

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

**ANSI***net*

Asian Network for Scientific Information  
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

## A Simultaneous Localization and Mapping Method Based on Fast-Hough Transform

Wu Zu Yu, Huang Xin Han, Li Xin de, Wang Min and Yan Huai Cheng  
Department of Control Science and Control Engineering,  
Huazhong University of Science and Technology, Wuhan, Hubei, 430074, China

**Abstract:** A Fast Hough transform method is proposed to perform mobile robot's Simultaneous Localization and Mapping (SLAM) under the strategy step by step. The laser range finder is used to detect the operating environment, where three important loops are included. First, though the laser data is very precise, its uncertainty still exists, i.e., the detecting points always are around the real points. Then the laser points are clustered according to a distance principle, the slope between two points and then apply the Hough transform method within sub-class to extract the straight line feature. Second, the mergence of line segments is needed by justify the similarity of two straight lines. Third, find out the difference between newly detected lines and reference lines by considering the similarity principle and the robot pose error can be obtained to calibrate the robot. So the SLAM can be implemented through the three loops finally, Experiment results show the validity of present method.

**Key words:** Mobile robot, laser, Fast Hough transform, SLAM

### INTRODUCTION

With the rapid development of intelligent mobile robots, their applications are more and more extensive, for example, robot rovers have been developed for planetary and space exploration, autonomous submersibles have been devoted to submarines prospecting and surveying, mobile robots have been developed for automation purposes or for operation in hazardous mining environments, nuclear facilities and eliminating-explosion spot and so on. One of the most difficult tasks of mobile robot is to determine its pose (position and direction), this is because odometer can accumulate error without bound and the robot cannot get an accuracy pose. So a reliable and stable method of self-localization is needed. In the past decade many effective methods have been proposed, such as Markov localization (Fox *et al.*, 1998), Monte Carlo Localization (MCL) (Thrun *et al.*, 2001), scan match (Fox, 2003), particle filter (Zang and Yuan, 2006), Kalman Filter (KF) (Kwon *et al.*, 2006) and Extended Kalman Filter (EKF) (Lee *et al.*, 2006) etc. However, they also have many shortcomings, for example, Markov-based method owns greater computation amount, Kalman filter based method can only deal with linear problem etc. Because Map building and localization are tightly associated, someone compares their relation as egg and chicken. Therefore, Simultaneous Localization and Map Building (SLAM) is very challenging, as the application environment is becoming more and more complicate and non-structured. Considering the greater computation

amount and less precision of SLAM based on Classical Hough transform, in this study, a new SLAM method based on fast Hough transform is proposed.

### PRINCIPLE OF FAST HOUGH TRANSFORM

**Principle of Hough transform:** A method to extract line feature in point set. In general, the line equation is presented in polar coordinate (Dudar and Hart, 1972). A line in image space is mapped to parameter space. If  $n$  pixels in image are on a line, then the corresponding  $n$  sine curves in Hough space cross through a common point  $(\rho, \theta)$ . The line equation in polar coordinate:

$$\rho = x \cos \theta + y \sin \theta \quad (1)$$

Where:

- $\rho$  = Distance from the origin to the line.
- $\theta$  = Angle from positive axis to the line.
- $(x, y)$  = One point on the line.

The main principle which involved in Hough transform involved shown in Fig. 1.

**Improved Fast Hough transform:** The original Hough transform suffers from huge computation amount. It does not satisfy the real time requirement in self-localization due to the high amount of computation. We here put forward an improved Fast Hough transformation.

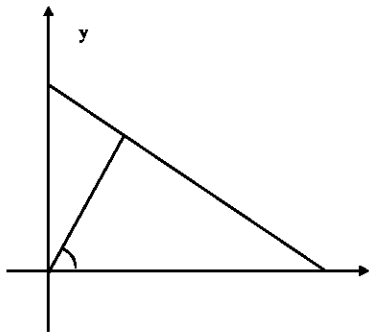


Fig. 1: Principle of Hough transform

The basic principle is when mobile robot moves to a certain position, we cluster all scanned points in the detecting range of laser range finder (0~180 degrees) and classify them into subset  $S_i$  ( $i \in 1, 2, \dots$ ). Supposed that there are  $n$  points in the subset  $S_i$ , in order to extract the refined feature, so it is necessary to extend some points between two points in sequence and then compute respectively the slope between every two points. Setting up a counter, if the slope is in the setting range, we add one on the counter. When finished the statistic of all points in the subset  $S_i$ , we classify all points having the maximum number to support a certain slope into a new subset  $S_i^a$ . We can pick up the straight line  $l_{ij, i \in \{1, 2, \dots\}}$  through Hough transformation and then find those points owning the sub maximum number and put them into another subset  $S_i^b$ . Also pick up the second straight line until the number to support a certain slope is lower than the threshold. The mentioned above is the key difference from other Fast transformation methods. Then we begin to deal with another subset  $S_j$  by same way until we finish all points scanned in the detecting range of laser. By doing this we not only reduce the complexity and amount of computation, but also improve the precision of picking up straight line, as well as find smaller features which might be omitted by other Hough transformation methods.

**REALIZATION OF MOBILE ROBOT SLAM**

**Feature extraction process:** Clustering all the points according to adjacent points distance by scanning the environment in the Fig. 2. Because the scan points are ordered clockwise and expressed by  $p\{(x_i, y_i) | i = 1: n\}$ , i.e., A, B, C, D, E, F, L, M... in Fig. 3. If the distance between point  $p_{i+1}$  and point  $p_i$  is bigger than a threshold and then  $p_{i+1}$  belongs to the other class, i.e., the distance between B and C is bigger than the threshold, then point B belongs to class 1 and point C belongs to class 2 in

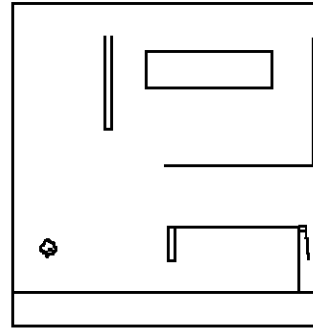


Fig. 2: The original environment

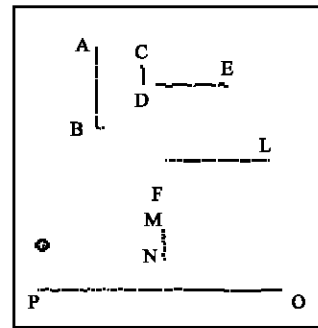


Fig. 3: Extend laser scan points

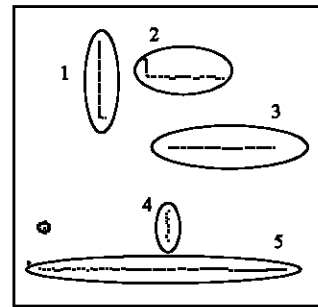


Fig. 4: Cluster all points into 5 classes

Fig. 4. In this way the scan points are clustered into  $S_i$  classes. Then we can pick up all feature lines according to the principle of fast Hough transform. Then we can get outline the pick-up features of Fig. 5.

**Match of feature lines:** In order to compare lines in the course of self calibration we define some attributes about a line, i.e., Line:  $(\rho_i, \theta, n, l, sp, ep, dir, d)$  shown in Fig. 6. Where,  $\rho_i$  is the distance OP between the origin O and the feature straight line AB.  $\theta$  is the angle of incline between the perpendicular OP and the positive direction of x axis (0~360°).  $n$  is the point number including all points which

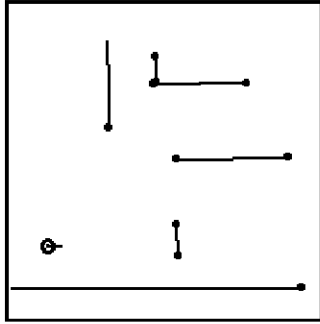


Fig. 5: Pick-up features of outline

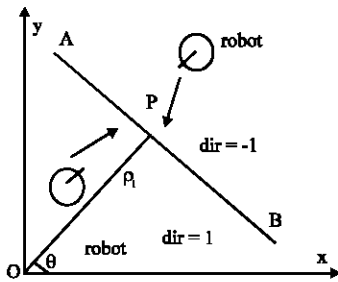


Fig. 6: The sketch of observation direction for feature line

may be fitted into a feature straight line.  $sp$  is the starting point of feature straight line  $AB$ .  $ep$  is the endpoint.  $dir$  is the observation direction relying on the location between robot and feature straight line  $AB$  i.e.,  $dir = \text{sign}(\rho - x \cos(\theta) - y \sin(\theta))$ , where,  $\text{sign}$  is the sign function. If robot is nearer the origin than the feature straight line, then  $dir = 1$ ; if robot is farther away from the origin than the feature straight line  $AB$ , then  $dir = -1$ ; if robot lied on the feature straight line, then  $dir = 0$ .  $d$  is the distance between robot and feature straight line.

In order to justify whether a feature straight line is similar to another one we must propose the following four necessary conditions.

- Justify whether the observation direction is consistent.
- Whether the intersection angle of two feature straight lines is smaller than a setting threshold.
- Whether the distance from the endpoint of one line segment to another feature straight line is smaller than a setting threshold.
- Justify whether two lines are independent by projecting line segment 1 to line segment 2. If the projective line segment has a common part with line segment 2 the two line segments might be not independent.



Fig. 7: Both lines are similar

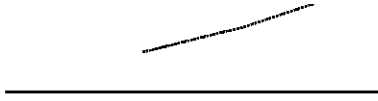


Fig. 8: Both lines are not similar



Fig. 9: Both lines are not similar



Fig. 10: Both lines are not similar

We give several examples in Fig. 7-10 to justify whether a feature straight line is similar to another one according to the aforementioned conditions. For example, two feature straight lines are similar in Fig. 7. The two feature straight lines aren't similar in Fig. 8-10 because Fig. 8 satisfies condition (d) and does not satisfy condition (c), Fig. 9 does not satisfy condition (b) and (d), Fig. 10 does not satisfy condition (b) and (c).

**Self calibration and map building:** In a structured environment, if we don't consider the small accumulative error and metrical error, then the coordinate of whole object outline in the environment detected by laser range finder on mobile robot will be consistent completely with the real one in global coordinate system. In fact, it is impossible because of inaccurate robot's pose. So robot's pose must be calibrated.

In present research the main strategy about SLAM of mobile robot with Fast-Hough transformation method is step-by-step mode. That is, when mobile robot moves a step, its pose can be calibrated with the referenced straight line and gets a new pose. According to the new pose calibrate some new straight line segments detected and choose some of them as the new referenced straight lines and plot them in the environment and so on.

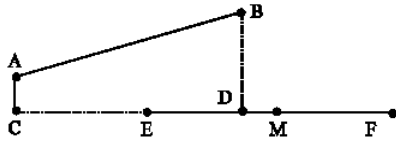


Fig. 11: The sketch of mergence of two similar straight line segments

**Mergence of straight lines:** When mobile robot moves along a very long wall which is out of the range of laser, thus the laser range finder can't detect the whole wall in one time. In this case, the mergence between new line segment and the old one is very necessary. We must justify whether the two line segments are on the common line according to the similar conditions. The simple way to merge two line segments is shown in Fig. 11. To extend the referenced straight line we first project the new detecting straight line AB to the relative referenced line EF and then the two endpoints of projection line segment CD can be reached. After a simple sort of four endpoints (C, E, D, F) justify whether the distance CM is bigger than EM, here M is the midpoint of referenced line segment EF. If yes, then we will extend EF to point C and get the new line segment CF. In this course we must consider the process of self calibration because correct pose is the key of merging two line segments.

**Self calibration:** Suppose the initial localization of mobile robot is correct, then the detecting feature straight lines are taken as the referenced straight line (generally speaking we choose one or several longer straight lines) and add them into the historical list of referenced straight line which is the real outline of the object in the environment theoretically. When robot moves a bit, get some old feature straight lines near referenced lines which might coincide completely or partially with these referenced straight lines. Then compute the deviation angle and distance between the new detecting line segment and the corresponding referenced one and calibrate the robot's pose. At the same time we also get some new feature straight lines and calibrate them according to new pose of robot. At last choose new longer straight lines and add them into historical list of referenced lines and so on. Thus the robot is able to finish the task of self-localization in a polygonal environment.

### SIMULATION RESULT

In this simulation experiment, Pioneer II mobile robot shown in Fig. 12 evolves in a virtual environment with the



Fig. 12: Pioneer mobile robot

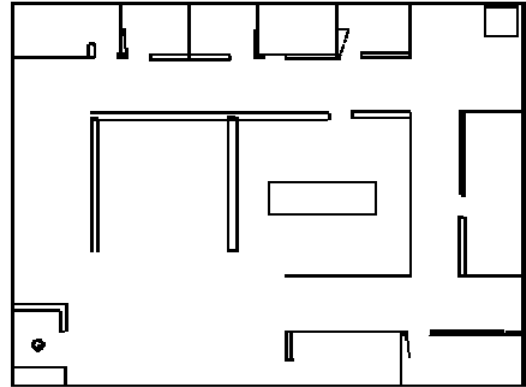


Fig. 13: Original environment map

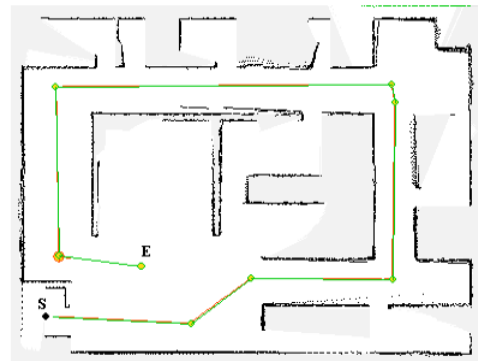


Fig. 14: The experimental result of SLAM with our method

size 18.7×14.0 m in Fig. 13. The method presented in this work is applied to SLAM by using SRIsim simulator developed by Activmedia Company (Fig. 14).

Main steps of experimental procedure is listed as follows:

- Receive the laser scan data and robot pose.

- Transform the laser scan data and robot pose data into global coordinate.
- Clustering the laser scan points
- Extend laser scan points, add some pixel points between two adjacent points, in order to make the line features more prominent and more accurate.
- In every points set, select two points randomly compute the angle of the line cross the two points, record the frequency of angles, pick up the line with Hough transform according to the most possible angle, find out the start and end point, then eliminate the points belong to the line, repeat the process till there are not enough points left.
- Add the detected lines to current detected lines list.
- If the reference lines list is empty, put current detected lines into it, otherwise, compute the pose error between current picking-up lines and reference lines, if they are similar and them merge them, calibrate reference lines and calibrate the robot coordinate.
- Plot the map with reference line.

**The experiment result:** In Fig. 11, the robot starts off from the position S. The color line is the running trace of the mobile robot in the environment. Position E is the end point of this running. In this course, SLAM can be performed successfully. By comparing from Fig. 10 and 11, the reconstructed map is very similar to the original one, so it has a high accuracy of SLAM.

The algorithm based on Fast Hough transform is very fast compare to classical Hough transform and has low computation amount, it is because classical Hough transform searches feature lines in global map space, especially, when the size of map is becoming larger, the computation amount is very prodigious, i.e., let alone multi-robot run in the environment simultaneously, just for single robot, there occurs no enough memory space and resorting phenomenon of running. Whereas Fast Hough transform searches feature lines in detecting field, more especially, in a certain angle range, which avoid from search feature lines in the range (0~360°).

Because we adopt the clustering method twice in Fast Hough transform algorithm, the precision of recognition is improved greatly.

However, this method is just fit to polygonal environment, because of the essential shorting of Hough transform.

## CONCLUSIONS

A SLAM algorithm based on fast Hough transform is presented by considering the strategy step and step mode. It overcomes the shortcoming of high computation amount and low accuracy of Classical Hough transform. In an polygonal environment, the algorithm runs perfectly, however, in unstructured environments we must consider other methods.

## REFERENCES

- Duda, R.O. and P.E. Hart, 1972. Use of the HT to detect lines and curves in pictures. *Comm. ACM.*, 15: 211-215.
- Fox, D., W. Burgard and S. Thrun, 1998. Active markov localization for mobile robots. *Robotics Autonomous Syst.*, 25: 195-207.
- Fox, D., 2003. Adapting the sample size in particle filters through KLD-sampling. *Int. J. Robotic Res.*, 22: 985-1004.
- Kwon, S., W. Yang and S. Park, 2006. An effective kalman filter localization method for mobile robots. *IEEE/RSJ International Conference on an Effective Kalman Filter Localization Method for Mobile Robots Intelligent Robots and Systems*, pp: 1524-1529.
- Lee, S., J. Lim and D. Cho, 2006. EKF Localization and mapping by using consistent sonar feature with given minimum landmarks. *SICE-ICASE. International Joint Conference Oct.*, pp: 2606-2611.
- Thrun, S., D. Fox and W. Burgard, 2001. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128: 99-141.
- Zang, Y. and K. Yuan, 2006. A two-step particle filter for SLAM of corridor environment. *2006 IEEE International Conference on Information Acquisition 20-23 Aug.*, pp: 370-375.