TWO NEW APPROACHES TO MINIMUM DISTANCE LOCALIZATION

HAI CÁCH TIẾP CẬN MỚI BÀI TOÁN ĐỊNH VỊ KHOẢNG CÁCH NHỎ NHẤT

Dao Duy Nam; S.A. Ivanovskiy

Saint Petersburg Electrotechnical University "LETI"

Email: duynam81@yahoo.com; saivanovsky@mail.ru

ABSTRACT

The problem of minimum distance localization in environments that may contain self-similarities is addressed. A mobile robot is placed at an unknown location inside a 2*D* self-similar polygonal environment *P*. The robot has a map of *P* and can compute visibility data through sensing. However, the self-similarities in the environment mean that the same visibility data may correspond to several different locations. The goal, therefore, is to determine the robot's true initial location while minimizing the distance traveled by the robot. We consider four approximation algorithms for the robot localization problem. Two approximation algorithms are presented to solve minimum distance localization. One of them is based on a simple triangulation of a polygon representing a map. And another algorithm is based on using windows of the overlay intersection region of a simple polygon representing a map. A simulation program of experimental studies of these algorithms is conducted. The numerical results and their interpretation are given.

Key words: Robotics; robot localization; computational geometry; algorithm complexity; approximation algorithm; polygon triangulation

TÓM TẮT

Bài toán định vị khoảng cách nhỏ nhất trong môi trường bản đồ chứa đồng dạng được đề cập. Một rô bốt được đặt tại một vị trí không biết trước trong đa giác bản đồ hai chiều *P* có chứa các đa giác giống nhau. Rô bốt có bản đồ *P* và có thể tính toán dữ liệu vùng nhìn thấy nhờ vào cảm biến. Tuy nhiên, các hình đồng dạng trong bản đồ là những dữ liệu vùng nhìn thấy giống nhau tương ứng một vài vị trí khác nhau. Mục đích là xác định đúng vị trí ban đầu trong khi rô bốt di chuyển khoảng cách nhỏ nhất. Chúng ta đề cập tới bốn thuật toán gần đúng cho vấn đề định vị rô bốt. Hai thuật toán gần đúng được giới thiệu để giải quyết bài toán định vị khoảng cách nhỏ nhất. Một trong số đó dựa vào tam giác hóa đa giác bản đồ. Và thuật toán khác sử dụng cửa sổ của vùng giao của đa giác bản đồ. Các nghiên cứu thực nghiệm trên chương trình mô phỏng được thực hiện. Các kết quả và các diễn giải được đưa ra.

Từ khóa: Kỹ thuật rô bốt; định vị rô bốt; hình học máy tính; độ phức tạp thuật toán; thuật toán gần đúng; tam giác hóa đa giác

1. Introduction

Numerous tasks for a mobile robot require it to have a map of its environment and knowledge of where it is located on the map. Determining the position of the robot in the environment is known as the *robot localization problem* [1]. We are given a random environment modeled by an *n*-vertex simple polygon *P* without holes positioned somewhere in the 2*D* plane (see fig. 1). A mobile robot is placed at an unknown initial location within *P*, for which it has a map. First, the robot must determine if its initial location is unique by sensing its surroundings and matching the resulting visibility polygon *V* to the map of the environment. Given *P* and *V*, the robot must generate the set *H* of all hypothetical locations $p_i \in P$ such that the visibility at p_i is congruent under translation to *V*. Next, the robot must determine its true initial location by sensing and traveling in order to eliminate all hypothetical locations but one from *H*, while minimizing the distance traveled.

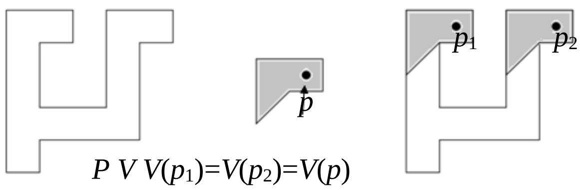

***Figure 1.** Given a map polygon P(left) and a visibility polygon V (center), the robot must determine which of the 2 possible initial locations p_1 and p_2 (right) is its actual location in P*

2. Solutions

Well-know algorithms of the robot localization problem [2], [3], [4] have two phases: hypothesis generation and hypothesis elimination. Hypothesis generation: Given *P* and *V*, determine the set *H* of all points $p_i \in P$ such that the visibility polygon of p_i is congruent under translation to *V* (denoted by $V(p_i) = V$); $H = \{h_1, h_2, ..., h_k\}$, $(\forall i \in 1..k \mid h_i : p = p_i)$.

Hypothesis elimination: Devise a strategy by which the robot can correctly eliminate all but one hypothesis from *H*, thereby determining its exact initial location. Ideally, the robot should travel a distance as small as possible to achieve this. An optimization problem of localization mobile robot is NP-hard [2]. Further four approximation algorithms of localization mobile robot are considered:

1. Algorithm 1 localization mobile robot with using decomposition on visibility cells of a map [2].
2. Algorithm 2 randomization in the hypothesis elimination [3].
3. Algorithm 3 localization mobile robot with using triangulation map (It is offered by authors).
4. Algorithm 4 localization mobile robot with using windows map(It is offered by authors).

We can now present the algorithm 3 localization mobile robot with using triangulation map (*LUT*). Given an input polygonal environment *P* and a robot placed at an unknown initial location in *P*, the *LUT* algorithm proceeds as follows:

1. Sense visibility polygon *V* from the robot's current unknown initial location.
2. Triangulate the map.
3. Generate the set of all hypothetical locations *H* in the environment *P* that match the visibility polygon sensed *V*.
4. Choose an arbitrary hypothetical location in *H* as the origin.
5. Construct an overlay arrangement centered on the origin.
6. Compute the connected overlay intersection component containing the origin, *OI* [3].
7. Find a point *r* in the set of points at middle of edges triangles and centers of triangles in the current overlay intersection area *OI*, which is the point nearest the origin and can eliminate some hypotheses.
8. The robot moves to the point *r* in the overlay.
9. Now, eliminate hypotheses by comparing visibility polygon sense by the robot at *r* with the visibility polygon computed at all the equivalent points corresponding to all the active hypotheses.
10. Let us call the set of eliminated hypotheses *E*. We repeat the overlay arrangement with reduced set of hypotheses *H* – *E*. Steps 4 – 9 are repeated until only one hypothesis, corresponding to the true initial location of the robot, is left in *H* – *E*.

We can now present the algorithm 4 localization mobile robot with using windows map (*LUW*). Given an input polygonal environment *P* and a robot placed at an unknown initial location in *P*, the *LUW* algorithm proceeds as follows:

1. Sense visibility polygon *V* from the robot's current unknown initial location.
2. Generate the set of all hypothetical locations *H* in the environment *P* that match the visibility polygon sensed *V*.
3. Choose an arbitrary hypothetical location in *H* as the origin.
4. Construct an overlay arrangement centered on the origin.
5. Compute the connected overlay intersection component containing the origin, *OI* [3].
6. Find a point *r* in the set of points at middle of windows [4] in the current overlay intersection area *OI*, which is the point nearest the origin and can eliminate some hypotheses.
7. The robot moves to the point *r* in the overlay.

8. Now, eliminate hypotheses by comparing visibility polygon sense by the robot at r with the visibility polygon computed at all the equivalent points corresponding to all the active hypotheses.

9. Let us call the set of eliminated hypotheses E. We repeat the overlay arrangement with reduced set of hypotheses $H - E$. Steps 3 – 8 are repeated until only one hypothesis, corresponding to the true initial location of the robot, is left in $H - E$.

Comparative theoretical performances of complexity of algorithms 1 and 2 are considered in [5].

Complexity of algorithm 3 and 4 can be estimated as $O(n^4 \log n)$.

3. Study results and comments

The experimental research is based on the program realization of algorithms 1, 2, 3 and 4 (Visual C ++ 2010) and comparison of such performances of algorithms, as magnitude of a summarised way of transition of the robot and an operating time of algorithm of localization. At carrying out experiment various modelling types of maps generation were used. Examples of such generation (a polygon of a map in the form of "steps") and works of algorithm of localization are reduced on fig. 2, 3, 4, 5.

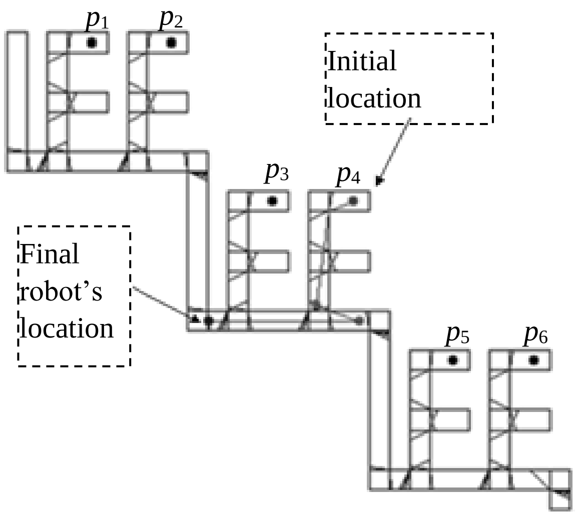

Figure 2. *Algorithm 1 localization mobile robot with using decomposition on visibility cells of a map*

On fig. 2 for the generated modelling map the outcome of work of the algorithm 1 using

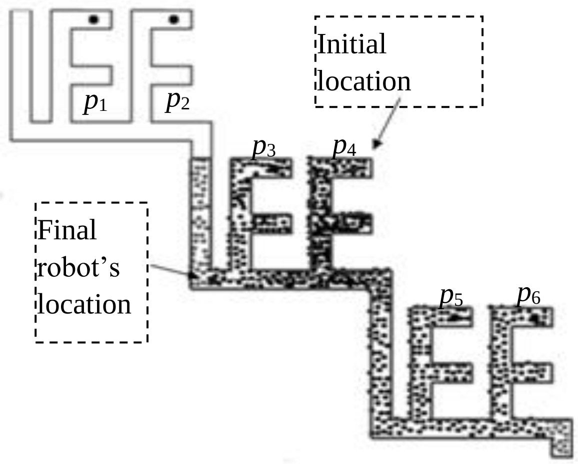

Figure 3. *Algorithm 2 randomization in the hypothesis elimination*

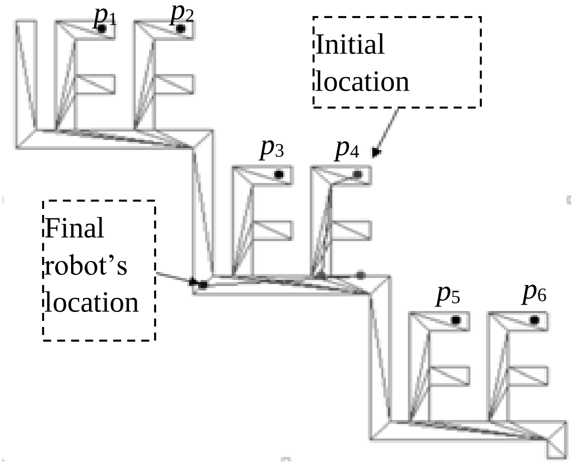

Figure 4. *Algorithm 3 localization mobile robot with using triangulation map*

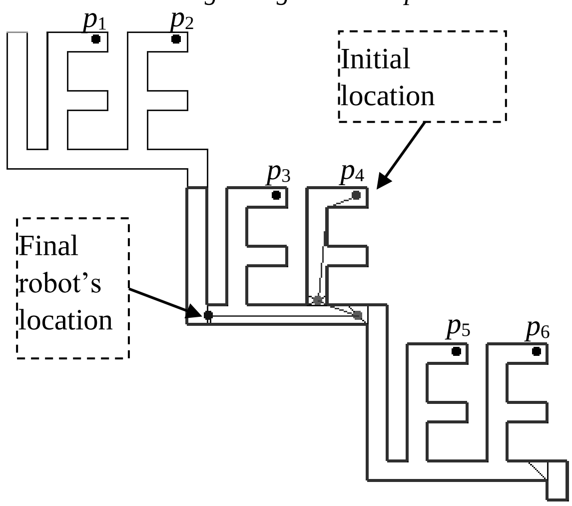

Figure 5. *Algorithm 4 localization mobile robot with using windows map*

decomposition of a map on skeleton of visibility is reduced. Here the map at n = 76 looks like three "steps" with the branches placed on them and number of hypotheses $k = 6$.

For the generated modelling map with the same values k and n on fig. 3 the outcome of work randomized algorithm 2, on fig. 4 - outcome of work of triangulation algorithm 3 and on fig. 5 - outcome of work of windows algorithm 4 is reduced.

In Table 1 and Table 2 outcomes of similar experiments are reduced at a size of a polygon of a map n = 148 (6 "steps") and number

Table 1. *Length of way passed by the robot at localization*

№	*d*				
	Algorithm 1	Algorithm 2		Algorithm 3	Algorithm 4
		$X = 100$	$X = 500$		
1	52,8	52,2	51,3	58,2	52,4
2	118,0	114,4	113,8	111,1	109,9
3	269,6	268,3	266,8	274,1	264,0
4	547,3	539,4	546,8	541,0	530,3
5	910,7	916,9	904,5	916,8	897,6
6	920,8	977,2	911,3	915,4	899,9
7	911,6	930,6	905,5	916,8	897,2
8	917,9	934,7	910,8	916,3	900,9
9	536,1	558,6	531,3	543,1	526,7
10	277,3	283,5	274,0	274,4	266,7
11	105,9	105,4	106,8	113,4	106,2
12	61,9	64,6	61,1	57,6	56,9

Table 2. *Time for the robot 's movement*

№	*t*, s				
	Algorithm 1	Algorithm 2		Algorithm 3	Algorithm 4
		$X = 100$	$X = 500$		
1	201,5	27,0	36,8	23,9	4,5
2	249,5	38,6	68,7	41,8	6,8
3	297,6	46,6	108,9	61,8	7,3
4	347,5	58,0	144,4	82,9	9,0
5	384,5	66,3	172,8	99,7	9,4
6	403,0	66,5	173,1	100,5	10,1
7	453,4	67,0	175,0	98,4	8,8
8	461,9	66,4	174,8	100,5	9,5
9	683,3	57,4	144,4	82,1	8,0
10	486,2	47,4	102,8	63,8	7,6
11	473,7	38,7	66,0	44,8	5,8
12	399,8	27,1	36,5	26,5	4,8

of hypotheses k = 12. Here four approximation algorithms of localization of the mobile robot are specified № - hypothesis number (indexing on 6 "steps" from left to right and from top to down), d - length of the way passed by the robot at localization, and t - an algorithm operating time. In randomized algorithm 2 there are 2 variants for an amount of casual points X = 100 and X = 500 have been used.

For comparison of outcomes of experiment we will define averages on the fixed values d and t. For example, $\bar{d} = \frac{1}{12}\sum_{i=1}^{12} d_i$ and $\bar{t} = \frac{1}{12}\sum_{i=1}^{12} t_i$.

On fig. 6 the graph of average length of a way on hypotheses as functions from a map size n.

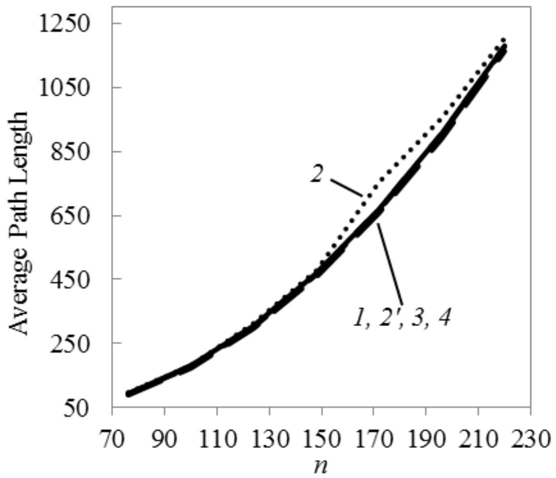

Figure 6. *Average path length*

Various types of lines of graphs on fig. 6-7, correspond to algorithms. 1 - to algorithm 1; 2 - to algorithm 2 at X = 100; 2 ′ - to algorithm 2 at X = 500; 3 - to algorithm 3; 4 – to algorithm 4.

On fig. 7 the graph of an average operating time $\bar{t}$ of algorithm as functions from a map size n is reduced. On graphs for instructions of algorithms the same markers, as on fig. 6-7 are used.

The comparative analysis of the data, corresponding fig. 6-7 shows the following.

1. Values of an average way for all four algorithms are very close. For example, for a case n = 148, corresponding to tab. 1 and tab. 2, we have:

$$\bar{d}_1 = 469{,}16;\ \bar{d}_2/\bar{d}_1 = 1{,}021;\ \bar{d}_{2'}/\bar{d}_1 = 0{,}99;$$
$$\bar{d}_3/\bar{d}_1 = 1{,}001;\ \bar{d}_4/\bar{d}_1 = 0{,}97;$$

Where inferior indexes correspond to numbers of algorithms, and for algorithm 2 the index 2 is used at X = 100 and an index 2′ at X = 500.

2. The biggest time shows algorithm 1. At n = 148, for example,

$$\bar{t}_1 = 403{,}49\text{ s};\ \bar{t}_2/\bar{t}_1 = 0{,}12;\ \bar{t}_{2'}/\bar{t}_1 = 0{,}29;$$
$$\bar{t}_3/\bar{t}_1 = 0{,}17;\ \bar{t}_4/\bar{t}_1 = 0{,}02;$$

I.e. algorithms 2, 3 and 4 are fulfilled essentially faster. And a variant windows algorithm has got the minimum time.

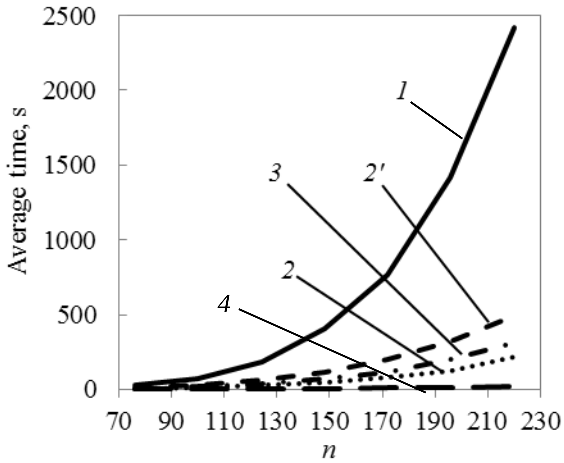

Figure 7. *Average operating time*

4. Conclusions

The made experiments have confirmed that fact that considered algorithms are fulfilled for rather notable time. Application of algorithm 1 for an operative solution of a problem of localization of the robot at great values n, probably, is inexpedient. Algorithms 2, 3 and 4 work on an order faster. Thus in the considered experimental situations the offered triangulation algorithm and windows algorithm ensure accuracy, comparable with other algorithms, and have good (and in some situations the best) temporary performances.

Therefore it is supposed to concentrate the further researches to effective realization of these algorithms (randomized, triangulation and windows) and their modifications:

1) At the expense of use of effective algorithms of a triangulation [6], [7], [8] that will allow to solve effectively as a problem of determination of intersection of simple polygons for each pair of hypotheses, and evaluations of the shortest ways in a polygon;

2) At the expense of working out of realizations of algorithms on parallel structures, first of all with the use of graphic accelerators and technology CUDA [8], [9].

REFERENCES

[1] Guibas L. J., Motwani R., Raghavan P. The robot localization problem, *SIAM J. Comput.1997*. Vol.26. P. 1120-1138.

[2] Dudek G., Romanik K., Whitesides S., Localizing a robot with minimum travel, *SIAM J. Comput.1998*, Vol 27. P. 583-604.

[3] Rao M., Dudek G., Whitesides S., Randomized algorithms for minimum distance localization, *Internat, J. Robotics Research*, 2007, Vol 26. P. 917-934.

[4] Koenig S., Mitchell J. S. B., Mudgal A., Tovey C. A near-tight approximation algorithm for the robot localization problem, *SIAM J. Comput. 2009*, Vol 39. P. 461-490.

[5] Дао Зуй Нам. Сравнительный анализ алгоритмов локализации мобильного робота, использующего карту// 66-я научно-техническая конференция профессорско-преподавательского состава университета, СПБЭТУ «ЛЭТИ» 2013, Сборник докладов студентов, аспирантов и молодых ученых, 1-8 февраля 2013,г. Санкт Петербург. С. 112-115.

[6] Скворцов А.В. Построение объединения, пересечения и разности произвольных многоугольников в среднем за линейное время с помощью триангуляции, *Вычислительные методы и программирование*, 2002, Т. 3. С. 116-123.

[7] Seidel R. A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons// Computational Geometry, Theory Application, 1991, Vol 1(1). P. 51-54.

[8] Qi M., Cao T.-T., Tan T.-S. Computing 2D Constrained Delaunay Triangulation Using Graphics Hardware, School of Computing; National University of Singapore, *Technical Report* # TRB3/11, March 2011. 9 с.

[9] Боресков А.В., Харламов А.А., *Основы работы с технологией CUDA*.- М.: ДМК-Пресс, 2010. 232.с

(The Board of Editors received the paper on 07/12/2013,
its review was completed on 29/12/2013)