

Fuzzy Control Simultaneous Localization and Mapping Strategy Based on Iterative Closest Point and k -Dimensional Tree Algorithms

Jih-Gau Juang* and Jia-An Wang

National Taiwan Ocean University, 2 Pei-Ning Road, Keelung 20224, Taiwan

(Received July 2, 2014; accepted February 9, 2015)

Key words: SLAM, iterative closest point, laser sensor, k -D tree, mobile robot

In this study, we apply laser and infrared sensors to a wheeled mobile robot (WMR) for simultaneous localization and mapping (SLAM). The robot utilizes a laser measurement sensor to detect obstacles and identify unknown environments. Fuzzy theory and the iterative closest point (ICP) algorithm are applied to control design. The proposed control scheme can control the WMR movement along walls and avoid obstacles. In addition, the k -dimensional (k -D) tree is used to reduce the computation time and achieve real-time positioning. By calculating the rotation and translation matrices among different sets of measured points, distance and angle information of the moving robot can be recorded. Furthermore, the worst point rejection method is applied to delete less corresponding points that can prevent the ICP process convergence to a local optimum.

1. Introduction

The wheeled mobile robot (WMR) is the most used carrier in mobile robot applications. It has some good properties such as high-speed mobility, easy control, and energy storage capacity.^(1,2) Simultaneous self-positioning, environment map building, path planning, and obstacle avoidance are essential abilities for autonomous mobile robots. With the installation of various sensors or tools, mobile robots can be applied to multipurpose applications such as home services, medical care, entertainment, space exploration, military, and industrial automation. In this study, we focus on unknown environment exploration. Precise position estimation is one of the core issues in simultaneous localization and mapping (SLAM) research. In ref. 3, the sensor network provided an effective method for a mobile robot to adapt to changes and guided it across a geographical network area. To enhance the performance, a charge-coupled device camera and artificial landmarks were used for self-localization.⁽⁴⁾ Hwang and Song⁽⁵⁾

*Corresponding author: e-mail: jgjuang@ntou.edu.tw

examined the monocular-vision-based SLAM of a mobile robot using an upward-looking camera. Gallegos and Rives⁽⁶⁾ described a composite sensor approach that combined the information given by an omnidirectional camera and a laser range finder to efficiently solve the indoor SLAM problem. Most of the studies applied image processing to obtain feature points; the process was complex and time-consuming. Another drawback is that most visual sensors are sensitive to the light source, and the brightness will affect the result of image processing and its accuracy. The positioning systems currently known and widely used are laser scanning positioning systems, odometer calculation positioning systems, image video positioning systems, and ultrasonic sensing systems. In this study, the laser measurement sensor is used owing to its high accuracy, high receiving rate, and wide scanning range. Moreover, the laser is not affected by light and is very suitable for indoor usage. It has sufficient time for position calculation in real time using the iterative closest point (ICP) algorithm⁽⁷⁾ and achieves positioning in an unknown environment. Fuzzy theory is applied to control the movement of the WMR.

2. System Setup

A WMR called Ihomer⁽²⁾ is used in the entity test, as shown in Fig. 1(a). The control scheme mainly uses a laser measurement sensor SICK-LMS100, which is placed on top of the WMR, to detect the unknown environment around the WMR. Two wheels are located at the left and right sides under the body of the WMR and are driven by two 18 V DC motors. Two small casters are located at the rear and front under the body of the WMR. They support the balance and movement of the WMR. The encoders, which are located at the sides of the wheels, provide the measured value of the body movement. The receivers of the LMS100 measure the object distance and help collect information that is used for building the map and wall-following. Dynamic analysis of the moving distance and turning angle of the WMR can be found in ref. 4. The WMR is located on the Cartesian coordinate system (global coordinate system) with no lateral or sliding movement, as shown in Fig. 1(b).

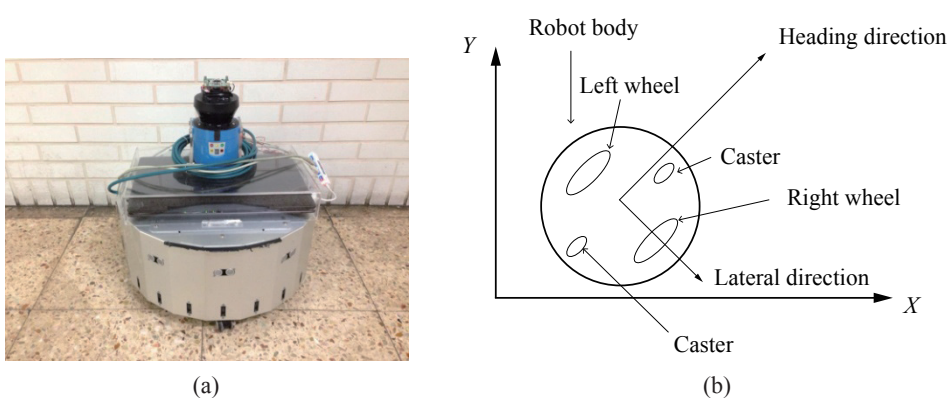


Fig. 1. (Color online) (a) Ihomer wheeled mobile robot with LMS100 laser sensor and (b) WMR coordinate diagram.

3. Control Design

3.1 Wall-following

There are two common inputs of the proposed fuzzy controller for speed and angle control, which are the detected distances dr and r , where r is $r = dfr \cos(45^\circ) - dr$, as shown in Fig. 2.

The input variable for the distance dr is SR_r , and the fuzzy values are very near, near, medium, far, and very far. Fuzzy values of the input r are large negative, negative, medium, positive, and large positive. The output variable for turning angle is TA, and the fuzzy values are TR3, TR2, TR1, TZ, TL1, TL2, and TL3, which represent the angles turn right very large, turn right large, turn right, go forward, turn left, turn left large, and turn left very large, respectively. Fuzzy speed is selected on the basis of distance factor. Some examples of the control rules for turning angle are shown in Table 1.

3.2 Improved simultaneous localization and mapping

When accessing the door of an unexplored room, the control system starts to record the coordinates and angles, which are calculated using the ICP algorithm. The ICP uses two different curves from LMS100 as inputs, and the outputs are rotation and translation matrices. The curves are sets of 180 points from 0 to 180°. The characteristics can be used to find the modified rotation and translation matrix to revise the imprecise data

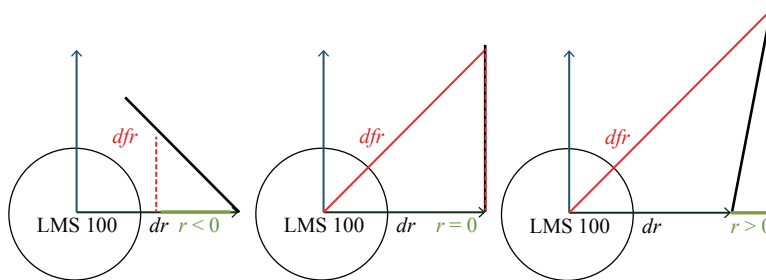


Fig. 2. (Color online) Schematic view of r .

Table 1
Fuzzy logic rule table.

r	LN	N	M	P	LP
SR_r					
VN	TL3	TL1	TZ	TZ	TZ
N	TL2	TR1	TZ	TZ	TZ
M	TL2	TL1	TZ	TZ	TZ
F	TL2	TL1	TZ	TR1	TR1
VF	TL2	TL1	TZ	TR2	TR3

from the robot encoder. There are three disadvantages of the ICP algorithm. The first one is that the computation time of the ICP algorithm is very long to be used in real time. To reduce the computation time in the process and achieve real-time positioning, the k -dimensional (k -D) tree⁽⁸⁾ is applied to this study. The second disadvantage is the accumulated error. When the turning angle and environment changing rate are both small, a strategy of reducing the number of reads⁽⁹⁾ is applied to obtain the rotation angle and displacement. The third problem is that the ICP algorithm usually converges to a local optimum. To avoid this situation, in this study, we apply strategies of increasing the initial angle and worst point rejection⁽¹⁰⁾ to map building when the turning angle of the WMR and the environment changing rate are both large.

3.2.1 Iterative closest point algorithm

The ICP is often used to reconstruct 2D or 3D surfaces from different scans. It iteratively revises the translation and rotation, and minimizes the distance between the points of two different curves. In general, the ICP uses two different curves as inputs, and the outputs are rotation and translation matrices. The characteristics can be used to find the modified rotation and translation matrix to revise the incorrect data from the encoder of the robot. The ICP algorithm is described as follows.⁽⁷⁾

Step 1: Define the data model of data shape (\mathbf{P}) and model shape (\mathbf{X}) as

$$\mathbf{P}_i = \begin{bmatrix} P_{x_i} \\ P_{y_i} \end{bmatrix}_{i=1 \cdots n} \quad \mathbf{X}_i = \begin{bmatrix} X_{x_i} \\ X_{y_i} \end{bmatrix}_{i=1 \cdots k} \quad (1)$$

Here, n and k are the total numbers of data. The data measured by the LMS100 is a set of distance r and angle θ . Record a distance every 0.5° from -45 to 225° . At each iteration, 180 points are retrieved, from 0 to 180° . Then, use eq. (2) to translate polar coordinates to Cartesian coordinates.

$$\begin{bmatrix} DX_{x_i} \\ DX_{y_i} \end{bmatrix} = \begin{bmatrix} r_i \cos \theta_i \\ r_i \sin \theta_i \end{bmatrix} \quad (2)$$

After data processing, set a threshold value, which is the stopping condition of the ICP. If the mean square error is greater than the threshold value, then continue to repeat the process at each iteration. The threshold is the mean square error calculated using \mathbf{X} and a new data shape (updated \mathbf{P}). When the process achieves convergence and the mean square error is less than the threshold, stop the calculation process.

Step 2: After initial setting, find the correspondence between two sets of \mathbf{P} and \mathbf{X} . The corresponding relationship is the shortest distance between corresponding points in the ICP algorithm. Find the minimum distance from the point p_i on \mathbf{X} . The corresponding point a_i is stored in \mathbf{A} shape. The corresponding relationship is calculated as $d \min_i = \min d(p_i, \mathbf{X})_{i=1, \dots, n}$.

Step 3: The geometric transformation matrix consists of two types of rotation and translation matrices. First, calculate the center of mass. p_i and a_i are points of \mathbf{P} and \mathbf{A} ,

respectively. N_p and N_a are numbers of \mathbf{P} and \mathbf{A} , respectively. By using the following equations, the rotation angle and displacement between the two data can be obtained as

$$\boldsymbol{\mu}_p = \frac{1}{N_p} \sum_{i=1}^{N_p} \mathbf{p}_i, \boldsymbol{\mu}_a = \frac{1}{N_a} \sum_{i=1}^{N_a} \mathbf{a}_i. \quad (3)$$

With the center of mass, calculate the cross covariance matrix between \mathbf{P} and \mathbf{A} .

$$\boldsymbol{\Sigma}_{pa} = \frac{1}{N_p} \sum_{i=1}^{N_p} [(\mathbf{p}_i - \boldsymbol{\mu}_p)(\mathbf{a}_i - \boldsymbol{\mu}_a)^T] = \frac{1}{N_p} \sum_{i=1}^{N_p} [\mathbf{p}_i \mathbf{a}_i^T] - \boldsymbol{\mu}_p \boldsymbol{\mu}_a^T \quad (4)$$

The cyclic components of the antisymmetric matrix $\mathbf{B}_{ij} = [\boldsymbol{\Sigma}_{pa} - (\boldsymbol{\Sigma}_{pa})^T]_{ij}$ are used to form the column vector $\Delta = [B_{23} \ B_{31} \ B_{12}]^T$. This vector is then used to form the 4×4 matrix.

$$\mathbf{Q}(\boldsymbol{\Sigma}_{pa}) = \begin{bmatrix} \text{tr}(\boldsymbol{\Sigma}_{pa}) & & & \\ & \Delta^T & & \\ & \Delta & \boldsymbol{\Sigma}_{pa} + (\boldsymbol{\Sigma}_{pa})^T - \text{tr}(\boldsymbol{\Sigma}_{pa})\mathbf{I}_3 & \\ & & & \end{bmatrix} \quad (5)$$

Here, \mathbf{I}_3 is the 3×3 identity matrix. The unit eigenvector $\mathbf{q} = [q_0 \ q_1 \ q_2 \ q_3]^T$ corresponding to the maximum eigenvalue of the matrix $\mathbf{Q}(\boldsymbol{\Sigma}_{pa})$ is selected as the optimal rotation.

$$\mathbf{R} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 + q_3^2 - q_2^2 - q_1^2 \end{bmatrix} \quad (6)$$

The optimal translation vector is given by

$$\mathbf{T} = \boldsymbol{\mu}_a - \mathbf{R}(\mathbf{q}_R)\boldsymbol{\mu}_p. \quad (7)$$

Step 4:

\mathbf{D} is the data shape (\mathbf{P}) and m is the number of iterations.

$$\mathbf{D}_{m+1} = \mathbf{D}_m \cdot \mathbf{R} + \mathbf{T} \quad (8)$$

The changes in displacement and angle at each iteration can be obtained. Then, add the changes in displacement and angle individually. When the process achieves convergence and the mean square error is less than the threshold, the final displacement and angle are the displacement and angle between two different curves.

3.2.2 *k-D tree*

One of the most well-known data structures suitable for the ICP algorithm is the *k-D tree*.⁽⁸⁾ The *k-D tree* can reduce the computation time and find the significant corresponding point. It is a space-partitioning data structure for organizing points in a *k-D* space. The *k-D tree* is a binary tree in which every node is a *k-D* point. Every nonleaf node can be considered as implicitly generating a splitting hyperplane that divides the space into two parts, known as half-spaces. Points to the left of this hyperplane are represented by the left subtree of that node, and points to the right of the hyperplane are represented by the right subtree. The hyperplane direction is chosen in the following manner: every node in the tree is associated with one of the *k* dimensions, with the hyperplane perpendicular to the axis of that dimension. Two experiments are carried out to compare the computation time. The results are shown in Table 2. The computation time obtained by using the *k-D tree* with ICP is shorter than that without the *k-D tree*. Its computation time is less than the SLAM's sampling time of 0.5 s; thus, real-time positioning can be achieved.

3.2.3 Reducing the number of reading cycles

Reducing the number of reading cycles⁽⁹⁾ means a smaller number of alignments and less accumulated error. Consecutive data is not used since 1 s of computation time cannot tell the difference between the original ICP and the less-reading-cycle ICP. Here, a 20-s test is performed. Let the WMR continuously move straight forward for 20 s. The moving distance of the WMR measured is 197.82 cm, and that calculated using the less-reading-cycle ICP is 196.12 cm, as shown in Fig. 3, where the red dotted line is the track of the WMR, the blue dotted lines are the calculated results, and the solid black lines are the actual walls.

3.2.4 Worst point rejection

When the WMR turns at a large rotation angle, delete the points that do not correspond to the model shape.⁽¹⁰⁾ Points that do not correspond make the ICP process converge to a local optimum. Thus, the predicted angle is used. The predicted angle is in accordance with the turning angle, and then carry out some fine tuning to fit the actual condition. The deleted points are according to the predicted angle and one point at each 0.5°. For example, if there are 20 points at 10°, we delete some points at the tail of the data shape and the head of the model shape. Figure 4 shows the comparison of the original ICP and the ICP with the worst point rejection. In the test, the WMR, i.e., the red dot on the map, turns left at 90°.

Table 2
Computation times of two experiments.

Method	Closest point	Overall computation time
ICP	1.19 s	1.33 s
ICP with <i>k-D tree</i>	0.13 s	0.38 s

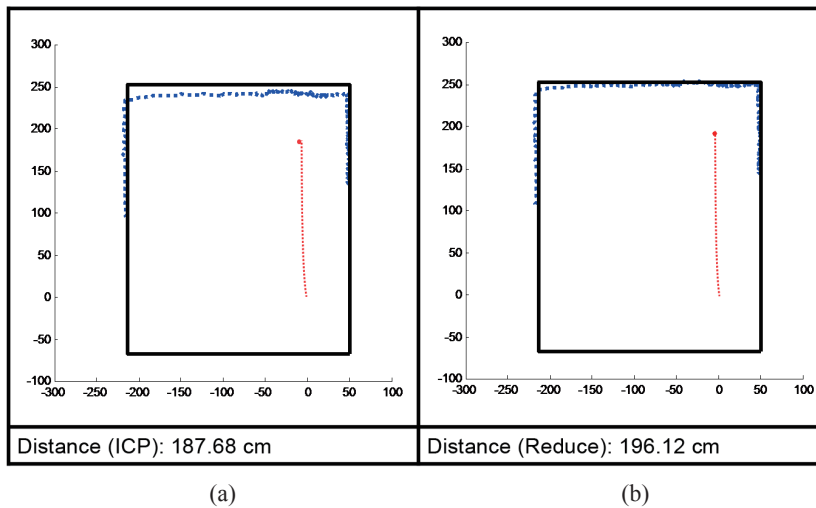


Fig. 3. (Color online) Results of using (a) original ICP and (b) less-reading-cycle ICP.

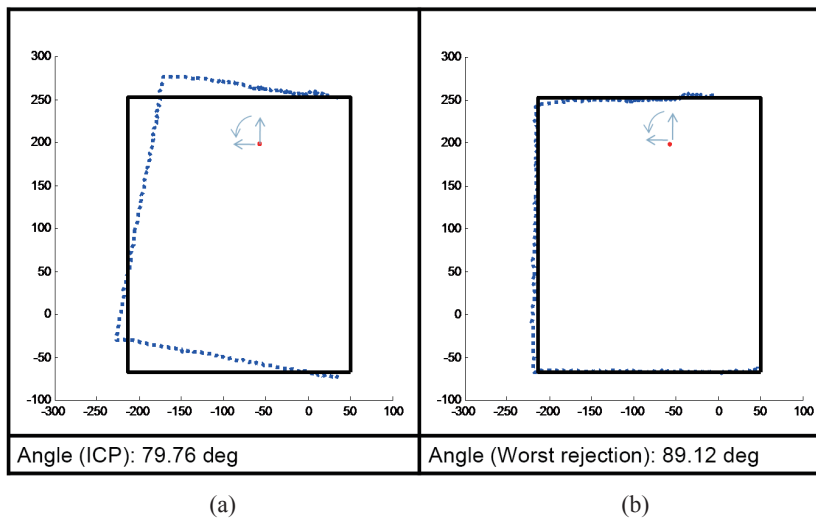


Fig. 4. (Color online) Results of (a) ICP and (b) ICP with worst point rejection.

4. Experimental Results

The improved SLAM strategy was tested in different environments. One of the experimental results is shown in Fig. 5. The left part of the figure is the real situation of the WMR and the right part is the map of the scanned environmental data and the path calculated using the ICP algorithm, where the red lines are the WMR's moving path, and the yellow points are environmental data. Without the weighting process, the map can be obtained as the dashed line in Fig. 6(a). Compared with the real contour of the unknown



Fig. 5. (Color online) Unknown environment exploration.

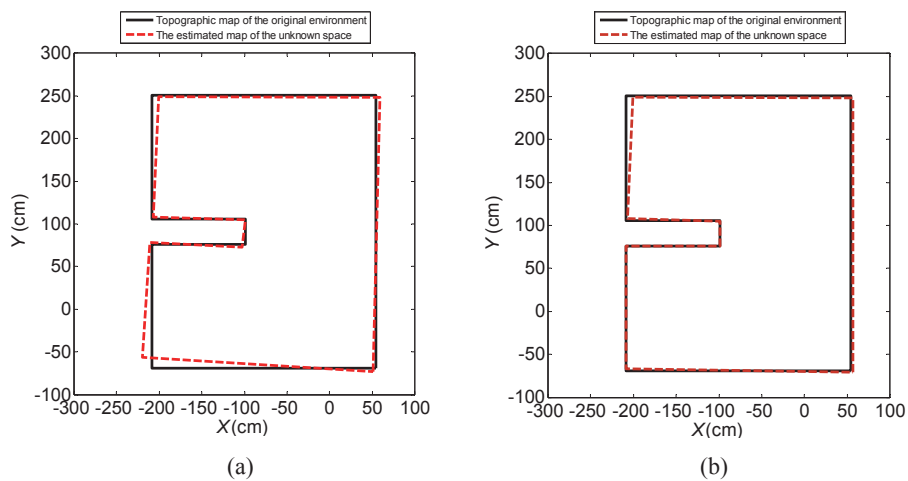


Fig. 6. (Color online) Results of curve fitting (a) without and (b) with weighting process.

environment, the average error is 3.8 cm. Furthermore, with the weighting process that gives more weight to the old data and the weights are descending with the new data, the result is shown in Fig. 6(b). The average error decreases to 1.6 cm. In this experiment, the WMR took 2 min and 6 s to complete the exploration using the improved SLAM strategy.

5. Conclusions

In this study, a control scheme based on an improved SLAM strategy and fuzzy logic theory is proposed to control a mobile robot for building an unknown environment map. In this research, we apply a laser measurement sensor to mark obstacles and positions. According to the laser measurement sensor's location and the detected distance between the sensor and the object, map building can be completed after the environmental

exploration is finished. The task of building an unknown environment map can be accomplished without using visual images. In previous studies, the SLAM based on visual sensors is sensitive to the light source that affects the accuracy of the image process. In this study, the laser sensor is not affected by light; even in a dark room, it still works properly. Real-time PC-based control of the WMR on wall-following and map building is performed successfully. Moreover, in this study, a modified ICP algorithm is presented. It uses the k -D tree to reduce the computation time, less-reading-cycle ICP to reduce the number of alignments and the accumulated error, and the worst point rejection method to delete less corresponding points and prevent the ICP process convergence to a local optimum. Experiments show that the average error is less than 2 cm, which is better than the results of most of the previous studies.

References

- 1 H. J. Kwak and G. T. Park: *Int. J. Eng. Technol. Innovation* **2** (2012) 97.
- 2 J. G. Juang, C. L. Yu, C. M. Lin, R. G. Yeh and I. J. Rudas: *Acta Polytech. Hung.* **10** (2013) 5.
- 3 M. L. Corradini, T. Leo and G. Orlando: *J. Rob. Sys.* **19** (2002) 177.
- 4 Y. G. Kim, J. An and K. D. Lee: *Int. J. Adv. Rob. Sys.* **8** (2011) 203.
- 5 S. Y. Hwang and J. B. Song: *IEEE Trans. Ind. Electron.* **58** (2011) 4804.
- 6 G. Gallegos and P. Rives: *Proc. IEEE Int. Conf. Rob. Autom.* (2010) p. 3519.
- 7 K. Pathak, A. Birk, N. Vaškevičius and J. Poppinga: *IEEE Trans. Rob.* **26** (2010) 424.
- 8 K. Zhou, Q. Hou, R. Wang and B. Guo: *ACM Trans. Graphics* **27** (2008) Article 126.
- 9 Y. C. Lin: Master Thesis, National University of Kaohsiung (2011).
- 10 K. Pulli: *Proc. Int. Conf. 3-D Digital Imaging Modeling* (1999) p. 160.