., Reju V.G., Khong A.W.H., et al. Swarm intelligence based particle filter for alternating talker localization and tracking using microphone arrays. *IEEE/ACM Trans. Audio, Speech, Language Process.*, 2017, vol. 25, iss. 6, pp. 1384–1397.
DOI: <https://doi.org/10.1109/TASLP.2017.2693566>

- [18] Fox D. Adapting the sample size in particle filters through KLD-sampling. *Int. J. Rob. Res.*, 2003, vol. 22, iss. 12, pp. 985–1003.
DOI: <https://doi.org/10.1177/0278364903022012001>
- [19] György K., Kelemen A., Dávid L. Unscented Kalman filters and particle filter methods for nonlinear state estimation. *Procedia Technol.*, 2014, vol. 12, pp. 65–74.
DOI: <https://doi.org/10.1016/j.protcy.2013.12.457>
- [20] Montemerlo M., Thrun S., Koller D., et al. FastSLAM: a factored solution to the simultaneous localization and mapping problem. *Proc. AAAI Conf. on Artificial Intelligence*, 2002, pp. 593–598.
- [21] Saarinen J., Andreasson H., Stoyanov T., et al. Normal distributions transform Monte-Carlo localization (NDT-MCL). *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013, pp. 382–389. DOI: <https://doi.org/10.1109/IROS.2013.6696380>
- [22] Lim T.Y., Yeong C.F., Su E.L.M., et al. Enhanced localization with adaptive normal distribution transform Monte Carlo localization for map based navigation robot. *Electr. Eng.*, 2019, vol. 18, no. 3-2, pp. 17–24.
DOI: <https://doi.org/10.11113/elektrika.v18n3-2.193>
- [23] Duda R.O., Hart R.E. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM*, 1972, vol. 15, no. 1, pp. 11–15.
DOI: <https://doi.org/10.1145/361237.361242>

Bobkov A.V. — Cand. Sc. (Eng.), Assoc. Professor, Department of Automatic Control Systems, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

Alkhatib M.N. — Post-Graduate Student, Department of Automatic Control Systems, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

Please cite this article in English as:

Bobkov A.V., Alkhatib M.N. Introducing the particle filter method in determination of the robot initial position in premises. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2024, no. 1 (146), pp. 74–92 (in Russ.).
EDN: CTDPBE