Absolute Localization of a Mobile Robot with an Omnidirectional Vision Sensor

Young Jin Lee Myung Jin Chung

Department of Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology
373-1 Kusong-dong, Yusong-gu, Taejon 305-701, Korea
lyj@cheonji.kaist.ac.kr mjchung@ee.kaist.ac.kr

Abstract

Our research addresses how to find a pose of a mobile robot in an indoor environment using only angular information of the sensed data. An omnidirectional vision sensor is used to get azimuth angles of the features in the environment. The matching problem between observed angular data and given landmarks in the map is solved by Interpretation Tree (IT) search method. We propose a simple and effective pruning rule to reduce the search space in IT. The primary advantage of the proposed method is that the localization is accomplished simultaneously in the matching phase. The localization algorithm and the feature matching method are presented and simulation results are added to show that considerable reduction of the search space can be achieved by proposed method.

1. Introduction

Localization is one of the main research issues for the implementation of an autonomous mobile robot. It is a process of realizing where the robot is in the environment, based on the sensor information and a prior knowledge like a map [1]. Numerous techniques have been developed to solve this problem. In many cases, they predict the pose of a robot using internal sensors like an odometer and then adjust the pose using information from the external sensors like sonar sensors, vision sensors or lasers to compensate possible accumulative errors of the internal sensors [1],[2],[3],[4]. So if the initial pose of the robot is not given, or if the robot gets lost in the map by some reasons, the prediction and correction scheme cannot proceed. In such a case, absolute localization is needed, in which only external sensors are used to localize the robot.

Absolute localization is similar to a recognition problem in that we should find which part of the map is best fit to the local information from the sensors. Whole features in the map should be compared to the currently observed features to find correspondence between them. In this paper, the correspondence problem is solved by Interpretation Tree (IT) search method. We propose a simple and effective pruning rule to reduce the search space in IT. The primary advantage of the proposed method is that the localization is accomplished simultaneously in the matching phase.

Our omnidirectional vision sensor consists of a CCD camera in a glass tube and a conic mirror aligned with the optical axis of the camera. Parallel lines to optical axis are projected into radial lines in the image [5],[6]. Therefore, angular measurements of vertical edges can be easily obtained in an indoor environment. Moreover, an omnidirectional vision sensor has some advantages over ordinary perspective camera system. The one is the wide angle of view, i.e. 360 degrees. It contributes to the accuracy of localization because narrow view angle makes estimation error to be biased in one direction. The second is that it has no need for complicated camera calibration. We just need to align the optical axis of a camera with the axis of conical mirror. And aspect ratio can be found easily by measuring ellipticity of the mirror in the image. From these reasons, there have been increasing use of omni-directional vision sensor for localization, and navigation of a mobile robot [5],[6],[7],[8]. But there has been no consideration about absolute localization except in [8]. We describe absolute localization technique in a given map with angular information of observed features.

This paper is organized as follows. In section 2, a pose estimation algorithm using angular information is presented. And Interpretation Tree (IT) search algorithm and pruning rule to find correspondences are proposed in section 3. Section 4 shows simulation results to verify the validity of our method.

2. Pose estimation algorithm

This section shows how to find a pose in a given map of a mobile robot that can observe only angular information of the features (i.e. landmarks) in the environment. To make the problem simple, we assume that correspondence problem can be solved. That is, for each observed landmarks, the corresponding landmarks on the map can be identified. In fact, the matching is carried out by Interpretation Tree search in the next section.

The problem to be solved in this section can be stated as follows. "Given current sensor measurements $\theta_0^{(R)}, \theta_1^{(R)}, \ldots, \theta_N^{(R)}$, and corresponding landmark positions $L_0^{(w)}, L_1^{(w)}, \ldots, L_N^{(w)}$ on the map, find robot position $R^{(w)}$ and orientation $\theta_R^{(w)}$ "(Fig. 1). The superscripts represent reference frames.

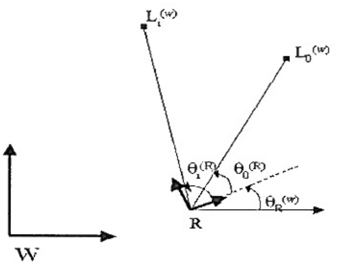


Fig. 1. Coordinates of the robot and landmarks

In [9], Betke proposed a linear optimization method to solve that problem in case more than three landmarks are observed. Complex number representation of a 2D vector made nonlinear problem to be linear. In our work, some modifications are made to reduce computation time. By changing the cost function, the calculation of an inverse of an N by N matrix can be replaced by that of a 2 by 2 matrix, where (N+1) is the number of landmarks. We describe the procedure of finding robot position $R^{(w)}$ and orientation $\theta_R^{(w)}$ with given correspondence between observed angles and those of landmarks on the map.

$$a_i = vec\left\langle \frac{e^{j(\theta_i - \theta_0)}}{comp\left\langle L_i - L_0 \right\rangle} \right\rangle \quad (1)$$

$$b_i = vec\left\langle \frac{1}{comp\left\langle L_i - L_0 \right\rangle} \right\rangle \quad (2)$$

where $vec\langle\cdot\rangle$ transforms a complex number to a 2D vector and $comp\langle\cdot\rangle$ transforms a vector to a complex number. And $L_i = \left[L_{ix}, L_{iy}\right]^T$ is a position of a landmark in a given map. θ_i's are observed azimuth angles from robot coordinate. So a_i and b_i in above equations are 2 by 1 vectors.

$$P_i = \frac{a_i a_i^T}{a_i^T a_i} - I_{2\times2} \quad (3)$$

$$f = -\left[\sum_{i=1}^{N} P_i\right]^{-1} \cdot \sum_{i=1}^{N} (P_i \cdot b_i) \quad (4)$$

Then, we can find the final solution as follows.

$$R^{(w)} = L_0 - \frac{1}{\|f\|^2} \begin{bmatrix} f_x \\ -f_y \end{bmatrix} \quad (5)$$

$$\theta_R^{(w)} = \tan^{-1}\left(\frac{L_{0y} - R_y}{L_{0x} - R_x}\right) - \theta_0 \quad (6)$$

In order to derive (1)~(6), we used following relations in Fig.1. [9].

$$\frac{1}{L_0 - R} = \left(\frac{|L_i - R|}{|L_0 - R|} e^{j(\theta_i - \theta_0)} - 1\right) \frac{1}{L_i - L_0} \quad (7)$$

We omit $comp\langle\cdot\rangle$ for convenience. By applying (7) for $i = 1, 2, \ldots, N$, (i.e. for more landmarks), we get

$$\begin{aligned} \frac{1}{L_0 - R} &= \left(\frac{|L_1 - R|}{|L_0 - R|} e^{j(\theta_1 - \theta_0)} - 1\right) \frac{1}{L_1 - L_0} \\ &= \left(\frac{|L_2 - R|}{|L_0 - R|} e^{j(\theta_2 - \theta_0)} - 1\right) \frac{1}{L_2 - L_0} \\ &\vdots \\ &= \left(\frac{|L_N - R|}{|L_0 - R|} e^{j(\theta_N - \theta_0)} - 1\right) \frac{1}{L_N - L_0} \end{aligned} \quad (8)$$

If we define f as

$$f = \frac{1}{L_0 - R} \quad (9)$$

and a_i and b_i as (1) and (2), (8) can be rewritten as

$$\begin{aligned} f &= r_1 a_1 - b_1 \\ &= r_2 a_2 - b_2 \\ &\vdots \\ &= r_N a_N - b_N \end{aligned} \tag{10}$$

with unknown f and r_i's. Thus, we only have to find such f and r_i's that satisfy (10). This can be solved by linear least squares method. Finding f and r_i's that minimize

$$\sum_{i=1}^{N} \left\| f - (r_i a_i - b_i) \right\|^2 \tag{11}$$

results in (4). Details are listed in appendix. After getting f, R can be obtained from (9), which is restated as (5) in vector form. Then we can find the robot heading directly in Fig.1. as in (6).

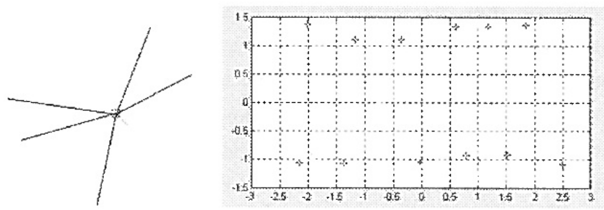
We need to inverse a 2 by 2 matrix in (4), so N must be greater than or equal to two. This means that at least three or more landmarks are required to solve above problem.

In this section, we formulated pose estimation algorithm, when the robot can identify corresponding landmarks on the map. Next section shows how to identify those landmarks.

3. Correspondence problem

The problem dealt with in this section is "given current sensor measurements $\theta_0^{(R)}, \theta_1^{(R)}, \ldots, \theta_K^{(R)}$, and known landmark positions $L_0^{(w)}, L_1^{(w)}, \ldots, L_N^{(w)}$ $(K < N)$ on the map, find the correspondence between θ_i's and L_i's ". The problem can be restated in Fig.2. "Find a robot pose of (a) in (b) such that all radial lines in (a) pass through some landmarks in (b)". We use Interpretation Tree (IT) search to tackle this problem. IT was used in [10] to identify an object in a scene and in [11] to localize a robot by sonar data. But localization using IT with angular measurements is a different problem that needs different constrains. Mouaddib used IT in [8] for localization using omnidirectional vision sensor. The subdivision of the robot's evolution field and verification of each divided rectangle were main idea in [8]. But IT search was needed for all the rectangles divided. And the final robot pose could be obtained only after correspondence problem is solved by IT search. In our method to be described in this section, only one IT search is sufficient. Moreover, robot position $R^{(w)}$ and orientation $\theta_R^{(w)}$ are obtained directly as a result of IT search.

IT search is based on hypothesis and verification. A path in a tree structure corresponds to a hypothesis or assumption. Some verification rules or pruning rules are required to find out whether the hypothesis is true or not. In our case, a complete path from starting node to the deepest one is a candidate of possible correspondence pairs (Fig.3). The tree has combinatorial number of paths according to the number of observations and landmarks on the map. We have to examine all the paths in a tree. In other words, we should verify all the hypotheses. But efficient pruning rule can reduce the search space greatly so that computational time can be saved.



(a) (b)

Fig. 2. Angular measurement data and landmark locations in a given map.

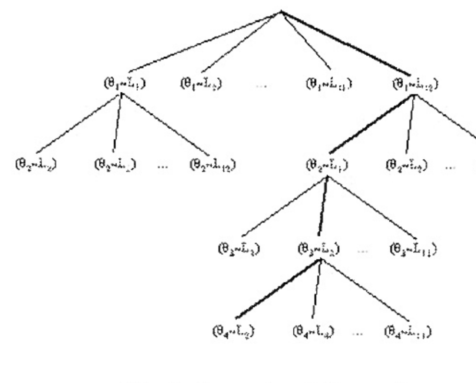


Fig. 3. Example of IT search

Now, we address our pruning rule to reduce search space. The basic idea is that if the hypothesis is correct, i.e. if all the pairings in a complete path have right correspondences, all the observation in that path should produce the same pose estimation.

We can find a pose of a robot using equations in section 2 if correspondences of more than three landmarks are given. Therefore, a pose can be estimated using the first three nodes in a path, and the 4th observation and one afterwards can be used to check if the estimated pose is consistent with these observations. If the estimated pose from the first three landmarks is not correct, 4th or others may not have corresponding landmarks in the map.

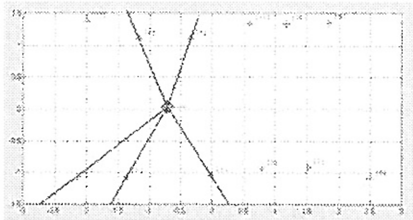
In Fig.3, not all the paths need to be searched to the deepest node. As we mentioned above, the 4th node can be used to determine whether to keep going to the next node or to stop searching in this branch. At the 4th node, we may have pose estimation from three nodes in the passed branch. Therefore we can predict observations of other landmarks based on estimated pose. If real observation at the 4th node is different from estimated angle in a degree beyond a certain threshold, we can conclude that the estimated pose is not consistent with other observations and that no more examination in this branch is required. Then we move to the next branch to examine the validity of hypothesis attached to that branch. In general, inconsistency is checked at the 4th node or at the subsequent nodes. But in some cases, stop condition occurs at the 3rd node, because certain improper candidate of 3 correspondences cannot yield pose solution. The proposed pruning rule is summarized as follows.

(i) If node-depth = 3, check pose solution existence.
(ii) If node-depth > 3, check
| predicted angle – measured angle| $\leq$ threshold

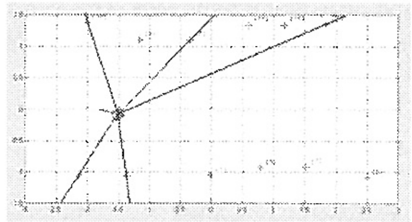
A validated node is used to estimate more accurate pose in the next step. If a node fails in the test above, discard branches below the node and move to next branch. If all the nodes in a complete path are verified to be valid, a consistent solution of robot pose is obtained. The solution may not be unique. The uniqueness is dependent on the map structure and the number of observations.

4. Simulation results

The proposed method is verified in this section by computer simulation. We constructed an artificial map with 12 landmarks as in Fig.2. (b). In real situation, a robot cannot observe all the landmarks in the map. So we have assumed that the robot can observe some of them at a given pose. We will show that our algorithm works well though partial information is given to the robot.



(a)



(b)

Fig. 4. Two solutions when 5 measurements are made

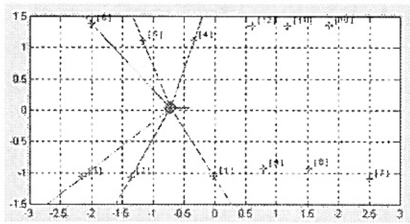


Fig. 5. A solution when 6 measurements are made.

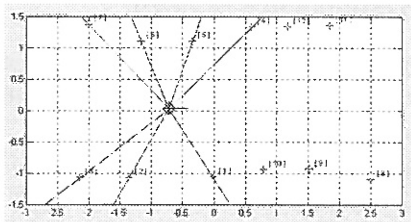


Fig. 6. A solution when 7 measurements are made.

In Fig. 4, (a) is a given pose and true solution, and (b) is another solution that the proposed algorithm found. In ideal case, if we set the threshold to be zero in pruning rule, (b) can be discarded. But in real situation, there can be many solutions if the number of observations is small and the map structure is simple. In the same map, a unique solution was obtained in Fig.5 in which six measurements are used and in Fig.6 where seven measurements are used.

Table 1 shows the variation of the search space and the number of solutions according to the number of observations. The effects of applied pruning rue are shown. The entries of the first column are numbers of observations at a given pose in a map with twelve landmarks. The second column shows total search space in IT. In the third column, we present the number of paths actually traced more than three nodes in depth by the algorithm. The total number of paths in a search space is increased as the robot observes more landmarks, but the number of examined paths has increased relatively small. It shows that a considerable reduction of the search space is achieved by the proposed pruning rule.

Number of observations	Total number of path in IT	Number of actually examined path	Number of solutions found
4	11880	3752	6
5	95040	4573	2
6	665280	4585	1
7	3991680	4591	1

Table. 1. Search space reduction

5. Conclusions

We presented in this paper an absolute localization technique of a mobile robot using omnidirectional vision sensor. The proposed algorithm uses Interpretation Tree search method to obtain correspondence between observed landmarks and models in the map. Localization and matching are accomplished simultaneously by our pruning rule that can reduce the search space considerably.

We will conclude this paper with some remarks. The search space can be more reduced if the ordering constraint is used. In our work, we have considered all possible permutations to make the tree structure. But the observed landmarks have a certain angular orders and the landmarks in the map have fixed positions. So, the impossible pairings can be removed in advance.

Appendix

We will derive here eq. (4). That is, it will be shown that 2x2 matrix inversion is sufficient to find the solution which minimizes eq. (11).

Eq. (10) can be rewritten as

$$\begin{aligned} r_1 a_1 - f &= b_1 \\ r_2 a_2 - f &= b_2 \\ &\vdots \\ r_N a_N - f &= b_N \end{aligned} \qquad (12)$$

In matrix form, eq. (12) becomes

$$\begin{bmatrix} a_1 & 0 & 0 & \cdots & 0 & -I_{2\times2} \\ 0 & a_2 & 0 & \cdots & 0 & -I_{2\times2} \\ & & & & \vdots & -I_{2\times2} \\ 0 & 0 & 0 & \cdots & a_N & -I_{2\times2} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \\ f \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \qquad (13)$$

We can find $X = [r_1 \quad r_2 \quad \cdots \quad r_N \quad f]^T$ by solving

$$\left(A^T A\right) X = A^T B \qquad (14)$$

where,

$$A = \begin{bmatrix} a_1 & 0 & 0 & \cdots & 0 & -I_{2\times2} \\ 0 & a_2 & 0 & \cdots & 0 & -I_{2\times2} \\ & & & & \vdots & -I_{2\times2} \\ 0 & 0 & 0 & \cdots & a_N & -I_{2\times2} \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \qquad (15)$$

Then, X is the solution which minimizes eq. (11).

Now, we can calculate $A^T A$ and $A^T B$ and they have simple forms.

$$A^T A = \begin{bmatrix} a_1^T a_1 & 0 & 0 & & 0 & -a_1^T \\ 0 & a_2^T a_2 & 0 & & 0 & -a_2^T \\ 0 & 0 & a_3^T a_3 & & 0 & -a_3^T \\ & & & \ddots & & \vdots \\ 0 & 0 & 0 & & a_N^T a_N & -a_N^T \\ -a_1 & -a_2 & -a_3 & \cdots & -a_N & N \cdot I_{2\times2} \end{bmatrix} \qquad (16)$$

$$A^T B = \begin{bmatrix} a_1^T b_1 \\ a_2^T b_2 \\ \vdots \\ a_N^T b_N \\ -\sum_{i=1}^{N} b_i \end{bmatrix} \qquad (17)$$

Thus, we can rewrite eq. (14) as follows.

$$\begin{aligned} &\left(a_1^T a\right) r_1 - a_1^T f = a_1^T b_1 \\ &\vdots \\ &\left(a_N^T a\right) r_N - a_N^T f = a_N^T b_N \\ &-a_1 r_1 - a_2 r_2 \cdots - a_N r_N + Nf = -\sum_{i=1}^{N} b_i \end{aligned} \tag{18}$$

Eq. (18) can be easily transformed into eq. (4) by substitution of r_i's.

References

[1] J. Leonard, H. F. Durrant-Whyte, "mobile robot localization by tracking geometric beacons", IEEE Trans. on Robotics and Automation, Vol. 7, pp. 376-382, 1991.

[2] Y. Watanabe, S. Yuta, "Position estimation of mobile robot with internal and external sensors using uncertainty technique", Proc. of IEEE Int. Conf. on Robotics and Automation, Vol. 3, pp. 2011-2016, 1990.

[3] K. Kyriakopoulos, A. Curran, "Sensor-based self-localization for wheeled mobile robotics," J. of Robot. Syst., Vol. 12, pp. 163-176, 1995.

[4] L. Delahoche, C. Pegard, E. M. Mouaddib, P. Vasseur, "Incremental map building for mobile robot navigation in an indoor environment", Proc. of IEEE Int. Conf. on Robotics and Automation, Vol. 3, pp. 2560-2565, 1998.

[5] Y. Yagi, S. Kawato, S. Tsuji, "Real-time omni-directional image sensor (COPIS) for vision-guided navigation", IEEE Trans. on Robotics and Automation, Vol. 10, pp. 11-22, 1994.

[6] L. Delahoche, C. Pegard, B. Marhic, P. Vasseur, "A navigation system based on an omnidirectional vision sensor", Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Vol. 2, pp. 718-724, 1997.

[7] Y. Matsumoto, K. Ikeda, M. Inaba, H. Inoue, "Visual navigation using omnidirectional view sequence", Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Vol. 1, pp. 317-322, 1999.

[8] E. M. Mouaddib, B. Marhic, "Geometrical matching for mobile robot localization", IEEE Trans. on Robotics and Automation, Vol. 16, pp. 542-552, 2000.

[9] M. Betke, L. Gurvits, "Mobile robot localization using landmarks", IEEE Trans. on Robotics and Automation, Vol. 13, pp. 251-263, 1997.

[10] O. Faugeras, "Three dimensional computer vision", The MIT Press, 1993

[11] M. Drumheller, "Mobile robot localization using sonar", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 9, pp. 325-332, 1987.