

## АЛГОРИТМ РАЗРАБОТКИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ МИНСКОГО МЕТРОПОЛИТЕНА

Пурхади А.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Кравцов А.Г. – доктор технических наук, профессор

В работе приводится один из алгоритмов работы мобильного приложения – алгоритм определения местоположения пользователя.

Мобильное приложение «Метро Минска» является транспортным навигатором с элементами справочно-навигационного приложения. Его основные задачи определение местоположения пользователя и построение оптимального маршрута от станции отправления к станции назначения.

Существует большое количество технологий для определения местоположения объекта внутри помещения. В последнее время, наибольшее распространение получили системы, основанные на использовании беспроводной сети базовых станций с возможностью измерений времени распространения и уровня сигнала. Часто для уточнения локации используют данные от встроенных в мобильное устройство акселерометра и гироскопа. Для расчета координат устройства по доступным измерениям в большинстве случаев применяются алгоритмы на основе байесовской фильтрации.

Обзор технологий и методов определения местоположения показал, что применение дискретного байесовского фильтра является удобным в случае отсутствия измерений от встроенного модуля определения траектории. Разработанный алгоритм позволяет рассчитать локацию с точностью до одного метра при использовании измерений от базовых станций, плана помещения и карты уровней сигналов.

Основной задачей алгоритма локации является оценка местоположения объекта по информации об измерениях от внешних и встроенных датчиков, а также карте помещения.

Помимо информации о координатах на плоскости  $(x, y)$ , алгоритм локации может определить направление движения объекта  $\alpha$  и его скорость  $v$ . Описанный набор параметров принято называть состоянием объекта или состоянием системы. В большинстве случаев набор указанных параметров объединяется в вектор, который называется вектором состояния системы.

В случае рассматриваемой системы вектор состояния  $x$  включает только координаты  $(x, y)$  объекта на плоскости. А дискретная сетка состоит из набора равномерно распределенных точек на плоскости, в которых может находиться объект.

Процедура генерации и выбора точек дискретной сетки представлена на рисунке 1. В верхней части рисунка выбор точек осуществляется без учета плана здания.

На первом шаге  $t = 1$  точки дискретной сетки генерируются внутри области пересечения окружностей, образованных новыми измерениями (рисунок 1а).

Затем к моменту получения новых измерений полученный набор точек расширяется во всех направлениях, ограниченных дискретной сеткой (рисунок 1б). Скорость расширения определяется максимально возможной скоростью движения объекта. Затем расширенный набор точек урезается при получении новых измерений (рисунок 1с и 1д).

В случае, если измерения происходят часто, количество точек, соответствующих возможным положениям объекта, заметно сокращается. Тем самым достигается сходимость и уменьшается вычислительная сложность.

Информация о структуре здания используется для уточнения локации. Расширение точек дискретной сетки в выбранном направлении может происходить только в том случае, если на пути распространения нет препятствия в виде стен. Учет стен позволяет ускорить сходимость фильтра и улучшить точность локации.

Входными данными являются набор точек  $G_{t-1}$  из  $N_{t-1}$  точек  $\{x_i, t-1, i = 1, \dots, N_{t-1}\}$ , определенных в предыдущий момент времени  $t - 1$ , серия измерений расстояний между мобильным устройством и базовыми станциями  $r_t: \{r_1, \dots, r_n\}$ , а также карта стен здания  $\Psi$ , представленная набором отрезков. Выходными данными является набор точек дискретной сетки  $G_t$  из  $N_t$  точек  $\{x_i, t, i = 1, \dots, N_t\}$ , полученный с учетом максимально возможной скорости движения и набора новых измерений.

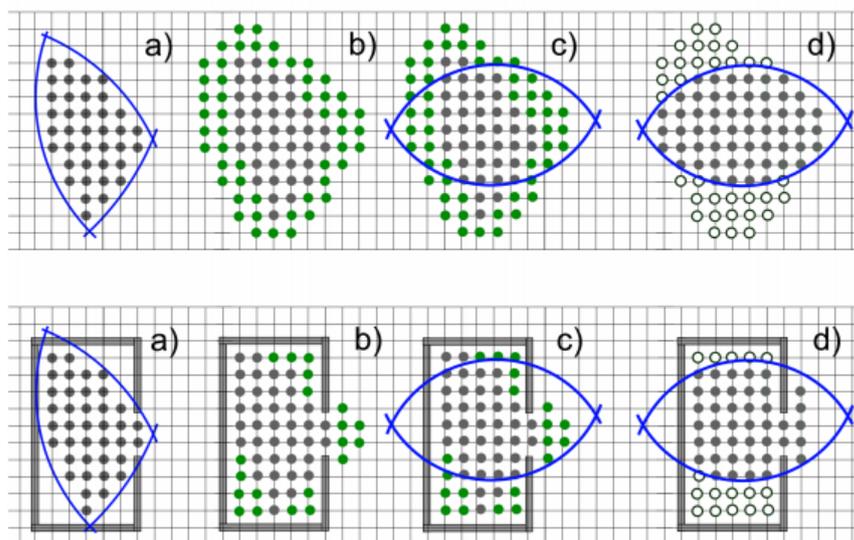


Рисунок 1 – Процедура выбора и отсечения точек дискретной сетки

Алгоритм использует следующие заранее определенные функции:

- $\text{intersects}(x_1, y_1, x_2, y_2, \Psi)$ , которая проверяет, пересекает ли отрезок  $(x_1, y_1) - (x_2, y_2)$  какую-либо из стен множества  $\Psi$  (рисунок 2);
- $\text{isInsideLocationArea}(x, y, r)$ , которая проверяет, находится ли точка с координатами  $(x, y)$  внутри области пересечения окружностей, построенных по измеренным расстояниям  $r$ .

Кроме того, предполагается, что период времени  $T$  между измерениями и размер дискретной сетки  $l_{grid}$  заранее определены. На рисунке 2 представлен алгоритм определения местоположения.

**input** :  $\mathcal{G}_{t-1} : \{x_{i,t-1}, y_{i,t-1}, i = 1 \dots N_{t-1}\}, \mathbf{r}_t, \Psi$

**output**:  $\mathcal{G}_t : \{x_{i,t}, y_{i,t}, i = 1 \dots N_t\}$

```

1 begin
2    $\mathcal{G}_t \leftarrow \emptyset$ 
3    $d \leftarrow T \cdot l_{grid}$ 
4   for  $i \leftarrow 1$  to  $N_{t-1}$  do
5      $x_1 \leftarrow x_{i,t-1} + d$ 
6      $y_1 \leftarrow y_{i,t-1} + d$ 
7      $x_2 \leftarrow x_{i,t-1} + d$ 
8      $y_2 \leftarrow y_{i,t-1} - d$ 
9      $x_3 \leftarrow x_{i,t-1} - d$ 
10     $y_3 \leftarrow y_{i,t-1} + d$ 
11     $x_4 \leftarrow x_{i,t-1} - d$ 
12     $y_4 \leftarrow y_{i,t-1} - d$ 
13     $x_5 \leftarrow x_{i,t-1}$ 
14     $y_5 \leftarrow y_{i,t-1}$ 
15    for  $j \leftarrow 1$  to 5 do
16      if !intersects  $(x_{i,t-1}, y_{i,t-1}, x_j, y_j, \Psi)$  and
17         isInsideLocationArea  $(x_j, y_j, \mathbf{r}_t)$  then
18         $\mathcal{G}_t \cup (x_j, y_j)$ 
19    return  $\mathcal{G}_t$ 

```

Рисунок 2 – Алгоритм определения местоположения пользователя

**Список использованных источников:**

1. K. E. Bekris, Robotics-based Location Sensing Using Wireless Ethernet // Proceedings of the 8th Annual International Conference on Mobile Computing and Networking. New York, NY, USA: ACM, 2002. P. 227–238.