GRANT journal
ISSN 1805-062X, 1805-0638 (online), ETTN 072-11-00002-09-4
EUROPEAN GRANT PROJECTS | RESULTS | RESEARCH & DEVELOPMENT | SCIENCE

Mobile Robot Localization and Path Planning in Open Street Map

Martina Dekanová[1]
František Duchoň[2]
Peter Pásztó[3]
Michal Adamík[4]
Marián Kľúčik[5]

[1] Fakulta elektrotechniky a informatiky STU; Ilkovičova 3, Bratislava, Slovakia; martina.dekanova@stuba.sk
[2] Fakulta elektrotechniky a informatiky STU; Ilkovičova 3, Bratislava, Slovakia; frantisek.duchon@stuba.sk
[3] Fakulta elektrotechniky a informatiky STU; Ilkovičova 3, Bratislava, Slovakia; peter.paszto@stuba.sk
[4] Fakulta elektrotechniky a informatiky STU; Ilkovičova 3, Bratislava, Slovakia; michal.adamik@stuba.sk
[5] Fakulta elektrotechniky a informatiky STU; Ilkovičova 3, Bratislava, Slovakia; marian.klucik@stuba.sk

Grant: VEGA 1/0752/17
Název grantu: Metódy inteligentného riadenia bezpilotných lietajúcich prostriedkov pre inšpekciu v priemyselnom prostredí
Oborové zaměření: JD - Využití počítačů, robotika a její aplikace

Abstract This paper deals with path planning algorithms used in mobile robot navigation and robot localization method is shortly described. The position of mobile robot is estimated by GNSS measurements. This position is then used for localization of mobile robot in a known map. The road map is obtained from Open Street Map. This map can be considered as a weighted graph. On this graph, two different path planning methods are applied: A* algorithm and Dijkstra's algorithm. These algorithms are compared at the end of this paper.

Key words Open Street Map, Path Planning, Dijkstra's algorithm, A* algorithm, GNSS Localization, Mobile Robot

1. INTRODUCTION

In the past decades, different conventional methods for localization and path planning in different environments have been developed. Robot path planning is an important area of research in robotics and artificial intelligence. Path planning is a task to create a collision free path that mobile robot will make between starting point and goal in known obstacles environment. [1] There exist many different path planning methods each suitable for different environments and maps. [1] In this paper we will be using road map, which consist of roads from Open Street Map. Our environment is static with known position of obstacles. The paper is organized in three sections. First section describes theory of used algorithms. In the second section is described practical part of our work and in the last section are results and conclusion.

2. THEORETICAL BACKROUND

In the following section, theory of all used algorithms is described. This part is divided in two parts: Localization and Path Planning.

2.1 Mobile Robot Localization

For localization the GNSS measurements are used which are preprocessed by Kalman filter. Kalman filter is a recursive filter which means that all previous measurements are incorporated in the last filtered measurement. Kalman filter theory which was used in our work is described in [2][3][4][5].

When the mobile robot is moving on a certain path, algorithm determines on which road robot is located. The problem is when a robot moves through a crossroad. In this situation algorithm works with hypothesis that robot can be on multiple roads. For this reason parameter – road probability is added. This parameter asses the chance that robot is moving on a certain road. This multiple hypothesis solution is solved based on fuzzy logic. Fuzzy logic theory of our algorithm is described in [4]. In the map roads have different lengths. When the mobile robot is localized on a specific road there is no exact position. Due to this, particle filter is applied to determine probable positionon road which robot is moving. Particle filter theory which was used in our work is described in [3][6].

2.2 Path Planning

Path planning finds the best path or the shortest path between two points. Many different methods of path planning exist. [7][8] In our application we use a road map, which can be considered as weighted graph. Typically for weighted graph algorithms as Dijkstra’s algorithm and A* algorithm or their variations are used. [9][10][11] Dijkstra’s algorithm computes all shortest paths from a given starting node in a fully connected graph. This algorithm requires information about distance between all connected nodes in graph. [7]

Dijkstra’s algorithm [7]:

- Set start distance to 0.
- Do this in Loop until all nodes are READY:
  - Select node n which shortest distance is not in READY set.
  - FOR each neighbor node m of n

IF distance[n]+edge(n,m)<distance[m] – shortest path find
THEN distance[m]=distance[n]+edge(n,m);
Predecessor[m]=n;

Necessary condition for Dijkstra's algorithm is that edge between two nodes must be a positive number. In figure 1 and in table I is solution for Dijkstra's algorithm.

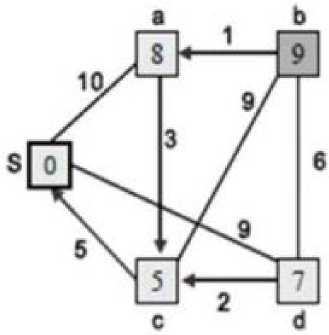

Figure 1: Dijkstra's algorithm - nodes with distance [7]

From s to:	S	a	b	c	d
Distance	0	8	9	5	7
Predecessor	-	c	a	s	c

Table I: Dijkstra's algorithm - numerical solution [7]

A* algorithm computes the shortest path from one given start node to one given goal node. This algorithm requires parameters/inputs such as distance between all connected nodes and lower bound of distance to goal from each node. Typically it is air-line or linear distance between node and goal node. [7]
A* algorithm [7]:

1. From actual node compute distance + estimate of remaining distance to goal to each connected node.
2. Choose node with shortest distance.
3. IF this is not goal node THEN go to 1.

The quality/precision of the lower bound goal distance from each node influences the computation demands of the algorithm. Shorter execution time is when the lower bound goal distance is closer to the true distance from node to goal node.

In figure 2 and table II is solution for A* algorithm in the same map as Dijkstra's algorithm.

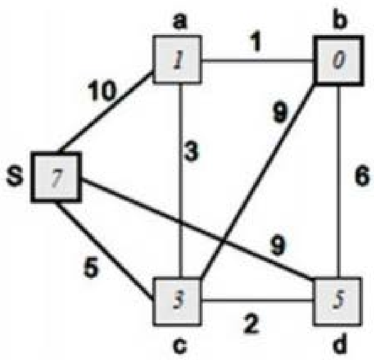

Figure 2: A* algorithm - nodes with distance and lower bound goal distance [7]

	Step 1	Step 2	Step 3
Distance	{S,a}=10+1=11	**{S,c,a}=5+3+1=9**	{S,c,a,b}=5+3+1+0=9
	{S,c}=5+3=8	{S,c,b}=5+9+0=14	-
	{S,d}=9+5=14	{S,c,d}=5+2+5=12	-

Table II: A* algorithm - numerical solution [7]

3. ALGORITHM

In the following section, the basic principles of proposed localization algorithm and path planning algorithm are described. The map where the mobile robot is localized is created from source obtained from Open Street Map.

3.1 Mobile Robot Localization

At first probability rating for each road is determined. This value describes odds that the mobile robot is located on the road. The probability is determined as a fuzzy function of the distance between GNSS position and its projection on the road. The function is determined on the basis of experimental measurements under different conditions. Fuzzy membership function is based on [4]:

- Precision of the sensor,
- shortest distance between filtered position,
- and the road and HDOP information from GGA sentence.

Secondly we use particle filter algorithm which is shown in figure 3 on roads with probability rating higher than 0.5.

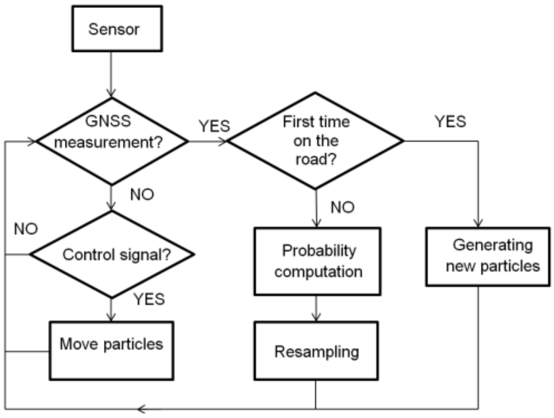

Figure 3: Particle filter algorithm [4]

The result of this algorithm is possible mobile robot location on the road. The localization algorithm is described in detail in our previous work [xx]. Verification of experimental test of localization algorithm is described in [4]. Result of this localization algorithm is in figure 4.

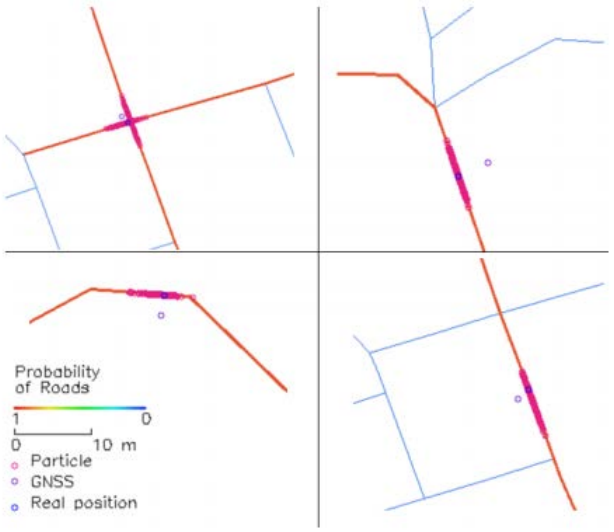

Figure 4: Results of localization algorithm [4]

The position from localization algorithm is input value to path planning algorithm. Center of gravity (CoG) of particles is taken as starting position for path planning algorithm. When this position is not in an existing node of road map, the road is split at this point. When it is not divided path planning algorithm can find longer path as is the shortest path to the goal from actual position.

3.2 Path Planning

As mention above the starting position for path planning algorithm is CoG of particles. Path planning algorithm starts in two situations:

- When the first position of mobile robot is determined.
- When mobile robot position is determined on another road than the path is calculated.

Both path planning algorithms mentioned requires input the distance between each connected node. Nodes position is in WGS84 coordinates system (longitude, latitude). Distance between nodes is calculated in kilometers by equation:

Distance (A, B) = 6378 * acos (cos (LatA) * cos (LatB) * cos (LngB - LngA) + sin (LatA) * sin (LatB))

Where latitude and longitude must be in radians, value 6378 is diameter of the Earth in kilometers according to WGS84 system.
In A* algorithm lower bound goal distance is calculated by the same equation as distance between two nodes. This distance is calculated from each node in map to the goal node.

More than 30 experiments were made on the road map. Both algorithms were used in experiments with different starting robot position like starting node, ending node and also between nodes in different distance from starting node. In 8 experiments recalculation of planned path was necessary, because robot was moving on different road as firstly localization algorithm determined. In next figure are shown some results computed by our algorithm.

In figure 5 is result for Dijkstra's algorithm and for A* algorithm. For this starting position and goal position is result same for both algorithms.

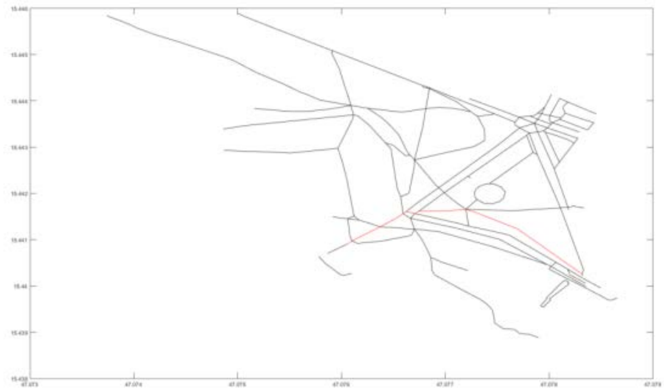

Figure 5: Result for A* algorithm and Dijkstra's algorithm with the same results

In figure 6 are results for both algorithms. In this situation A* algorithm and Dijkstra's algorithm has different results.

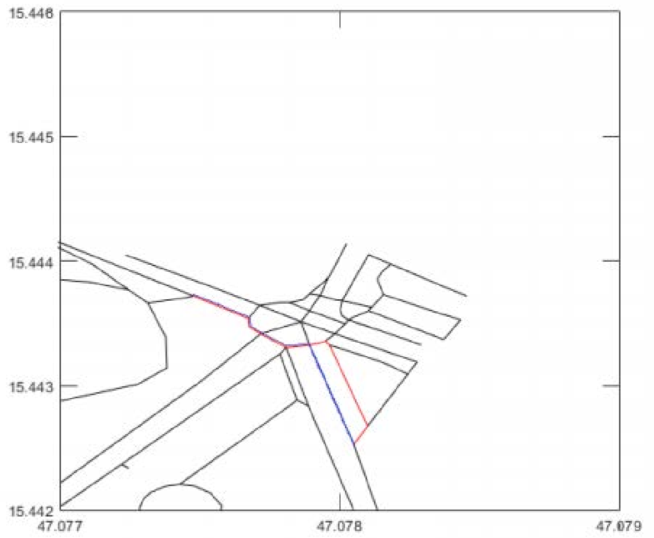

Figure 6: Result for A* algorithm (red) and Dijkstra's algorithm (blue) with different results

4. CONCLUSION

In this paper results of GNSS positioning and road identification with particle filter are used as actual mobile robot position for path planning in road map. In the introduction localization was described. Afterwards, used path planning theory was described. In the next section localization algorithm was shortly described and used path planning algorithm is described. At the end results of A* algorithm and Dijkstra's algorithm were shown and described. In our experiments A* algorithm and Dijkstra's algorithm have comparable time for path calculation. A* algorithm was slightly faster. 3 planned paths from actual position to goal position were different for A* algorithm and Dijkstra's algorithm. In figure 6 is one of these paths. Dijkstra's algorithm (blue) has shorter path than A* algorithm. It is caused by method of calculation. In our situation we choose Dijkstra's algorithm, because calculation speed are similar for both algorithms and Dijkstra's can find better path to the goal. When we will be navigating robot through bigger area A* algorithm can be better, because calculation will be faster than Dijkstra's algorithm.

Acknowledgment
The authors would like to thank the Ministry of Education, Science, Research and Sport of the Slovak Republic for funding project VEGA 1/0065/16, VEGA 1/0752/17.

Resources

1. SIEGWART, R., NOURBAKHSH, I.R. Introduction to Autonomous Mobile Robots. London: The MIT Press. 2004. ISBN: 0-262-19502
2. HOFMANN-WELLENHOF, B., LEGAT, K., WIESER, M. Navigation Principles of Positioning and Guidance. Wien: Springer, 2003. ISBN: 978-3-7091-6078-7
3. THRUN, S., BURGARD, W., FOX, D. Probabilistic Robotics. MIT Press, Massachusetts; 2005.
4. SZABOVÁ, M., DUCHOŇ, F., DEKAN, M., CHOVANEC, Ľ. Probabilistic Localization of Robot in Outdoor Environment Using GNSS. In EAN2017, Starý Smokovec, 2017.
5. SUBIRANA, J.S., ZORNOZA, J.M.J., HERNÁNDES-PAJARES, M. GNSS Data Processing, Vol I: Fundamentals and Algorithms. ESA Communications. Noordwijk; 2013
6. FOX, D., THRUN, S., BURGARD, W., DELLAERT, F. Particle Filters for Mobile Robot Localization. In: Sequential Monte Carlo Methods in Practice, Springer. 2015;401-428p.
7. BRAUNL, T. Embedded Robotics Mobile Robot Design and Applications with Embedded Systems. Berlin: Springer, 2006. 200-216p. ISBN 978-3-540-34319-6.
8. KOSTAVELIS, I., GASTERATOS, A. Semantic mapping for mobile robotics tasks: A survey. In: Robotics and Autonomous Systems. 2015, 86-103p.
9. WANG, H., YU, Y., YUAN, Q. Application of Dijkstra algorithm in robot pathplanning. In Mechanic Automation and Control Engineering (MACE), China, 2011, DOI 10.1109/MACE.2011.5987118.
10. KAVRAKI, L.E., SVESTKA, P., LATOMBE J.-C. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. In IEEE Transactions on Robotics and Automation, IEEE, 1996, 566-580p. DOI 10.1109/70.508439
11. VIRGALA, I., KELEMEN, M., KELEMENOVÁ, T., LIPTÁK, T., POLÁČEK, M. Design of Mobile Inspection Robot. American Journal of Mechanical Engineering.