Proceedings of the World Congress on Electrical Engineering and Computer Systems and Science (EECSS 2015)
Barcelona, Spain – July 13 - 14, 2015
Paper No. 335

Efficient SLAM Scheme–Based ICP Matching Algorithm Using Image and Laser Scan Information

Sanghun Han, Jaehyun So, Youngjoon Han
Department of Electronic Engineering, SoongSil University
369 Sangdo-Ro, Dongjak-Gu, Seoul, Republic of Korea
loup86@naver.com; ru2ror@gmail.com; young@ssu.ac.kr

Donghyuk Shin
Department of Information and Telecommunication Engineering, SoongSil University
369 Sangdo-Ro, Dongjak-Gu, Seoul, Republic of Korea
zkfndkdl@nate.com

Abstract– This paper proposes a scan-matching simultaneous localisation and mapping (SLAM)-based Iterative Closest Point algorithm using laser scan information and images in an indoor environment. The ICP algorithm, which is one of the scan-matching methods calculates the closest position iteratively by adjusting the motion vector and rotation matrix of the model. The matching process requires a great deal of time, in accordance with the size of the point cloud model and repetitive execution. The proposed method uses part of the feature model and performs the matching by setting up the adjusting range of the motion vector and rotation matrix through the maximum velocity of a mobile robot. It then estimates the pose of the mobile robot using the relationship between the models. In order to evaluate the efficiency of the proposed algorithm, experiments driving the mobile robot were executed, such as estimating the pose of the mobile robot and building a map.

Keywords**:** SLAM, Scan matching, ICP algorithm

1. Introduction

Many researchers have studied simultaneous localisation and mapping (SLAM), which is the process by which a mobile robot localises itself and builds a map of an unknown area. There are two typical sensors used to collect information about the environment. One is a range sensor like a sonar or laser (Endres et al., 2012; Hahnel et al., 2003). Such a range sensor can quickly obtain range data about the environment. The other type is a vision sensor (Endres et al., 2012; Klein and Murray, 2007; Steux and El Hamzaoui, 2010). A vision sensor can obtain much more information and exhibits a good price–performance ratio.

SLAM techniques are based on sensors, which obtain various forms of data. Klein and Murray (2007) used a single camera to estimate the camera pose. The algorithm set the key frame and implemented bundle adjustment (BA) to localise the camera. However, there is a problem with this method in that a large map will increase the processing time. Steux and El Hamzaoui (2010) used a line laser sensor to scan the impact points. Their algorithm was rapid and simple, but its accuracy was somewhat low. Endres et al. (2012) used a Kinect sensor which could obtain depth and RGB data. Their system required more time and techniques for dealing with vision and range data, but was able to obtain more information on localisation than the other approaches.

Hahnel et al. (2003) proposed the SLAM algorithm to integrate a laser sensor and FastSLAM (Montemerlo et al., 2002). They used a scan-matching method involving Rao-Blackwellised particle filter (RBPF) particles. In scan-matching approaches, there are different methods used for two-dimensional (2D) and 3D scan matching. These methods can estimate a mobile robot's pose using the correlation between past and present scan models. This paper is restricted to the 2D scan-matching approach for the implementation of the real-time system.

In this paper, we propose a scan-matching SLAM using the Iterative Closest Point (ICP) algorithm. Our proposed method sets the motion vector to reduce repetitive matching and uses part of the model. We assume that the mobile robot moves on flat ground in an indoor environment, and there is no horizontal movement of the mobile robot.

In chapter 2, we explain the proposed algorithm. Chapter 3 gives the test result, and chapter 4 provides the conclusion.

2. The Proposed Method

2. 1. ICP Modelling

For the ICP model, this paper implements a robust feature model by fusing edge information and range data. To generate the edge, it is necessary to set the region of interest (ROI) where the laser is projected onto image. The edge information is extracted from the ROI and strong vertical components are selected through a filter. The filter creates a histogram by allocating a bin through detecting the edge component of the neighbouring eight directions and selecting the edge component which accrues many bins in the 90° and 270° directions. The ICP model consists of the range data corresponding to the selected edge and the edge information, as a feature. These features indicate the start and end position between the objects scanned. Figure 1 shows the composition of the ICP model through the filter.

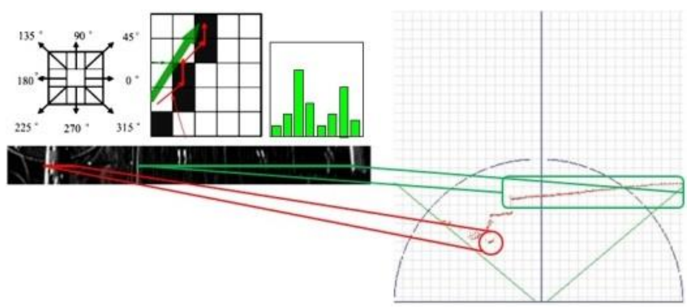


Fig. 1. The ICP modelling is used by a filter which makes histograms to find vertical edges and range data. The strongest value of the vertical edge and range data are fused, generating the ICP model.

If the filter detects several edge pixels, it searches in a direction which assigns more bins. These directions are the new starting point for performing the search again, and this is carried out until the edge pixels are no longer detected. The filter gives more weighting to the histogram, which has more bins at 90° and 270° amongst the obtained histograms, and generates the cumulative edge histogram. The process of creating the cumulative edge histogram is the same as that illustrated in Figure 2, and can be expressed as shown in (1).

$$H = \sum_{i=1}^{k} w_i h_i \tag{1}$$

Where H is the accumulated weighting edge histogram, k is the number of histograms and h_i is the detected histogram. The weight w_i is calculated by the ratio of the number of total bins in histogram and the number of assigned bins at 90° and 270°. Thus, these accumulated edge histograms and the corresponding range data generate the ICP model $M = \{m_1, \cdots, m_n\}$ by fusing.

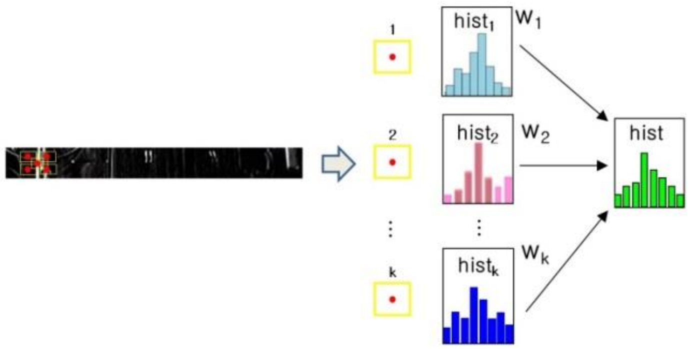


Fig. 2. Red points are detected by the edge histogram using the filter. The histograms are accumulated by (1).

2. 2. The ICP Algorithm

The ICP algorithm calculates the difference of the Euclidean distance between the two models by adjusting the motion vector and rotation matrix, and calculates the correlation between the models by selecting the lowest value. The proposed algorithm uses the following methods to reduce the repetitive execution of the algorithm:

1. Calculate the maximum movement of the mobile robot and the rotation per frame;
2. Set the range of the motion vector and rotation matrix;
3. Perform matching between the partial model from the previous frame and the model from the current one;
4. Iterate Step 3 within the set range and calculate difference of the Euclidean distance between the models; and
5. Select the smallest value and calculate the correlation between the models.

The maximum velocity of the mobile robot per frame is calculated to set the range of the motion vector and rotation matrix. We calculated the velocity using the moving distance and rotational angle through the encoder of the mobile robot, and set the range of the repetitive matching process. The moving distance and rotation angle had a very small value. The model for the matching used the frontal range of 50° ($101 \leq n \leq 300$) from the previous frame (t-1), and the whole part ($1 \leq n \leq 400$) from the current frame (t). Fig 3.a and Fig 3.b show the relation between the set range and models.

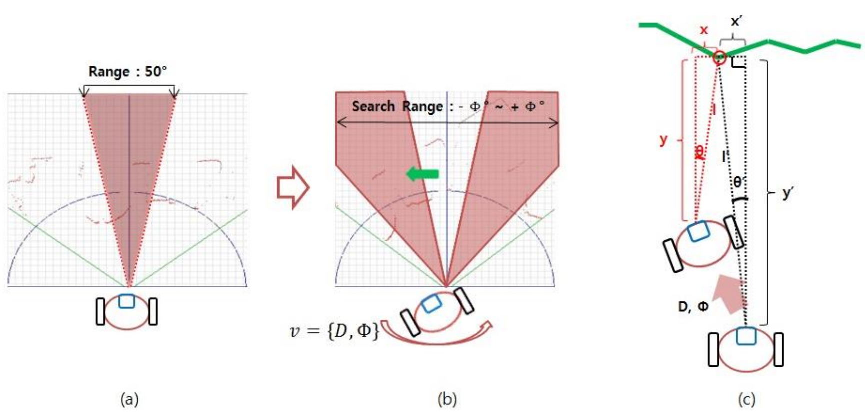


Fig. 3. The relationship between the models and the threshold. (a) Part of the model in the past frame. (b) The present model and threshold when the robot moves D and rotates Φ. The past model can be found in a search range. (c) The previous scan point is (x', y'), while (x, y) is the same point scanned at present; θ and θ' represent the angle between the robot and the point; and l and l' denote the distance. (c) Show the correlation of the same point between the past and present frame when the robot moves D and rotates ∅.

Fig 3.c shows the relationship between the mobile robot and the model. The previous model, $M_{t-1} = \{m'_{101}, \cdots, m'_{300}\}$, consists of the features $m'_i = \{x'_i, y'_i\}$. Each feature has $x'_i = l'_i \cos(\theta'_i)$ and $y'_i = l'_i \sin(\theta'_i)$. Here l'_i is the distance between the mobile robot and feature point, and θ'_i is the angle between the centre of the robot and the point. The positional relationship between the two models for the same point (x_i, y_i) can be written as:

$$\Delta x_i = D_t / \tan(\theta'_i + \Phi_t), \tag{2}$$

$$\Delta y_i = D_t. \tag{3}$$

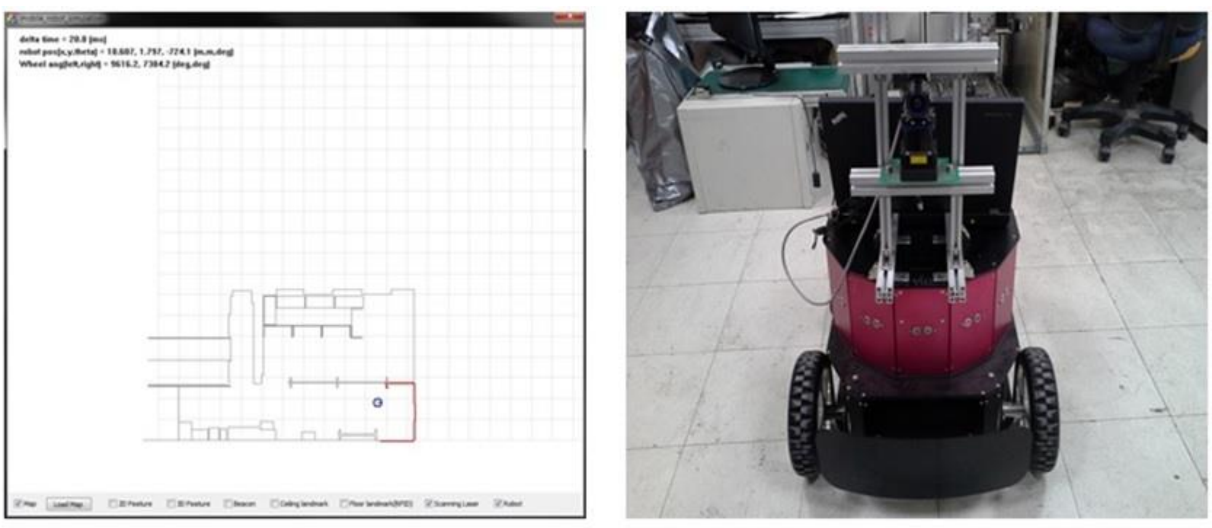
Here, D_t and Φ_t represent the maximum travel distance and rotation angle of the robot, and are adjusted for the matching. The ICP matching process between the two models through the setting range can be written as (4):

$$e = \sum_{-\Phi}^{\Phi} \sum_{-D}^{D} M_t - M_{t-1}(\Phi, D). \tag{4}$$

M_t and M_{t-1} are models from the current and previous frame, and e is the difference of the Euclidean distance between the models. The value which has the smallest Euclidean distance is selected by adjusting D and Φ to detect the best matching part between the previous model and the current model. Therefore, the pose of the robot through a relationship between the two models can be easily estimated, and the iteration of matching can be reduced by setting the range.

3. Experimental Result

In order to evaluate the performance of the proposed algorithm in this paper, we conducted an experiment using the Tetra-DS III and mobile robot simulation, as shown in Fig. 5. To evaluate the proposed algorithm, we used the mobile robot simulation that can simulate the SLAM algorithms. To collect data, we installed Imaging Source DFK 61BUC02 and UTM-30LX-EW on the mobile robot. The mobile robot was a differential wheel–driven robot where the resolution of the camera was 640 x 480 pixels and the accuracy of the laser sensor was $\pm$30mm at 10 m.



(a) (b)

Fig. 5. (a) is the mobile robot simulation, and (b) is the Tetra-DS III platform with the camera and the HOKUYO UTM-30LX-EW laser sensor.

3. 1. The mobile robot simulation

The coreSLAM (Steux and El Hamzaoui, 2010) that use the 2D scan-matching method is similar to our proposed method. We compared the proposed algorithm with the coreSLAM algorithm (Steux and El Hamzaoui, 2010) using the simulator to evaluate the performance, as shown in Fig. 6 and Table. 1. We performed the simulation several times in various maps. We can see that the coreSLAM algorithm has big error when features are not detected. As compared with the result, the proposed algorithm stays consistent.

Table. 1. Error denotes an average distance between the estimated position and the simulator, and frame per second (FPS) is an average operation time.

	Proposed	coreSLAM
Error(Pixel)	0.00409	0.05522
FPS	38.5	37

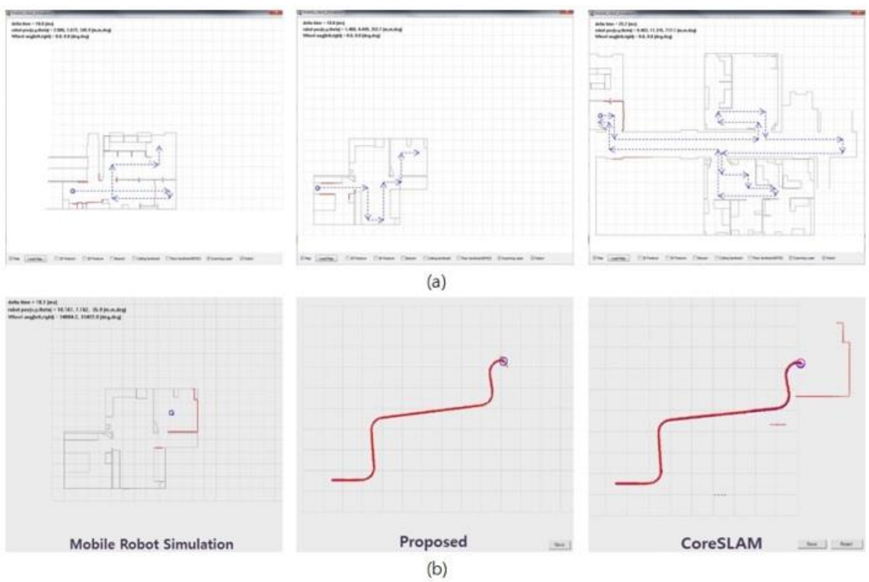


Fig. 6. (a) are the maps used in the simulation, and (b) is a scene of the simulation. First picture in b shows a ground-truth position, second picture shows an estimated position by proposed algorithm, last picture shows an estimated position by the coreSLAM algorithm.

3. 2. Experiment with the mobile robot

The experiment with the mobile robot was conducted. We calculated the distance $D = 15\text{mm}$ and the angle $\Phi = 0.2°$. This resulted in an average of 40 frames per second by reducing the iterations. As shown in Fig. 7, we drove the robot around our laboratory, whose size was 11.4 m x 10 m. Its environment was surrounded with chairs and desks. This result could be used to make a 2D map. The first map shows the estimated result using the proposed algorithm, while the second demonstrates the result of the encoder value of the robot. The final map shows the trace of the robot and the map of laboratory, as well as a comparison between the estimated result and the encoder value.

4. Conclusion

In this paper, we proposed a scan-matching SLAM which used the proposed ICP algorithm. We used a mobile robot which had a camera mounted on it and a laser sensor.

In order to find the robot's pose, we calculated the correlation between the ICP models. To match the model consisting of range and edge information, we used the part of the previous model which could be found in the present model. We set a range consisting of the velocity of mobile robot and the angle per frame to reduce the repetitive matching process. Finally, we were able to reduce the iterative matching process to estimate the robot pose and build a 2D map. The proposed algorithm was able to quickly obtain the robot pose and make a map.

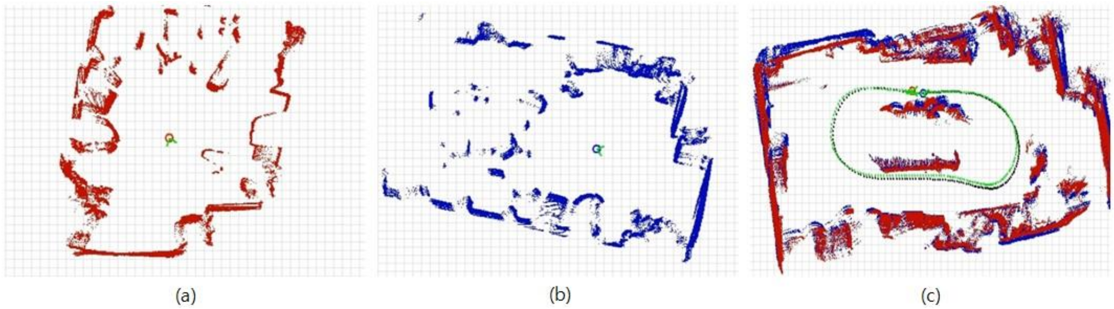


Fig. 7. (a), (c) The red point clouds and red circle are the results of estimation using the algorithm. The green dots show the trace result of robot pose estimation. (b), (c) The blue point clouds and blue circle indicate the results of the robot encoder value. The black dots show the trace using the robot encoder value

Acknowledgements

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (No. 2012M3C4A7032182).

This work was supported by the Industrial Convergence Core Technology Development Program (No. 10048474) funded by the Ministry of Trade, industry & Energy (MOTIE), Korea.

This work (Grants No. C0221180) was supported by Business for Academic-industrial Cooperative establishments funded Korea Small and Medium Business Administration in 20.

References

Besl, P. J., McKay, N. D. (1992). Method for Registration of 3-D Shapes. *Int. Soc. for Optics and Photonics*, 586-606.

Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., & Burgard, W. (2012). An Evaluation of the RGB-D SLAM System. *IEEE Int. Conf. on Robotics and Automation*, 1691-1696.

Hahnel, D., Burgard, W., Fox, D., & Thrun, S. (2003). An Efficient FastSLAM Algorithm for Generating Maps of Large-Scale Cyclic Environments from Raw Laser Range Measurements. *IEEE/RSJ Int. Conf. on Intelligent Robots and Syst.*, *1*, 206-211.

Klein, G., Murray, D. (2007). Parallel Tracking and Mapping for Small AR Workspaces. *6th IEEE and ACM Int. Symp. on Mixed and Augmented Reality*, 225-234.

Montemerlo, M., Thrun, S., Koller, D., & Wegbreit, B. (2002). FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. *AAAI/IAAI*, 593-598.

Steux, B., El Hamzaoui, O. (2010). TinySLAM: A SLAM algorithm in less than 200 lines C-language program. *11th Int. Conf. on Control Automation Robotics & Vision*, 1975-1979.