Филиал федерального государственного бюджетного

образовательного учреждения

высшего профессионального образования

«Национальный исследовательский университет

«МЭИ»

в г. Смоленске

номинация: «Исследования в области технических наук»

Киселев Константин Олегович, ФКТЭ,

Электроника и микропроцессорная техника, 4 курс, ПЭ-10

**КОМПЛЕКС АЛГОРИТМОВ РОБОТОТЕХНИЧЕСКОЙ НАВИГАЦИИ**

Смоленск-2014

Авторы научной работы:

 Киселев Константин Олегович

1 Актуальность и проблематика научной работы

Важной задачей в современной робототехнике является разработка способов определения положения робота в окружающем пространстве. Не зная положения робота в пространстве, не зная как выглядит окружающее пространство невозможно решить даже простейшую задачу движения из точки А в точку Б. Наиболее часто используемые способы определения положения – интегрирование перемещений робота (с помощью одометрии) или применение маяков, установленных в определенных местах. Использование маяков не универсально и требует предварительного оборудования рабочих помещений, при этом маяки постоянно должны быть в зоне видимости роботом. Интегрирование показаний одометров не обеспечивает точности позиционирования из-за накопления ошибки по всем отслеживаемым координатам.

При этом необходимо не только определение собственного положения, но и запоминание и сохранение изображения окружающего пространства для областей, информации о которых нет в памяти робота. Так же необходимо предотвращение столкновений робота с окружающими предметами, для этого эффективно использовать дальномеры.

Наиболее передовые алгоритмы используют изображение с веб-камер и дальномеров для определения положения робота. Эти методы являются более точными и универсальными. Разработками подобных алгоритмов занимаются ведущие мировые университеты. В иностранной литературе алгоритмы определения положения робота на карте одновременно с построением карты называют аббревиатурой SLAM (Simultaneous Location and Mapping). Большинство реализованных алгоритмов основаны на применении фильтров частиц. Для определения положения они используют "особые точки" в имеющихся данных, которых обычно не так уж много, особенно в привычных прямоугольных помещениях, где ими являются только углы. Поэтому при длительной работе возникают ошибки – полученный образ пространства искажается, хотя возможность навигации по карте сохраняется. Значит, необходим алгоритм вычисления положения робота, адекватно работающий при малом числе особых точек и не искажающий пространство при длительной работе.

2 Цели научной работы

Целями данной работы являются разработка более эффективного SLAM алгоритма, использующего все данные от сканирующего дальномера об окружающем пространстве, а не только "особые точки"; разработка алгоритма поиска положения робота на имеющейся в памяти карте.

3 Задачи научной работы

Разработка способа поиска положения робота на имеющейся в памяти карте, основываясь на показаниях сканирующего дальномера;

Разработка алгоритма определения перемещений робота, основываясь на показаниях сканирующего дальномера;

Разработка алгоритма построения образа окружающего пространства;

Исследование работоспособности разработанных алгоритмов.

4 Научная новизна и теоретическая значимость научной работы

Научная новизна работы заключается в следующем:

предложен алгоритм поиска положения робота на карте, хранящейся в памяти, основываясь на показаниях дальномеров;

разработан SLAM алгоритм, использующий максимальный объем данных для определения перемещений робота;

проведена оптимизация разработанного SLAM алгоритма;

проведено моделирование работоспособности алгоритмов в Matlab.

Практическая ценность работы состоит в следующем:

разработанные алгоритмы могут быть применены в реальных робототехнических системах для определения перемещений робота и построения образа окружающего пространства

сформулированы рекомендации к измерительной аппаратуре для применения алгоритма

5 Патентно-лицензионная ценность научной работы

Произведенный патентный поиск с помощью поисковых систем показал, что аналогичных решений в Российских патентных базах нет. Рассматривается возможность оформления заявки на получение патента на способ.

6 Материалы и методы исследования

В ходе работы были использованы следующие методы:

использование преобразования Хафа для поиска геометрических объектов;

метод математических преобразования для оптимизации вычислений;

корреляционный анализ;

Имитационное моделирование в Matlab;



а) б)

Рисунок 1 – Изображения а) реальной карты помещения б) карты помещения, полученной с помощью SLAM алгоритмов

На рисунке 1б видно типичное для SLAM (с помощью фильтров частиц) искажение образа пространства 1а.

Для устранения подобных искажений необходимо учитывать не только особые точки – разнообразные углы объектов, но и сами прямые объекты – стены и другие длинные прямые предметы. Для этого необходимо сначала выделить эти объекты во входных данных. Для поиска прямых линий обычно используется преобразование Хафа [D.H. Ballard, «Generalizing the Hough Transform to Detect Arbitrary Shapes»], входными данными для него являются двумерные изображения. Двумерное изображение можно построить, основываясь на информации от дальномеров.



Рисунок 2 – Обобщенный алгоритм работы системы определения положения робота

Алгоритм должен отслеживать окружающие робота прямые в течение времени работы робота и на основании параметров p и θ прямых определять положение робота. Параллельно с этим должна обновляться карта видимого окружающего пространства.

Необходим и поиск собственного положения на имеющихся картах для начальной привязки положения работа. Для этого предлагается воспользоваться корреляционным методом определения положения.

Работа проводилась в среде Matlab в режиме текстового программирования. Это обусловлено тем, что код из Matlab легко портировать на современные высокоуровневые языки программирования (С, С++) для применения разработанных алгоритмов в реальных роботах.

Начальное положение робота на имеющейся в памяти карте эффективно определять с помощью корреляционного анализа.

Предлагаемый способ заключается в применении дальномеров (ультразвуковых, лазерных или радиолокационных) для построения карты окружающего пространства. В дальнейшем эта карта сопоставляется с имеющейся картой помещения для получения текущего положения.

Исходными данными для работы являются: матрица M - план помещения (размером MaxNa) и массив измерений дальномера.

Матрица плана помещения представляет собой двумерный массив точек, каждая из которых может иметь одно из 3х значений:

* *a*c – свободное пространство
* *a*п – известные препятствия(стены, предметы мебели и пр.)
* *a*н – неисследованное пространство

Массив измерений дальномера представляет собой набор точек (*l*i, *α*i) – расстояния *l*i до объекта в направлении угла *α*i. При этом масштаб *l*i должен совпадать с масштабом плана помещения. По этим данным строится матрица T образа помещения.

Необходимо иметь переменную α0 – угол ориентации робота в помещении. При расчете положения обрабатываются несколько вариантов значения этого угла, если имеется информация о его предыдущем значении и максимально возможном изменении, либо перебираются все варианты от 0 до 360о с шагом Δα. Рассмотрим второй случай, тогда α0n=n\*360/Δα, где n –целое число, такое, что 0$\leq $a0n<360o.

Определяется *l*max - максимальное из расстояний *l*i, округленное до целого числа вверх. Матрица *T*n будет иметь размер MxN, где N=M=2\*(*l*max+2).

Зададим начало координат в точке (x0, y0) где x0=y0=*l*max+2. Зададим каждому элементу матрицы значение *a*н.

Зададим значение *a*с для всех точек матрицы *Т*n, лежащих на прямой между точками (x0, y0) и (x0+*l*i\*cos(α0n+*a*i); y0+*l*i\*sin(α0n+*a*i)). Для точки (x0+*l*i\*cos(α0n+*a*i); y0+*l*i\*sin(α0n+*a*i)) матрицы Т зададим значение *а*п. Если известна погрешность измерения расстояния ±Δl , то имеет смысл заполнить значениями *a*п ячейки между точками (x0+(*l*i-Δl)\*cos(α0n+*a*i); y0+(*l*i-Δl)\*sin(α0n+*a*i)) и (x0+(*l*i+Δl)\*cos(α0n+*a*i); y0+(*l*i+Δl)\*sin(α0n+*a*i)). При этом изначально следует принять значение *l*max=max(*l*i)+Δ*l*.

В результате получим n матриц Tn. Вычислив ВКФ матрицы M с каждой из матриц Tn, получим n матриц Cn – коэффициентов корреляции матриц Tn и M при различных взаимоположениях. Исходя из свойств коэффициента корреляции известно, что он принимает максимальное значение, когда данные на входе идентичны. И чем меньше расстояние между векторами на входе, тем выше значение коэффициента корреляции. Учитывая выражение, определяющее значение Cn(k,p)[1]:

$$C\_{n}\left(k,p\right)=\sum\_{m=0}^{M\_{a}-1}\sum\_{n=0}^{N\_{a}-1}M\left(m,n\right)\*T\_{n}\left(m+k,n+p\right)$$

$$0\leq k\leq M+M\_{a}-1$$

$$0\leq p\leq N+N\_{a}-1$$

Составим требования к значениям констант *а*с, *а*н, *а*п:

* Если нет данных о какой-либо точке, то эта точка не должна воздействовать на значение Cn(k,p), поэтому:

*а*н\**а*с=*а*н\**а*п=*а*н\**а*н=0 (1)

* Несовпадение значений в какой-либо точке должно приводить к снижению Cn(k,p), поэтому:

*а*н\**а*с<0 (2)

* Совпадение значений в какой либо точке должно приводить к возрастанию Cn(k,p), поэтому:

*а*н\**а*н=*а*с\**а*с>0 (3)

Из условия (3) следует, что числа *а*с, *а*н, *а*п – действительные, из условия (1) *a*н=0, из условия (2) следует, что *а*с, и *а*п отличны от нуля и имеют противоположные знаки.

Набор матриц Сn можно представить как трехмерную матрицу С размером nx(M+Ma-1)x(N+Na-1). Найдя максимум в этой матрице получим его координаты: (аc, xc, yc).

ac – информация о угле α0 поворота робота в пространстве:

α0= ac\*360/Δα

По xc, yc рассчитывают положение робота на карте по формулам:

x=x0+Ma-xc+1

y=y0+Na-yc+1

Тогда робот находится в точке M(x,y) и установлен в направлении α0. Эти координаты полностью описывают положение робота на плоскости, чего вполне достаточно для роботов, работающих внутри помещений.

Аналогичным образом возможно и определение координат в трехмерном пространстве. Входными данными при этом будет набор векторов (li,θi,φi) в трехмерной сферической системе координат, матрица M так же должна быть трехмерной.

При реализации алгоритма имеет смысл корректировать размер матриц Tn для увеличения быстродействия, т.к. если lmax значительно превосходит среднее расстояние до окружающих объектов, то значительная часть матрицы Tn останется заполненной *a*н и не будет принимать участия в расчете, однако размер Cn и объем расчетов при этом окажется выше необходимого. Вычисление корреляции так же эффективнее реализуется не по определению, а при помощи двумерного БПФ.

Проиллюстрируем работу метода моделированием в Matlab.



а б

Рисунок. 3 а - изображение карты тестового помещения, б - изображение карты тестового помещения для получения набора векторов (*l*i, *α*i).

На рис. 3 белым обозначены области, заполненные ac, черным – aп. Карта рис. 3а будет использоваться для расчета, эта карта хранится в памяти робота, используется в качестве матрицы *M*. Карта на рис. 3б имитирует наличие необозначенных в памяти робота препятствий: стульев, ног людей, мусорных урн и т.п.

При моделировании робот помещается в тестовую точку с произвольными координатами (при условии, что точка свободна) и произвольным углом α0, измеряется расстояние до предметов вокруг по карте рис. 3б. Расчет угла ведется с разрешением Δα=1о (n=360). Так же на расстояние до предметов действует шум с нормальным распределением. Используется алгоритм с вычислением корреляции с помощью БПФ по оптимизированной карте. Размер матрицы *M* 200x200. Расчет производился для 10 случаев, при этом каждый раз робот устанавливался в новую точку, не связанную с предыдущим расчетом.

Из таблицы 1 видим стабильную работу алгоритма. Погрешность расчета координат: ±1 (разрешение расчета); максимальная ошибка измерения угла 0,0098 рад = 0,56о, что меньше разрешения расчета.

Таблица 1 – Результаты моделирования

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | α0, рад. | α0, рад.Вычисл. | Х | Х выч. | Y | Y выч. | Ошибкаα0, рад. | Ошибка X | Ошибка Y |
| 1 | 1.8812 | 1.885 | 187 | 187 | 15 | 15 | 0.0038 | 0 | 0 |
| 2 | 0.5225 | 0.5236 | 30 | 30 | 175 | 175 | 0.0011 | 0 | 0 |
| 3 | 3.1297 | 3.1241 | 173 | 173 | 94 | 94 | 0.0056 | 0 | 0 |
| 4 | 1.0738 | 1.0821 | 69 | 69 | 58 | 58 | 0.0083 | 0 | 0 |
| 5 | 5.1389 | 5.1487 | 181 | 181 | 188 | 188 | 0.0098 | 0 | 0 |
| 6 | 3.4681 | 3.4732 | 167 | 167 | 103 | 102 | 0.0051 | 0 | 1 |
| 7 | 0.5715 | 0.576 | 162 | 162 | 93 | 93 | 0.0045 | 0 | 0 |
| 8 | 5.5558 | 5.5501 | 63 | 63 | 81 | 81 | 0.0057 | 0 | 0 |
| 9 | 2.1229 | 2.1293 | 30 | 30 | 171 | 171 | 0.0064 | 0 | 0 |
| 10 | 5.2929 | 5.2883 | 105 | 105 | 45 | 45 | 0.0046 | 0 | 0 |
| Макс. |  |  |  |  |  |  | 0.0098 | 0 | 1 |

Для обеспечения работоспособности алгоритма единственным требованием к дальномеру является минимальный апертурный угол. Этому требованию соответствует применение лазерных дальномеров. УЗ дальномеры обычно имеют апетрурный угол шире 10о, поэтому они сильно искажают данные измерений и не позволяют добиться высокой точности позиционирования.

Чувствительность алгоритма задается с помощью значений *а*п и *а*с. Корреляционный максимум имеет наиболее острый пик при *а*п=-*а*с. Однако такое сочетание параметров при обнаружении не отмеченных на карте объектов может приводить к ошибкам. Поэтому в некоторых случаях имеет смысл *а*с задавать равным *а*н, а *а*п>0.

Для реализации SLAM-метода, основанного на анализе прямых, окружающих робота необходимо следовать алгоритму, представленному на рис. 4. Этот алгоритм раскрывает этап вычисления перемещения из алгоритма на рис. 2. и выполняется на каждой итерации рабочего цикла робота. Последним этапом алгоритма является обновление карты окружающего пространства.



Рисунок 4 – Алгоритм вычисления перемещения робота

Вычислить перемещение между i и j измерениями можно по формулам:

$$\begin{matrix}ψ\_{i}=atan⁡\left(\frac{p\_{i,1}\*\cos(\left(θ\_{i,2}\right))-p\_{i,2}\*cos⁡(θ\_{i,1})}{p\_{i,2}\*\sin(\left(θ\_{i,1}\right))-p\_{i,1}\*sin⁡(θ\_{i,2})}\right)\\r\_{i}=\frac{\frac{p\_{i,1}}{cos⁡(θ\_{i,1})}+\frac{p\_{i,2}}{cos⁡(θ\_{i,2})}}{2}\end{matrix}$$

$$\begin{matrix}ψ\_{j}=atan⁡\left(\frac{p\_{j,1}\*\cos(\left(θ\_{j,2}\right))-p\_{j,2}\*cos⁡(θ\_{j,1})}{p\_{j,2}\*\sin(\left(θ\_{j,1}\right))-p\_{j,1}\*sin⁡(θ\_{j,2})}\right)\\r\_{j}=\frac{\frac{p\_{j,1}}{cos⁡(θ\_{j,1})}+\frac{p\_{j,2}}{cos⁡(θ\_{j,2})}}{2}\end{matrix}$$

$$\begin{matrix}∆x=r\_{i}\*\cos(\left(ψ\_{i}\right))-r\_{j}\*cos⁡(ψ\_{j})\\∆y=r\_{i}\*\sin(\left(ψ\_{i}\right))-r\_{j}\*sin⁡(ψ\_{j})\\∆φ=\frac{\left(θ\_{j,1}-θ\_{i,1}\right)+\left(θ\_{j,2}-θ\_{i,2}\right)}{2}\end{matrix}$$

Где pi,1, θi,1 и pi,2, θi,2 - параметры 1 и 2-ой прямых в измерениях под номером i, а pj,1, θj,1 и pj,2, θj,2 – параметры 1 и 2-ой прямых в измерениях под номером j. Δx, Δy, Δφ – изменение положения робота (координат и направления) между двумя моментами i и j. Эти уравнения выведены из соотношений изменения параметров двух пересекающихся прямых при перемещении начала координат.

Отслеживание наблюдаемых прямых основано на условии, что измерения расстояний должны производиться достаточно часто в процессе движения. При этом, чем чаще производится измерение, тем меньше смещение локального максимума, соответствующего конкретной прямой в накопительных пространствах преобразования Хафа текущего и предыдущего измерений.

Так, после вычисления преобразования Хафа в накопительном пространстве производится поиск локальных максимумов со значением более Hmin. Это обеспечивает использование в расчете только прямых, на которых лежит минимум Hmin точек. После необходимо отфильтровать локальные максимумы. Фильтрация осуществляется для исключения ложных локальных максимумов, образующихся вокруг истинных прямых из-за дискретности расчета и погрешностей измерений. При фильтрации сохраняются те прямые, на которые попало больше точек измерений.

Для установления соответствия прямых, обнаруженных в новых данных и известных ранее вычисляется рейтинг каждого из возможных вариантов связывания прямых из разных измерений и выбирается вариант с максимальным рейтингом.

Был опробован алгоритм Лукаса-Канаде [B. D. Lucas, T. Kanade, «[An iterative image registration technique with an application to stereo vision](http://www-cse.ucsd.edu/classes/sp02/cse252/lucaskanade81.pdf)» Proceedings of Imaging Understanding Workshop, 1981, -С. 121-130], используемый для определения оптического потока. Применение его для накопительного пространства преобразования Хафа позволило бы легко связать прямые из разных измерений. Однако результат работы оказался неудовлетворительным, т.к. алгоритм требует равномерности движения всех пикселей в исследуемой области и минимального изменения их значений. Ни одно из этих условий не выполняется, т.к. число измерений, попадающих на прямую при движении меняется (изменяется значение пикселей) и в зависимости от расстояния до прямой (которое изменяется при движении) изменяется острота максимума, что приводит к неоднородности потока. Поэтому потребовалось разработать метод связывания прямых, основанный на вероятностном подходе.

Для вычисления рейтингов вариантов связывания используются критерии, основанные на допущении, что границы помещения значительно не изменяются и измерения производятся достаточно часто: смещение локального максимума стремится к нулю, расстояния между параллельными линиями не изменяются.

Тогда вероятность Pi,j что i линия в новых измерениях является j линией предыдущих измерений, при условии что j линия имеет n известных параллельных линий k:

$$P\_{i,j}=\frac{a}{\sqrt{(p\_{i}-p\_{j})^{2}+(θ\_{i}-θ\_{j})^{2}}}+\sum\_{n}^{}\frac{b}{\sqrt{(p\_{i}-p\_{p}\_{k})^{2}+(θ\_{i}-θ\_{p}\_{k})^{2}}}$$

Где $θ\_{p}\_{k}$ и $p\_{p}\_{k}$ - параметры k параллельной к j линии из предыдущих измерений. Переменные a и b являются параметрами настройки алгоритма. Оптимальным значением для них будет a=b, равное максимально возможному перемещению между точками измерений. Это могут быть показания одометрии, либо значение, рассчитанное исходя из частоты измерений и максимальной скорости движения робота.

Pi,j образует матрицу вероятностей соответствия линий. Выбирая максимум в каждой строке линии различных измерений связываются между собой. Основываясь на этих данных, строится матрица истории линий Ln,i, где n – номер измерения в ходе работы алгоритма, а i – порядковый номер обнаруженной прямой в ходе работы алгоритма.

Для расчета перемещения между положениями n1 и n2 в матрице Ln,i находят соответствующие измерениям n1 и n2 группы линий, выбирают из них пару не параллельных и проверяют, чтобы угол между ними не изменился. Если подходящие для расчета линии находятся, то перемещение рассчитывают по формулам выше.

При относительно небольших перемещениях такой метод не дает значительного выигрыша по точности в сравнении с интегрированием показаний одометрии. Однако, если измерения n1 и n2 сделаны при достаточно большом перемещении робота, или до и после перемещения по сложной кривой, то точность значительно возрастает, т.к. погрешность расчета зависит от точности дальномера и разрешения вычисления преобразования Хафа и не накапливается в ходе работы.

Классическое преобразование Хафа является достаточно ресурсоемким. Для обеспечения работоспособности в реальном времени потребовалось разработать ускоренное преобразование Хафа для обработки показаний дальномеров, т.к. для применения классического преобразования Хафа набор входных данных (φi, ri) необходимо сначала преобразовать в двумерное изображение достаточного размера, чтобы вместить все точки (φi, ri), при этом точка расположения дальномера – начало системы координат, в которой представлены входные данные смещается к центру изображения, а преобразование Хафа принимает за начальную точку нижний левый угол изображения, что неудобно для дальнейшей обработки данных.

Выходными данными преобразования Хафа является двумерный массив H(p, θ), называемый накопительным пространством. Координаты локального максимума в этом пространстве определяют найденные прямые линии в виде:

$x\*\cos(\left(θ\right))+y\*\sin(\left(θ\right))=p$(4)

 где p и θ – определенны, с помощью преобразования Хафа параметры прямой.

При заполнении накопительного пространства для каждой не нулевой точки Ax,y входного изображения вычисляются все возможные проходящие через нее прямые. Через каждую точку Ax,y может проходить бесконечное число прямых, удовлетворяющих уравнению

$p\_{x,y}(θ)=x\*\cos(\left(θ\right))+y\*\sin(\left(θ\right))$ (5)

Изменяя θi от -90 до 90 градусов с шагом Δθ и округляя вычисленные по (5) значения px,y(θi) до ближайшего p'i=n\*Δp, где n – целое, а Δp – шаг расчета по расстоянию, получаем массив (pi',θi), содержащий набор параметров прямых, проходящих через точку Ax,y.

Далее значение каждой точки H(pi',θi) инкрементируется. Таким образом осуществляется голосование точками входного изображения A за проходящие через них прямые.

Найдя локальные максимумы в H(p,θ), определим все найденные прямые. Каждому локальному максимуму под номером j с координатами (pj,θj) соответствует прямая на изображении, определяемая выражением (4), где p=pj, θ=θj. Координаты при этом отсчитываются от левого нижнего угла изображения.

Так осуществляется вычисление преобразования Хафа для изображения.

При обработке показаний дальномеров на входе мы имеем список точек (φi,ri) длинной N. Для обработки таких данных потребуется предварительно преобразовать их в двумерное изображение размером rmax x rmax. Такой алгоритм имеет вычислительную сложность

O=(rmax/Δr)2\*180/Δθ (6)

Стоит отметить, что сложность напрямую не зависит от объема входных данных N, но зависит от самих данных – максимального значения расстояния rmax.

Чтобы избежать лишних вычислений, необходимо изменить формулу (2) с учетом того, что входными данными является набор точек Di=(φi,ri):

$p\_{i}(θ)=r\_{i}\*\cos(\left(φ\_{i}-θ\right))$ (7)

Аналогично (5) изменяя θi от -90 до 90 градусов с шагом Δθ и округляя вычисленные по (7) значения pi(θi) до ближайшего pi'=n\*Δp, где n – целое, а Δp – шаг расчета по расстоянию, получаем массив (pi',θi), содержащий набор прямых, проходящих через *i* точку входных данных. Для каждой *i* точки из D инкрементируя точки накопительного пространства H(p'i,θi) в результате получим двумерный массив преобразования Хафа, аналогичный массиву, полученному с использованием классического алгоритма. При этом модифицированный алгоритм имеет сложность

O=N\*180/Δθ (8)

Так же при этом не требуется дополнительного расхода оперативной памяти для хранения построенного по точкам Di изображения Ax,y.

 Рассмотрим работу обоих алгоритмов на примере линии с параметрами θ=45o, p=10. С помощью Matlab вычислим расстояния ri до такой линии из точки с координатами (0,0) в направлениях угла φi. Получим набор точек Di:



Рисунок 5 – массив Di в Matlab

Далее по классическому алгоритму его необходимо преобразовать в изображение A размером 33x33



Рисунок 6 – Входное изображение A

Вычислим классическое и оптимизированное преобразования Хафа.



Рисунок 7 – Результат вычисления классического преобразования Хафа



Рисунок 8 – Результат вычисления оптимизированного преобразования Хафа

Из изображений 7 и 8 видим, что координата X, соответствующая параметру θ, совпадает в обоих случаях, т.к. масштабы этих осей совпадают. Координаты Y отличаются, т.к. накопительные пространства имеют разный размер и преобразования принимают за начало координат разные точки. У оптимизированного преобразования Хафа оно имеет размер 180x33, а у стандартного 180x93. Это происходит из-за того, что классическое преобразование Хафа ведет отсчет координат из левого нижнего края исходного изображения, которое должно иметь размер 2$∙$rmax$×$2$∙$rmax чтобы полностью вместить полученные данные с дальномеров. Т.е. p может изменяться в пределах $-2\sqrt{2} r\_{max}÷2\sqrt{2} r\_{max}$. При вычислении оптимизированного преобразования Хафа p изменяется в пределах $- r\_{max}÷r\_{max}$.

Для правильной интерпретации параметров прямых результатом работы преобразований Хафа является не только накопительное пространство H, но и еще 2 массива для преобразования координат локального максимума в параметры прямой. А для классического преобразования Хафа потребуется дополнительный пересчет параметров найденной прямой в новую систему координат, связанную с центром изображения. Разработанное оптимизированное преобразование Хафа избавлено от этого недостатка. Аналогичным методом возможно и вычисление трехмерного преобразования Хафа для поиска плоскостей во входном облаке точек (ri, φi, θi). Тогда накопительное пространство H будет уже трехмерным

Для тестирования зададим траекторию движения робота, состоящую из движения по прямой и двух разворотов (121 точка). Разворот осуществлялся при старте, и после следования по первой прямой - поворот на вторую прямую. В качестве карты для проведения измерений используем карту, использованную для тестирования корреляционного алгоритма.



Рисунок 9 – Исходная карта и траектория движения

На рисунке 9 черным отмечены препятствия, а красным – траектория движения. При расчета разрешение вычисления преобразований Хафа составляло ∆θ=1o, ∆p=1 единица расстояния (пиксель). Расстояния на исходной карте измерялись с точностью 0.1 единица расстояния (пиксель). Проведя моделирование в Matlab, получим результаты:



Рис. 10 – Зависимость ошибки вычисленных координат от расчетной точки n. Зеленый – ошибка определения направления Ea(n). Синий – координаты Х Ex(n). Красный – координаты Y Ey(n).

Из рисунка 11 видим, что вычисление положения осуществляется с очень высокой точностью, и хотя ошибка нарастает, она остается на порядки ниже разрешения расчета. Используя данные о перемещении легко пересчитать данные с дальномеров для построения карты окружающего пространства. Полученная в ходе работы алгоритма карта изображена на рисунке:



Рисунок 11 – построенная в ходе моделирования карта

Из рисунков 10 и 11 видим, что карта построена корректно: на ней отображены все препятствия, зафиксированные дальномером. Все объекты сохранили свою исходную форму, видимых нелинейных искажений не заметно. Они принципиально не могли появиться, т.к. алгоритм заточен под работу с прямыми линиями и учитывает геометрию помещения.

Разработанные алгоритмы с успехом могут быть применены для реализации систем навигации различных роботов, таких как: роботы-пылесосы, роботы-официанты, роботы – экскурсоводы, роботы для помощи людям с ограниченными возможностями, охранных роботах. Возможно применение и в роботах для работы на улице, но там проблема навигации уже не стоит так остро, т.к. для навигации на уровне улиц обычно достаточно точности глобальных систем позиционирования.

**8 Результаты, теоретическая и практическая ценность научной работы**

Показаны преимущества SLAM-метода с использованием прямых над методом фильтра частиц.

Разработано оптимизированное для обработки показаний дальномеров преобразование Хафа.

Разработан алгоритм поиска положения робота на карте

Разработан алгоритм вычисления перемещений робота по карте, одновременно с построением карты(SLAM-метод)

Показана возможность использования алгоритмов для работы в трехмерном пространстве.

Сформулированы требования к оборудованию, необходимому для обеспечения работоспособности алгоритма.

Виртуальный эксперимент подтвердил работоспособность разработанных алгоритмов.

**8 Список литературы, опубликованной авторами по теме научной работы**

1. Киселев К.О. / [ПРИМЕНЕНИЕ КОРРЕЛЯЦИИ В РОБОТОТЕХНИЧЕСКОМ ЗРЕНИИ: НАВИГАЦИЯ В ПРОСТРАНСТВЕ](http://www.smolensk.ru/user/sgma/MMORPH/N-40-html/kiselev/kiselev.htm)// Математическая морфология. Электронный математический и медико-биологический журнал. - Т. 12. - Вып. 4. - 2013
2. Киселев К.О. / ОПТИМИЗИРОВАННОЕ ПРЕОБРАЗОВАНИЕ ХАФА ДЛЯ ОБРАБОТКИ ПОКАЗАНИЙ ДАЛЬНОМЕРОВ // Математическая морфология. Электронный математический и медико-биологический журнал. - Т. 12. - Вып. 1. – 2014
3. Киселев К.О. / Метод определения перемещений робота с помощью дальномеров // 11-я Международная научно-техническая конференция студентов и аспирантов «Информационные технологии, энергетика и экономика» - СФ МЭИ, 2014.