

# A self-adaptive loop matching algorithm based on GA for 2D lidar SLAM

Honghao Zhao<sup>§</sup>, Wenxing Zhu<sup>\*</sup>

School of Control Science and Engineering, Shandong University, Jinan, China

## ABSTRACT

Simultaneous localization and mapping (SLAM) based on 2D lidar is the vital technology for indoor mobile robot mapping and navigation, and graph optimization has become a common method to solve this problem in the recent years. In graph-based SLAM, loop detection is a key step to obtain global pose constraints, and the real-time performance of this process ensures that the back-end optimization of the current frame can be completed smoothly before the arrival of the data for the next moment. However, due to the limitation of mobile robot's computing resources, when the global map reaches a certain scale, the success rate of loop detection which has a positive impact on the mapping accuracy will decrease with the number of loop constraints is directly proportional to the number of all poses. Therefore, we propose a self-adaptive matching method based on genetic algorithm (GA) to calculate the loop closure constraints between the current scan and each local map, so as to speed up the loop detection process. The experimental result shows that our method is superior to the traditional graph-based SLAM solutions in large scale map construction.

**Keywords:** 2D lidar SLAM, genetic algorithm, self-adaptive, loop closure, scan matching

## 1. INTRODUCTION

Indoor robot is an important research direction in the field of artificial intelligence. Environment perception, decision making and motion control are the basic steps for robot to realize intelligent operation. Simultaneous localization and mapping (SLAM) is the behavior that autonomous mobile robots use sensors to obtain the range information of the surrounding environment in an unknown scene for constructing the environment map while localizing. SLAM algorithms based on 3D lidar<sup>1</sup> and multi-sensor fusion<sup>2</sup> are research hotspots in the past few years, but 2D lidar makes 2D SLAM a good choice for indoor mapping<sup>3</sup> because of its low cost and simple implementation.

Most of the early 2D SLAM methods are built map by direct matching, for example, reference<sup>4</sup>. Iterative closest point (ICP) is one of the most widely used algorithm, which finds the best pose transformation to align the two adjacent laser point cloud data by iterative solution, and the sum of the pose transformation at each time is the motion path of the robot. However, these methods can not eliminate the errors caused by the completed local matching, and the cumulative error will cause a significant drift when building large-scale maps.

Filter-based SLAM is based on the Bayes' law, and can be expressed by the prior and posteriori probabilities. The Gmapping algorithms based on extended Kalman filter (EKF) and particle filter<sup>5</sup> are the main branches of this method. In this algorithm, the particle filter needs to maintain the system state of each particle at each moment, which will cause generous computing consumption with the growth of map scale until it is unbearable.

The graph-based SLAM corrects the pose estimation of the robot by using the observation of the sensor. In this method, the trajectory of the robot is represented by the pose graph and the global pose is optimized according to all the constraint relations in the graph. In the widely used graph-based SLAM algorithms, such as Karto, the global loop constraints are usually obtained by correlative scan matching (CSM). CSM<sup>6</sup> is a probabilistically motivated matching algorithm, which uses exhaustive search to find the optimal pose transformation with the highest response value in the search space. It also uses multi-resolution searching strategy and look-up table to speed up the searching process. This matching method improves the robustness of loop construction, but in the case of limited computing resources, the high time complexity will limit the number of loop constraints, which is especially obvious in large-scale map construction.

Genetic Algorithm is a kind of heuristic which can find the optimal solution in the global search space by genetic iteration in a short time and is not easy to fall into the local minima. GA is used in combination with SLAM in reference<sup>7</sup>,

§201934537@mail.sdu.edu.cn; \*zhuwenxing@sdu.edu.cn

which speeds up the optimization process of scan matching and improves the overall efficiency of SLAM. In this paper, we first give an overview and theoretical analysis of the graph-based SLAM system, and propose a self-adaptive loop matching method based on genetic algorithm for the back-end loop detection process of SLAM, which takes into account both the efficiency and accuracy of the matching process. We use the open source map data set for comparative experiments to verify the optimization effect of our algorithm in map construction.

## 2. GRAPH-BASED SLAM SYSTEM

Our SLAM method uses 2D LiDAR to obtain the distance information of the surrounding environment, and uses odometer to provide pose estimation in advance. The whole SLAM system is functionally composed of two parts, the front-end and the back-end.

### 2.1 Front end of SLAM

The task of the front end of the SLAM system is to find the optimal pose estimation  $\delta = (\delta_x, \delta_y, \delta_\theta)$  of the current frame in the local map, which includes the translation component  $(\delta_x, \delta_y)$  and rotation component  $\delta_\theta$ , and insert the laser scanning data to the maintaining local map. Through the correlative scan matching (CSM), we can combine the predicted values obtained by odometer with the observed values obtained by lidar to get the pose estimation at the current moment.

*Correlative scan matching.* Correlative scan matching is a scan-to-map matching algorithm which solve the problem in a probabilistic way. The goal of CSM is to find the optimal pose  $\hat{\delta}$  of the current frame in the local map that maximizes the probability of observing the objects. A rectangular area centred on the location estimated by the odometer is used to represent the possible extent of the final location. When matching begins, this region was exhaustive searched in  $(x, y, \theta)$  windows for obtaining the location  $\delta$  with the highest response value.  $M(p)$  is the probability function of the grid point in the local map, and  $T_\delta$  is the transformation matrix of pose  $\delta$ . The optimal pose  $\hat{\delta}$  can be computed by:

$$\hat{\delta} = \underset{\delta}{\operatorname{argmax}} \sum_{j=1}^N M(T_\delta p_j) \quad (1)$$

$$T_\delta = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \quad (2)$$

CSM also presents a multi-resolution searching strategy and uses lookup table that project the laser data in advance for better real-time performance, and specific theoretical details can be referred to Olson's paper.

### 2.2 Back end of SLAM

With the passage of time, the measurement error of odometer and the calculation error of scan matching will gradually accumulate, which greatly reduces the mapping accuracy of the system. We can eliminate these errors as much as possible through global pose graph optimization and loop closure detection. Pose graph optimization can be formulated as a nonlinear least squares problem:

$$\hat{x} = \underset{x}{\operatorname{argmin}} \sum E_i(x) \quad (3)$$

$$E_i(x) = e_i(x)^T \Omega_i e_i \quad (4)$$

where  $x$  is the estimated pose of robot, and  $\hat{x}$  is the optimized pose.  $\Omega_k$  is the information matrix. The goal of graph optimization is to minimize the sum of  $e_i(x)$  which represents the error between observations  $z_i$  and predictions  $f_i(x)$ , it can be computed by:

$$e_i(x) = z_i - f_i(x) \quad (5)$$

$$f_i(x) = \begin{bmatrix} R_i^T(t - t_i) \\ \theta - \theta_i \end{bmatrix} \quad (6)$$

Sparse pose adjustment (SPA) is used to solve the nonlinear optimization problem, and we use loop closure detection to detect loops which will be regarded as the observation constraints in the graph optimization problem. Traditional graph-based SLAM methods use correlative scan matching method to detect and compute the optimal loop closure

constraints, it leads to a high quality of matching results at the cost of computation time which will cause the decrease of loop detection efficiency that benefit the global mapping accuracy. In order to speed up the matching process, we propose a GA-based self-adaptive matching strategy for the optimal poses in the relevant local maps.

### 3. GA-BASED SELF-ADAPTIVE LOOP CLOSURE DETECTION AND MATCHING

#### 3.1 GA-based matching approach

By speeding up the loop closure matching process at the back-end, the frequency of loop detection will increase, which can improve the mapping accuracy with more loop constraints when the global map grows large. We achieve this by using the detection and matching method based on Genetic Algorithm.

*Determining the searching window.* For the scan matching problem, we can use  $W$  as the search window, we set the linear and angular search window offsets  $W_x = W_y = 8$  m and  $W_\theta = 30^\circ$ . The linear search resolution is  $r_t$ , so we can choose the angular search resolution  $r_\theta$  as:

$$r_\theta = \cos^{-1} \left( 1 - \frac{r_t^2}{2d_{max}^2} \right) \quad (7)$$

where  $d_{max}$  is the maximum range of scan points. We can get the integral number of steps covering given linear and angular search window sizes:

$$n_x = \left\lceil \frac{W_x}{r_t} \right\rceil, \quad n_y = \left\lceil \frac{W_y}{r_t} \right\rceil, \quad n_\theta = \left\lceil \frac{W_\theta}{r_\theta} \right\rceil \quad (8)$$

*Chromosome coding.* We first expand the linear search space in two dimensions to one dimension for representation, i.e., converting set  $\mathcal{N} = \{-n_x, \dots, n_x\} \times \{-n_y, \dots, n_y\}$  into linear index  $\bar{\mathcal{L}}$  of one dimension, the range of  $\bar{\mathcal{L}}$  is  $0 \sim 2n_x \cdot 2n_y$ . Assume the coordinate of search centre is  $\varepsilon_0(x_0, y_0)$ , we can get the linear index of point  $\varepsilon_p(x_p, y_p)$  as:

$$\bar{\mathcal{L}}^p = \left\lfloor \frac{y_p - y_0 + W_y}{r_t} \right\rfloor \cdot 2n_x + \left\lfloor \frac{x_p - x_0 + W_x}{r_t} \right\rfloor \quad (9)$$

And we use  $\bar{\mathcal{L}}_{max}$  to represent the max value of  $\bar{\mathcal{L}}$ , which is  $2n_x \cdot 2n_y$ . In terms of the angular search window, angular index  $\bar{\mathcal{R}}$  is created to represent the rotation component and the max value  $\bar{\mathcal{R}}_{max} = 2n_\theta$ .

In the coding step, it is necessary to ensure that the chromosome can represent all the  $x$ ,  $y$  and  $\theta$  in the feasible solution for global optimization. Let the length of the chromosome be  $l = l_1 + l_2$ , where the length of the left part is  $l_1$  and the length of the right is  $l_2$ . In order to ensure that the genetic algorithm can reach any position of the feasible solution, the  $l_1$  and  $l_2$  must satisfy:

$$2^{l_1} > \bar{\mathcal{L}}_{max}, \quad 2^{l_2} > \bar{\mathcal{R}}_{max} \quad (10)$$

*Fitness function.* Response value  $R_p$  for scan matching is taken as fitness function in the GA,  $goal_i$  is the probability value of scan point  $p_i$ , and  $goalmax$  is the max response value with all the scan points hit surrounding objects.

$$R_v = \frac{\sum_{i=0}^n goal_i}{goalmax} \quad (11)$$

*Genetic operator.* Genetic operators include mutation operator, selection operator and crossover operator. The conventional roulette strategy is used for selection operation, and the two-point crossover is employed for crossover operation. The mutation operation adopts the basic position mutation method. Considering the discontinuity of the matching process, the crossover rate should be reduced and the mutation rate should be expanded, which can prevent the genetic algorithm from falling into local iteration.

*Chromosome decoding.* After chromosomes  $(i, j) = (i_1 i_2 \dots i_{l_1}, j_1 j_2 \dots j_{l_2})$  was obtained by crossover, mutation, and selection, the corresponding linear and angular index needs to be calculated from equations below, this is the decoding process of chromosomes.

$$\bar{\mathcal{L}} = \sum_{k=0}^{l_1-1} i_k \cdot 2^k, \quad \bar{\mathcal{R}} = \sum_{k=l_1}^{l_2+l_1-1} j_k \cdot 2^{k-l_1} \quad (12)$$

### 3.2 Self-adaptive matching strategy

In the process of loop detection, GA-based matching method can significantly improve the matching efficiency. However, when the geometric structure of the surrounding environment of the robot is relatively simple, it is difficult to find a sufficient number of loops, at this time the accuracy requirement of loop matching is more important than its speed requirement. Genetic algorithm, as a global heuristic approach, has a higher optimization speed than exhaustive search. But this method can't guarantee the final result to be globally optimal as exhaustive search. In view of the above situation, we make a compromise between matching speed and matching accuracy, and propose an adaptive matching strategy based on the expected number of loops to meet the actual needs of robots in different scenes. The actual number of loops at the current time can only be known after all the loop detection and matching are completed, and impossibly be calculated in advance. For this reason, we define the loop closure factor  $\Psi$  to estimate the degree of difficulty in loops detection. The loop closure factor  $\Psi_k$  at the current time can be calculated by the following equation:

$$\Psi = \eta \cdot \frac{\sum_{i=k-K}^{k-1} C_i}{K} \quad (13)$$

$$\eta = \frac{\sum_{j=0}^k d_j}{|t'_k - t_0|} \quad (14)$$

where  $C_i$  is the number of loop constraints at each time before the current moment, and its average value represents the preliminary prediction of the loop number at the current time. The loop gain  $\eta$  indicates the degree of geometric loop of the surrounding environment, which is essentially the ratio of mileage scalar to distance vector.  $d_j$  is the measurement of odometer at each time,  $t_0$  is the translation component of robot pose at the beginning moment, and  $t'_k$  is the translation component of the estimated pose obtained by the front-end at the current time. In order to achieve self-adaptive matching in different environments, we set thresholds  $\Psi_1$  and  $\Psi_2$ , where  $\Psi_1 < \Psi_2$ . If  $\Psi \leq \Psi_1$ , the value of loopback detection is beneath expectation, so we could complete the matching through CSM to ensure the matching accuracy. When  $\Psi_1 < \Psi < \Psi_2$ , we adopt a compromise of using exhaustive search in the rough matching stage of CSM and GA-based search in the fine matching stage. If  $\Psi \geq \Psi_2$ , only GA based matching method is used to find the optimal solution in the search space.

## 4. EXPERIMENTAL RESULTS

In this section, we use the benchmark measure toolkit provided by University of Freiburg to compare our method to other 2D lidar algorithms. The benchmark evaluates SLAM algorithms by comparing the relative displacements of estimated poses with the corresponding ground truth relations which is manually aligned by hand. We use the Radish data set which contains several typical indoor maps (shown in Figure 1) to conduct our comparative experiment, and our SLAM system is running on an Intel i5-3230M CPU with a single thread for backend processing.



Figure 1. 2D map of intel research lab with manually aligned ground truth relations.

Table 1 shows the simulation results of comparing our algorithm with the traditional graph-based Karto SLAM. The errors contained in the tables are divided into translational and rotational, which can comprehensively evaluate these algorithms. And the same test of the Radish data set which is displayed in Table 2 evaluates our algorithm with the Gmapping SLAM. Whether compared with traditional graph-based SLAM or Gmapping based on particle filter, it can be seen from the simulation results in the tables that the GA-based algorithm has a certain improvement in error elimination and mapping accuracy.

Table 1. Dataset test of the GA-based graph SLAM and Karto.

	<b>Absolute translational (m)</b>	<b>Squared translational (m<sup>2</sup>)</b>	<b>Absolute rotational (deg)</b>	<b>Squared rotational (deg<sup>2</sup>)</b>
Aces building GA-based	0.238 ± 0.533	0.411 ± 2.320	0.336 ± 0.829	1.056 ± 4.714
Karto	0.408 ± 1.325	1.604 ± 7.009	0.440 ± 0.683	0.874 ± 3.295
Intel Lab GA-based	0.165 ± 0.470	0.306 ± 1.924	1.253 ± 2.866	8.632 ± 25.707
Karto	0.207 ± 0.823	0.702 ± 3.204	1.875 ± 3.830	16.170 ± 83.348
MIT Killian GA-based	0.249 ± 0.608	0.507 ± 2.692	0.357 ± 1.433	1.734 ± 4.635
Karto	0.350 ± 0.752	0.608 ± 3.159	0.755 ± 2.514	7.548 ± 22.900

Table 2. Dataset test of the GA-based graph SLAM and Gmapping.

	<b>Absolute translational (m)</b>	<b>Squared translational (m<sup>2</sup>)</b>	<b>Absolute rotational (deg)</b>	<b>Squared rotational (deg<sup>2</sup>)</b>
MIT CSAIL GA-based	0.226 ± 0.482	0.384 ± 1.815	0.522 ± 2.054	4.732 ± 12.788
Gmapping	0.443 ± 0.879	2.175 ± 8.033	1.216 ± 3.724	14.915 ± 77.362
Freiburg building GA-based	0.328 ± 0.859	0.930 ± 3.556	0.921 ± 3.714	11.408 ± 33.224
Gmapping	0.507 ± 1.074	3.404 ± 14.538	1.030 ± 2.715	8.799 ± 17.036
Freiburg hospital GA-based	0.575 ± 0.836	1.255 ± 3.880	1.457 ± 2.602	10.575 ± 52.016
Gmapping	0.873 ± 2.825	8.239 ± 34.372	2.609 ± 5.330	34.152 ± 142.274

## 5. CONCLUSIONS

In this paper, we give a systematic overview and theoretical analysis of the current mainstream graph-based 2D SLAM scheme and elaborate the corresponding functions and implementation of the front end and the back end of the SLAM system. Aiming at speeding up the loop detection process of back end, we propose an GA-based self-adaptive loop matching algorithm which can enhance the construction efficiency of loop constraints in the scenario with a large number of loops, and finally improve the global mapping accuracy. The simulation result shows that our algorithm outperforms the traditional graph-based 2D SLAM in large scale indoor environment.

## ACKNOWLEDGEMENTS

This work is funded in part by the National Natural Science Foundation of China (Grant No.61773243), Major Technology Innovation Projects of Shandong Province (Grant No. 2019TSLH0203) and the National Key Research and Development Program of China (Grant No. 2020YFB1600501).

## REFERENCES

- [1] Liang, S., Cao, Z. Q., Wang, C. P. and Yu, J. Z., "A novel 3D LiDAR SLAM based on directed geometry point and sparse frame," *IEEE Robotics and Automation Letters*, 6(2), 374-381
- [2] Im, G., Kim, M. and Park, J., "Parking line based SLAM approach using AVM/LiDAR sensor fusion for rapid and accurate loop closing and parking space detection," *Sensors*, 19(21), 4811 (2019).
- [3] Liang, J., Iv, Q., Zhang, Y., Lin, H. C. and Wei, H., "A kinectV2-based 2D indoor SLAM method," *Artificial Intelligence, Automation and Control Technologies*, 1-5 (2017).
- [4] Lu, F. and Milios, E., "Robot pose estimation in unknown environments by matching 2D range scans" *Journal of Intelligent and Robotic Systems*, 18(3), 249-275 (1997).
- [5] Aristeidis, G. T., Emmanouil, G. T. and Loukas, P., "Topological based scan matching—Odometry posterior sampling in RBPF under kinematic model failures," *Journal of Intelligent & Robotic Systems*, 91(3-4), 543-568 (2018).
- [6] Olson, E. B., "Real-time correlative scan matching," *IEEE International Conference on Robotics and Automation*, 4387-4393 (2009).
- [7] Tang, M., Chen, Z., and Yin, F. L., "An improved adaptive unscented FastSLAM with genetic resampling," *International Journal of Control, Automation and Systems*, 19, 1677-1690 (2021).