Improved KNN Scan Matching for Local Map Classification in Mobile Robot Localisation Application

M A Markom 1, A H Adom 1, S A Abdul Shukor 1, N Abdul Rahim 1, E S Mohd Muslim Tan 2 and B Ilias 1

1 School of Mechatronic Engineering, Universiti Malaysia Perlis, Kampus Ulu Pauh, 02600 Arau, Perlis, Malaysia
2 Centre of Diploma Studies, Universiti Malaysia Perlis, Kampus Unicity, 02100 Padang Besar, Malaysia

Corresponding author: marni@unimap.edu.my

Abstract. Localisation is essential for autonomous mobile robot system enabling it to locate itself within its environment. One method to perform localisation is to use scan matching with iteration closest point (ICP) algorithm. However, typical ICP may be prone to inaccuracies in localisation and mapping due to problems associated with laser range data limitation such as overshoot data and blank data. This paper presents the improvement to the above problem by the inclusion of a threshold to the KNN scan matching algorithm during iteration process. The threshold is a percentage of nearest point of incoming input with respected to reference point. Threshold values of 0%, 70% and 90% were tested, and improvements of the classification performance were observed with the increase in the threshold values, with the latter achieving 100% accuracy. This work shows that the use of threshold in scan matching may improve the accuracy of local map classification.

1. Introduction

There has been an increase in the implementation of mobile robot systems to assist humans. The use of mobile robots has seen successful implementations in various fields, such as home appliances [1], military [2], exploration [3,4], as well as search and rescue [5,6].

In mobile robot system, mapping and localisation can be considered as priority parts. Both are usually incorporated together in order to facilitate the mobile robot to move in a particular environment autonomously. The work on localisation algorithm development typically has types of algorithm-based which are scan matching [7,8], probability method [9] and Kalman filter [10]. These algorithms are applied to calculate and estimate the mobile robot position based on surrounding information that obtained from range finder sensor such as Kinect, laser range finder and ultrasonic. However, these algorithms can only be applied depends on its application and environment [11].

This paper has worked in indoor environment with a reference map that has been established. The suitable algorithm-based is scan matching. The key idea of using scan matching is to determine a displacement vector of two 2D scans, which are taken from two different times, by rotating and shifting the two scans against each other in such a way that the two scans optimally matched. Scan matching can be categorised into two methods, Point-to-point and Feature-to-feature.

The point-to-point scan matching needs two scans compared directly. Some example of point-to-point method are Iteration Closest Point (ICP) [7] and cluster method [12]. Usually, it applies in a static environment, but, the method can be enhanced for the dynamic environment. There are two problems that should be considered when dealing with the point-to-point method;

i. Scan results of incoming input are not close to reference scans
ii. Certain point of scanning results are missing
Because of this problem, the use of point-to-point method is very limited. It is not suitable to apply for certain range finder sensor, such as ultrasonic. This sensor only provides one data sample for each scan. If the ultrasonic scan fails to give a result, then, the whole system will be useless. In order to solve the limitation, researcher used laser range finder, it provides more scanning points, or multiple sensors [13] and multiple types of sensors [14], as well as classifier to determine the matching issue [7].

Feature-to-feature method is a more flexible method to perform scan matching. It is suitable for applications in both static and dynamic environments, but still requires assistance by some sort of decision analysis, accuracy calculation and perhaps pattern recognition. This scan matching has high computational complexity compared to the point-to-point method. Also, the quality of the matching is depends on the reliability of the features. Here, some features, for example, line segments or corners, are extracted respectively from previous scan and current scan and then the two scans are matched according to the correspondences between the two feature groups [15].

Both of these point-to-point and feature-to-feature methods apply iteration process which is consists of ‘rotation’ and ‘translation/shift’ in order to match the incoming input sample to the previous/reference scans. It will stop when the incoming input sample is matched with any of reference local maps. In order to have the stop, some studies used percentage of error, and some used number of rotation. The rotation and shifting processes are described as below:

1.1. The rotation
The rotation process involves the rotation of the scanned map in terms of angle, such as 1° to 10° steps with relative to the reference map. This is to eliminate issues with the orientation of the mobile robot with respect to its environment, and hence the reference map. This rotation process will be stopped when the scan matching achieved its expected error percentage. Figure 1(a) shows illustration of the rotation process example. In this Figure, the black shape is a reference, while, the red shape is the incoming scan. The red shape is going to rotate to be matched with the red shape.

1.2. The shifting/translation
The translation process is involved with a varying of x-axis and y-axis of the incoming scan to be fitted with the x-axis and y-axis of the reference. Figure 1(b) shows the translation process illustration. In this Figure, the red shape must be transformed to be matched with the black shape. The black shape is the reference and the red shape is the incoming scan. The translation process will be stopped after reaching its percentage error. Once it’s stopped, the whole process of scan matching is stopped and the results are going for further classification decision.

![Figure 1](image-url)

Figure 1. (a) the rotation process, the input is altered in angles, and (b) the shifting process, the input is altered in term of horizontal and vertical.

2. Scan matching and threshold
For this work, scan matching is computed using Euclidean distance. When the scan matching results are unsatisfied the threshold, the iteration begins. Three values of threshold is applied which are set to be 0%, 70% and 90%. This threshold is used to represent the percentage of nearest point of incoming
input with respect to the reference point. Higher the percentage means the incoming input is very similar to one of the reference local map. Else, it is otherwise. The flow of scan matching with a threshold is shown in figure 2.

![Scan matching flowchart]

**Figure 2.** The iteration process (rotation and transformation) in the scan matching based on the Euclidean distance

### 3. Reference map

In mobile robot self-localisation, there are cases that need reference to show and illustrate the mobile robot location. The reference can be the previous or initial map [16], location of beacon [17] and specific landmark [18].

In this study, the reference are consisted of initial local map. The initial map is obtained during the initial experiment, which is also known as exploration work. At this work, a testing location is recognised, and a few position of scanning to create local maps are established. For each location, 10 times of scanning is performed. The scanning data are analysed to be repetitive and consistent to ensure the reliability and valid. Figure 3 shows the diagram of reference local map 1 to local map 6.

![Reference local map 1 to local map 6]

**Figure 3.** The reference of local map 1 to local map 6
4. The experiment

A testing is conducted using 100 sets of new scanning sample that involves the segment/area of local map 1 to local map 6 each. This testing will show the performance of the rotation and translation process in order to justify the point to stop the iteration and to get back to the main algorithm to complete the classification process. The results are shown in graphic and the percentage of accuracy. The graphic is used to represent the rotation and translation process from the beginning until it finds the match among the local maps with threshold of 90%.

5. Local Map Classification Using KNN

KNN is used to identify the local map that matched with the incoming input. This KNN implements Euclidean distance as its distance parameter. The flow of the scan matching with KNN and location estimation is as below:

- Ready the KNN model with sample class labels, \((s_i, \theta_i)\).
- Ready the input/unlabelled sample, \((s_j, \theta_j)\).
- Convert the \(\theta_i\) (deg) and \(\theta_j\) (deg) into \(\theta_i\) (rad) and \(\theta_j\) (rad)
  \[
  \theta \text{ rad} = \frac{\pi}{180} \times \theta \text{ deg}
  \]
- Convert the point into \(x\) and \(y\)
  \[
  x_j = s_j + \cos \theta \text{ rad}
  \]
  \[
  y_j = s_j + \sin \theta \text{ rad}
  \]
where \(s_j\) is the distance of the scanning sample, \(x_j\) and \(y_j\) are the estimation results.
- Determine the \(K\)–parameter. Here, \(K\)–parameter uses is 3.
- Calculate the Euclidean distance, \(d_{ij}\), for all reference samples in the KNN model, \((x_i, y_i)\), with the input, \((x_j, y_j)\)
  \[
  d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}
  \]
where \(i\)-th and \(j\)-th is the number of data input sample and data reference sample.
- Repeat for all samples in the KNN model.
- Iteration process with \(\tau\), \((a = 0°, 70°, 90°)\).
- Ready the Euclidean distance, \(d_{ij}(m)\), get the classification results based on the lowest summation of Euclidean distance.

6. Results and Discussion

There will be two results of this work, i. Iteration performance results, and ii. The map classification result. All are presented as below;

6.1. Iteration results with and without threshold

Table 1 shows the iteration process results for the scan matching using three thresholds. The results are showed in terms of accuracy percentage using incoming input of local map 3. Based on the table, there is no results with percentage of 0%, which is means the incoming input is able to show similarity features to the reference map. At initial iteration results, the incoming input is able to match with a reference local map which is achieved threshold of 70%. However, this result is wrong and will end up with wrong classification of local map. This is a mistake if threshold of 70% is applied. For iteration with threshold of 90%, the iteration is performed a few times until the incoming input matched with any reference local map with 90% of accuracy. Based on the table, the accuracy of the local map 3 gets higher, while, some of the local maps get lower, and, several others retain similar results. The local map 3 shows that it starts with 62% and after having its fifth rotation, it turns to 91%.

6.2. Map classification results

The local map classification results are shown in Table 2 based on confusion matrix of KNN. The testing are conducted using 100 sample of incoming input for each local maps. Based on Table 2, the results of algorithm without iteration, total corrected incoming sample for local map 1 to local map 6
are 40%, 45%, 44%, 42%, 38% and 32%, respectively. For threshold of 70%, the total corrected incoming sample are 78%, 80%, 76%, 74%, 70% and 69%, respectively. While, for threshold of 90%, all incoming sample are classified correctly. These show such results because there are similarity features between near local maps. For example as shown in algorithm without iteration, for local map 1, the incoming input are classified in local map 2 and local map 3, but, there are no classification results out with local map 4 to local map 6. When the threshold is increased to 90%, there are many similarity features from near local maps are eliminated, and just leave the real features for the particular class of local map. In other word, incoming input will precisely classify at its own local map.

Table 1. The example of iteration process result with six time of rotations with the incoming input of local map 3.

| No of iteration | ± degree (°) | Local map 1 (%) | Local map 2 (%) | Local map 3 (%) | Local map 4 (%) | Local map 5 (%) | Local map 6 (%) |
|-----------------|--------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| -               | -            | 54              | 78              | 62              | 23              | 63              | 60              |
| 1               | +2           | 54              | 76              | 71              | 23              | 63              | 60              |
| 2               | +4           | 55              | 76              | 76              | 21              | 63              | 61              |
| 3               | +6           | 55              | 78              | 78              | 20              | 64              | 61              |
| 4               | +8           | 55              | 76              | 87              | 18              | 64              | 63              |
| 5               | +10          | 55              | 74              | 91              | 14              | 61              | 61              |

Table 2. Confusion matrix of KNN classification result

| Ref | No iteration | Incoming input | Threshold 70% | Threshold 90% |
|-----|--------------|----------------|--------------|--------------|
|     |              | Local map 1 | Local map 2 | Local map 3 | Local map 4 | Local map 5 | Local map 6 | Local map 1 | Local map 2 | Local map 3 | Local map 4 | Local map 5 | Local map 6 |
| LM 1 | 00 | 26 | 10 | 0 | 0 | 0 | 70 | 2 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LM 2 | 00 | 10 | 23 | 12 | 2 | 0 | 0 | 8 | 75 | 4 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LM 3 | 00 | 4 | 12 | 30 | 6 | 1 | 0 | 0 | 3 | 70 | 4 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LM 4 | 00 | 0 | 0 | 2 | 25 | 12 | 2 | 0 | 0 | 2 | 70 | 5 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LM 5 | 00 | 0 | 0 | 5 | 17 | 12 | 0 | 0 | 0 | 0 | 0 | 62 | 4 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LM 6 | 00 | 0 | 0 | 4 | 8 | 18 | 0 | 0 | 0 | 0 | 0 | 3 | 65 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

7. Conclusion

As conclusion, the scan matching incorporated with threshold really helps localisation algorithm in term of accuracy. The work has proven by comparing the results without threshold or using low value of threshold, which is both results showed low classification accuracy compared to high value of threshold. The needs of high value of threshold, such as 90%, is due to the presence of similarity features among the near local maps. If each of local map has much different in features, then, low value of threshold can be used. The feature similarity can be analysed using statistical analysis, such as ANOVA, MANOVA and can be represented using Dendrogram.

8. References

[1] Do, H. M., Pham, M., Sheng, W., Yang, D., & Liu, M. (2018). RiSH: A robot-integrated smart home for elderly care. Robotics and Autonomous Systems, 101, 74-92.
[2] Siciliano, B., & Khatib, O. (Eds.). (2016). Springer handbook of robotics. Springer.
[3] Joochim, C., & Roth, H. (2010). Mobile Robot Exploration Based On Three Dimension Cameras Acquisition. IFAC Proceedings Volumes, 43(23), 116-121.
[4] Lauri, M., & Ritala, R. (2016). Planning for robotic exploration based on forward simulation. Robotics and Autonomous Systems, 83, 15-31.

[5] Bakhshipour, M., Ghadi, M. J., & Namdari, F. (2017). Swarm robotics search & rescue: A novel artificial intelligence-inspired optimization approach. Applied Soft Computing, 57, 708-726.

[6] Bai, L., Guan, J., Chen, X., Hou, J., & Duan, W. (2018). An optional passive/active transformable wheel-legged mobility concept for search and rescue robots. Robotics and Autonomous Systems, 107, 145–155.

[7] Konecny, J., Prauzek, M., & Hlavica, J. (2016). ICP Algorithm in Mobile Robot Navigation: Analysis of Computational Demands in Embedded Solutions. IFAC-PapersOnLine, 49(25), 396-400.

[8] Addie, I., Marni Azira, M., Abdul Hamid, A., & Mohd Muslim Tan, E. S. (2017). Scan Matching and KNN Classification for Mobile Robot Localisation Algorithm. In Proceeding of 2017 IEEE 3rd International Symposium in Robotics and Manufacturing Automation (ROMA). 19-21 Sept. 2017. Kuala Lumpur. DOI: 10.1109/ROMA.2017.8231836

[9] Vroegindeweij, B. A., IJsselrode, J., & van Henten, E. J. (2016). Probabilistic localisation in repetitive environments: Estimating a robot’s position in an aviary poultry house. Computers and Electronics in Agriculture, 124, 303-317.

[10] Zhu, D., Zhao, B., & Wang, S. (2018). Mobile target indoor tracking based on Multi-Direction Weight Position Kalman Filter. Computer Networks, 141, 115-127.

[11] Klancar, G., Zdesar, A., Blazic, S., & Skrjanc, I. (2017). Wheeled mobile robotics: from fundamentals towards autonomous systems. Butterworth-Heinemann.

[12] Rashid, A. T., Frasca, M., Ali, A. A., Rizzo, A., & Fortuna, L. (2015). Multi-robot localization and orientation estimation using robotic cluster matching algorithm. Robotics and Autonomous Systems, 63, 108-121.

[13] Ilias, B., Shukor, S.A.A., Adom, A.H., Ibrahim, M.F. and Yaacob, S., 2016, May. A novel indoor mobile robot mapping with USB-16 ultrasonic sensor bank and NWA optimization algorithm. In Computer Applications & Industrial Electronics (ISCAIE), 2016 IEEE Symposium on (pp. 189-194).

[14] Santos, J.M., Couceiro, M.S., Portugal, D. and Rocha, R.P., 2014, May. Fusing sonars and LRF data to perform SLAM in reduced visibility scenarios. In Autonomous Robot Systems and Competitions (ICARSC), 2014 IEEE International Conference on (pp. 116-121).

[15] Petrich, J., Brown, M. F., Pentzer, J. L., & Sustersic, J. P. (2018). Side scan sonar based self-localization for small Autonomous Underwater Vehicles. Ocean Engineering, 161, 221-226.

[16] Huang, H., Lee, D. H., Chang, K., Li, W., & Acharya, T. D. (2018). Development of mobile platform for indoor positioning reference map using geomagnetic field data. Computers & Electrical Engineering, 68, 557-569.

[17] Zhang, J., Shi, C., Sun, D., & Han, Y. (2018). High-precision, limited-beacon-aided AUV localization algorithm. Ocean Engineering, 149, 106-112.

[18] Qu, X., Soheilian, B., & Paparoditis, N. (2018). Landmark based localization in urban environment. ISPRS Journal of Photogrammetry and Remote Sensing, 140, 90-103.