Acta Technica *62* (2017), No. 5B, 199–216

An improved LIDAR SLAM algorithm based on FastSLAM for autonomous vehicle

Li ChenXi[2], Zhang Jun[2,3], Liu YuanSheng[2],Jin XinYu[2],Li GuangJing[2], Cheng Xu[2]

Abstract. In order to realize autonomous navigation of autonomous vehicles that don't rely on GPS, an improved LIDAR SLAM method based on FastSLAM algorithm is proposed in this paper. First, a large number of data obtained by LIDAR are prepossessed, including sparse points removal, Voxel Grid, point cloud separation. Then, the processed data are used to make the point cloud match of the adjacent frame by using the method of PLICP algorithm to obtain the coarse positioning of the autonomous vehicle. After that, the particle scatter points are carried out near the rough locations, and the Niche Technique Particle Swarm Optimization method is introduced to locate the autonomous vehicle accurately. The map of LIDAR point cloud is represented by the raster map. After getting the precise positioning of the autonomous vehicle, a local map which carried by the particle is added to the global map to update the map. Finally, the system outputs the map of the environment and the trajectory of the autonomous vehicle. The validity of the proposed method is proved by the experiment on autonomous vehicle platform. The experimental results show that the proposed technique could reduce the position error and successfully build the map of the environment.

Key words. LIDAR, Simultaneous Localization and Mapping(SLAM), autonomous vehicles, Niche Technique, Particle Swarm Optimization.

[1]Acknowledgment - This work was supported by the Premium Funding Project for Academic Human Resources Development in Beijing Union University (Academic Exploration Plan— 'Small Tornado' Special-Purpose Intelligent Vehicles Research Team); Academic Human Resources Development in Beijing Union University"(Grant No.BPHR2017EZ02);Newton Fund Project (Grant No. UK-CIAPP\324, Talents Cultivation and Cooperation Oriented to Intelligent Vehicle Industrialization); Innovation Team Building Promotion Plan of Beijing Municipal University (Grant No. IDHT20170511).

[2]Workshop 1 - College of Robotics, Beijing Union University, Beijing 100101, China

[3]Corresponding author: Zhang Jun; e-mail: xxtzhangjun@buu.edu.cn

1. Introduction

Simultaneous localization and mapping(SLAM) remains a key issue in the domain of autonomous vehicles for navigation. It can describe as: place an autonomous vehicle at an unknown location in an unknown environment and for the vehicle to incrementally build a consistent map of this environment while simultaneously determining its location within this map.

SLAM has been at the core of the autonomous robot system for many years and it is more vital and challenging especially for outdoor mobile robots nowadays. Significant SLAM investigation of autonomous robots has been performed during the last decades. Autonomous vehicle is a kind of special autonomous robot. In recent years, enormous interest in making autonomous vehicles more smarter continues to increase rapidly. Competitions are established both at home and abroad, for instance, DAPPA Grand Challenges and China Autonomous Vehicle Future Challenge. Especially the 2007 Urban challenges [1], though many simplification were made for the purpose of competition, it is the first significant demonstration of autonomous driving for the vehicles themselves through a city-like environment.

There are two main kinds of SLAM solution in autonomous vehicle application: Vision SLAM and LIDAR SLAM. SLAM with LIDAR is more popular as LIDAR can provide accurate and ample range measurements information. Hence, this paper also takes advantage of the 3D laser scan system to complete the SLAM process.

A good introduction of SLAM can refer to [2], which includes numerous classical existing algorithms. For example, EKF, UKF, SEI, RBPF, FastSLAM, GraphSLAM etc. The traditional algorithms [2] have been a great benefit to humans, but unfortunately, most of them are computationally and exist obvious limitations in their practical application. FastSLAM is one of the powerful techniques to solve SLAM problems, however the linear approximations of nonlinear functions and particle depletion phenomenon are the obvious drawbacks of it.

The contribution of this paper is that a new LIDAR SLAM strategy which combines PLICP and NTPSO is proposed for improving the obvious drawbacks of the standard FastSLAM algorithm. The data source of the algorithm is only LIDAR point cloud. Moreover, the experiment is verified on autonomous vehicle platform, and the effectiveness of the algorithm is proved.

The structure of the whole paper is organized as follows. The related work is introduced in section2, the method of the proposed algorithm is presented in section3 and an improved LIDAR SLAM algorithm is presented in section4, including the localization and the mapping technique. Then, experiments are conducted, the results and discussion are presented in section5. In the end, conclusions are described in section 6.

2. Related work

With the development of autonomous vehicle, 3D simultaneous localization and mapping technology in unstructured environment grow in popularity. Among them, LIDAR SLAM technology is one of the main method frequently used to solve the

problem, and there are different ways to reach the goal. One of the key issues of LIDAR SLAM is the registration of the point cloud.

In point cloud registration problem, standard ICP algorithm[3] is usually used to match laser returns between scans. ICP algorithm has been proved very effective in building the map of the environment, however, the slow convergence and locally optimal solutions are two main disadvantages of it. There are many researchers devoted to overcome these shortcomings, and different variations of ICP algorithm are developed. Alismail H et al. [4] extend the ICP algorithm and take into account inter-sample pose errors to estimate the continuous-time trajectory. Liu W et al.[5] Combine the advantages of iterative closest point (ICP) and iterated sigma point Kalman filter (ISPKF) to solve LIDAR-IMU time delay calibration. Wolfgang Hess et al.[6] combine scan-to-submap matching with loop closure and graph optimization. A local grid-based SLAM approach is used to create the individual subset trajectories, and all scans are matched to create loop closure constraints in their background. Zhang J et al.[7] divide the complex SLAM problem into two algorithm simultaneously. One algorithm performs odometry to estimate the velocity of the laser scanner and a second algorithm to match and register the point cloud. And the main method is to extract feature points from the LIDAR point cloud. Kumar G A et al.[8] consider the typical geometric structure of indoor environments and they use point-to-point scan matching algorithm from a horizontally scanning primary LIDAR to calculate the position of the UAVs and a vertically scanning secondary LIDAR to calculate the altitude. Besides, a Kalman filter is also used to derive the 3D position by fusing primary and secondary LIDAR data.

Additionally, there are also other sensors used to do SLAM. López E et al.[9] propose to integrate different state-of-the art SLAM methods based on vision, laser and inertial measurements using an Extended Kalman Filter (EKF).

3. The Method of The Proposed Algorithm

3.1. FastSLAM Algorithm

FastSLAM is based on R-B particle filter SLAM, which uses particles to represent the posterior of some variables, and all other variables are represented along with some other parameter probability density functions. The essence of FastSLAM is that for each particle, the error of a single map is conditionally independent. That is, in the FastSLAM algorithm, localization and map construction can be considered separately. FastSLAM uses particle filters to compute the posterior path of a robot.

$$\begin{aligned} &p\left(x_{0:t}, m_t | z_{0:t}, u_{1:t}\right) = \\ &p(\mathrm{x}_{0:\mathrm{t}}, m_t | z_{0:t}, u_t) \bullet \prod_{j=1}^{M} p(m_j | x_{0:t}, z_{0:t}, u_{1:t}) \end{aligned} \tag{1}$$

For each feature in the map $n(n{=}1,...,N)$, FastSLAM uses a separate estimator $p(m_n|x_{0:t}, z_{0:t})$ stands for its location. The advantage of FastSLAM is that the proposal distribution is locally optimal. That is, conditional on available environmental

information $X_{0:k-1}^{(i)} Z_{0:k}$ and $U_{0:k}$,it gives the smallest possible variance in importance weight $w_k^{(i)}$ for each particle. However, it suffers the particle depletion phenomenon. The main steps of FastSLAM can be described as follows.

(1)Sampling

For each particle, it is generated by proposal distribution functions on the basis of the previous generation of particles. Different from R-B particle filtering, the proposal distribution function is as follows:

$$x_k^{(i)} \sim P(x_k | X_{0:k-1}^{(i)}, Z_{0:k}, u_k) \tag{2}$$

Where

$$\begin{aligned} &P\left(x_k | X_{0:k-1}^{(i)}, Z_{0:k}, u_k\right) = \tfrac{1}{C} P(z_k | x_k, X_{0:k-1}^{(i)}, Z_{0:k-1}) \\ &P(x_k | x_{0:k-1}^{(i)}, u_k) \end{aligned} \tag{3}$$

And C is the normalizing constant.

(2) Weighting

Each particle has an important weight according to the importance function.

$$w_k^{(i)} = w_{k-1}^{(i)} C \tag{4}$$

(3) Resampling

Resampling is accomplished by selecting particles of significant weight from particle set $\left\{X_{0:k}^{(t)}\right\}_I^N$ instead of particles with small weights, including the associated maps of the particle.

(4) Mapping

Perform an EKF update on the observed landmarks as a simple mapping operation with known vehicle poses for each particle.

3.2. Particle Swarm Optimization

Particle swarm optimization[10] (PSO) is a bionic algorithm based on swarm intelligence which proposed by Kennedy and Eberhart in 1995. The algorithm is easy to operate and implement, and it has a global searching ability for low dimensional functions with a fast search speed.

In the PSO algorithm, each particle in the search space corresponds to a possible solution of the problem to be solved. Suppose that M particles are randomly initialof in the search space of Ddimensions. The position of the i^{th} particle is expressed as $X^i = \left(x^{i1}, x^{i2}, \ldots x^{iD}\right)$ and $V^i = \left(v^{i1}, v^{i2}, \ldots v^{iD}\right)$ indicates the speed.

The best location of the particle itself in history from the beginning to the present is denoted as P_t^i the best location of the whole particle swarm experience is represented as G_{t-1}. Each particle iteratively updates the speed and location of itself based on the two extreme values, update equation is as follows:

$$V_t^i = \omega \bullet V_{t-1}^i + c_1 \bullet r_1() \bullet \left(P_{t-1}^i - X_{t-1}^i\right) + c_2 \bullet r_2() \bullet \left(G_{t-1} - X_{t-1}^i\right) \tag{5}$$

$$X_t^i = X_{t-1}^i + V_t^i \tag{6}$$

r_1 and r_2 are the are random numbers between [0,1], ω is the inertia weight. In general, the bigger value of ω , the stronger global search ability. Otherwise, the local search ability is, c_1 and c_2 are the learning factors.

3.3. Niche Techniques

According to the above equation, in the standard PSO, the particle mainly relies on the best position of individual history and the best position of global history to guide its position update in the search space. In the search process, if the best location of the individual history is gathered in a locally optimal solution region of the space, all particles are rapidly approaching the region. For an optimization problem PSO algorithm can only find an optimal solution, and it can not guarantee that the optimal solution is different, so this leads to the loss of the population diversity falling into a local optimum, and premature convergence or stagnation of search will occur. In order to avoid the phenomenon, niche technique strategy is introduced into PSO algorithm.

Niche technology[11] is a group stochastic optimization technique for multiple optimal solutions in multi-mode problems. It is derived from the genetic algorithm domain. Niche is derived from ecology, which refers to a small living environment in which a species is located. In artificial systems, niche techniques are mainly used to improve the global search performance of swarm based stochastic optimization algorithms. Niche particle swarm optimization algorithm is divided into two stages: The first stage is to find niches for each particle depending on the distance between the particles. In the second stage, the particle swarm optimization (PSO) algorithm is used to update the location and speed of each niche, and it must ensure that the optimal value of the group of particle swarm is only in the small habitat group. Classical niche technology has three mechanisms: preselection mechanism, sharing mechanism, and exclusion mechanism. A small niche technology based on the exclusion mechanism is adopted in this paper. The main idea is to measure the similarity between individuals according to some measure function, and then crowd out similar individuals. As the iterative process is carried out, individuals in the group will gradually be classified into each small niche, thus maintaining the diversity of the group.

4. An Improved LIDAR SLAM Algorithm

The localization and mapping process of improved algorithm based on FastSLAM proposed in this paper are mutually aided. First of all, it needs to get the point cloud prepossessed in the positioning process. Next the PLICP algorithm is used to complete the crude positioning of the autonomous vehicle, and then the particles scattered points in the rough location near by the location of the particle. Finally, in order to optimize particles and find the best particle which closes to the actual

situation, the NTPSO optimization technology will be applied. After that, the local map carried by the optimal particle is converted to the global coordinate system to update the global map.

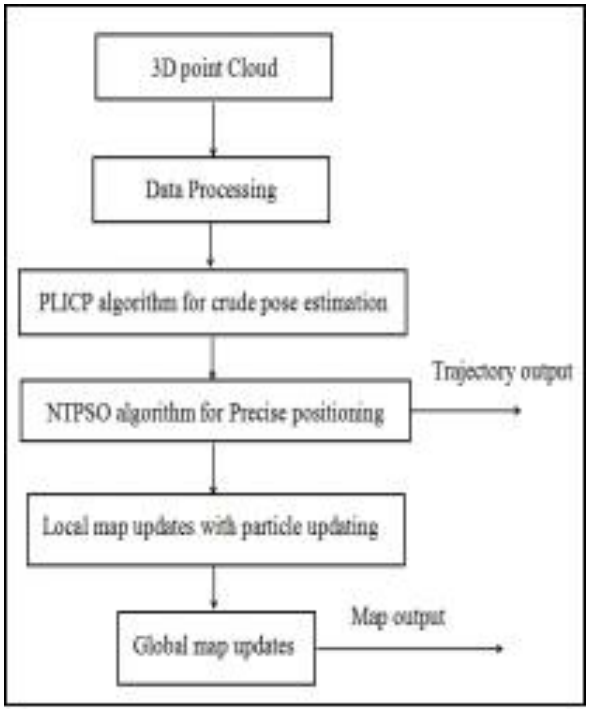

Fig. 1. The software system for LIDAR SLAM

4.1. Localization

4.1.1. Laser Scan Data Process In this paper, a 64 bit 3D-LIDAR for velodyne is mounted to the top of the autonomous vehicle. It provides the 3-dimensional but sparse point cloud of the surrounding environment with the frequency of 10 Hz. That is, there are 10 frames data are returned per second. The huge data are a big burden to its real time requirement. There is no doubt that we need to deal with the numerous data sets.

(1)Sparse points removal

As mentioned above, due to the sparseness of the point cloud returned by the Velodyne in far away area. It can't represent accurate environmental information but just as a load to the algorithm. So this paper chooses a valid reasonable distance to filtering out the invalid points. During our experiment, the maximum distance is set at 70 meters.

(2)Voxel Grid

Though down sampling the dense point cloud is essential in our algorithm, it needs to retain the detail information of the environment as much as possible. To

achieve this goal, and Voxel Grid filters seem a better solution.

To vocalize the point cloud, we build a 3D cubic voxel grid using the point cloud P,and M stands for the number of the points. Then, for each Voxel calculating the focus of all points using filtering like this:

$$\begin{cases} x = \frac{1}{M}\sum_{i=1}^{M} x_i \\ y = \frac{1}{M}\sum_{i=1}^{M} y_i \\ z = \frac{1}{M}\sum_{i=1}^{M} z_i \end{cases} \tag{7}$$

(3)Point cloud separation

The point cloud contains both ground points set$G$ and above-ground points set AG . This paper aims at achieving the pose estimation of the vehicle by register the point clouds of G. Obviously, the ground points are meaningless to our registration. Hence, the above-ground points[12] are extracted using a RANSAC plane filter algorithm from the points cloud map.

4.1.2. Localization based on PLICP and NTPSO LIDARs have been widely used in autonomous vehicle filed, because it can provide ample and accurate information of the surrounding environment. During the last decades, feature extract[13] is a mainstream method used in SLAM solutions, including EKF, GraphSLAM etc. What's more, data association usually must be associated with it. There is no doubt that the computation cost of dealing with the huge points is expensive. In order to reduce the computational cost and improve the positioning accuracy, a niche particle swarm optimization algorithm based on PLICP point cloud matching is proposed in this paper.

(1)Vehicle pose coarse estimation

Traditional particle filter requires a large number of particles to get closer to the real position[14] of particles, and this method significantly reduces the efficiency of the algorithm. In this paper, the rough estimation of pose is obtained by the PLICP algorithm which is used to match the point cloud of adjacent frames. The purpose of rough estimation of position and orientation is to find the high probability region of autonomous vehicle.

One of the main merits of scan matching is that it doesn't need to do feature extract in task execution. Therefore, the time consumption will be reduced, and a more accurate and FastSLAM approach will be presented.

Traditional scan matching method is mainly ICP[15] and its deformation[16], such as MBICP. It has a high requirement for its initial value, or else, the result of the method will be divergence and it would be a disaster for SLAM.

PLICP scan matching uses the point-to-line principle to replace the point-to-point principle of ICP. That is, when a points finds the correlation point, it doesn't just find the nearest point with it in current scany_t, but it will find two points which are closer to itself, j_i^1 and j_i^2, then compose a segment point $j_i^1 - j_i^2$, and the PLICP

process instructions are shown below.

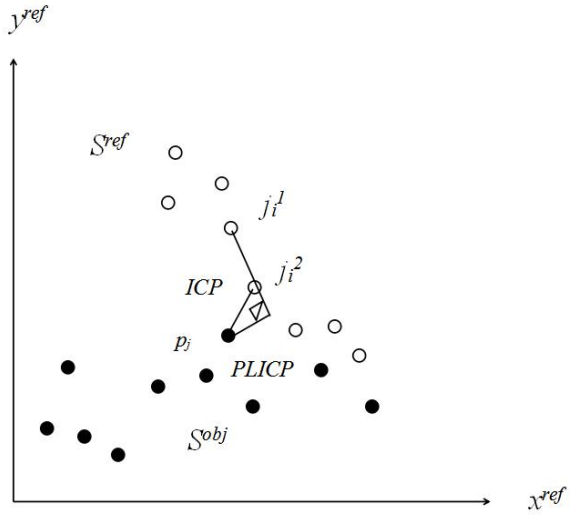

Fig. 2. The algorithm principle of PLICP method

The primary input reference scanned data was recorded as y_t^1 in the experiment, the points among them is represented as p_i,a second scan data y_t,the points among them is represented as $p_j.q_0$,a first guess for the roto-translation to be found. S^{ref} stands reference surface produced by first scanning data y_{t-1}, it is a polyline obtained by connecting sufficiently close points. S^{obj}stands for the target point surface set produced by scanning data y_t. The index i refers to the points in the scan ,and index j refers to the points in . In the end, index k refers to the iterations of the algorithm.

The point-to-line metric is as follows:

$$\underset{q_{k+1}}{min} \sum \left(n_i^T \left[p_i \oplus q_{k+1} - \Pi \left\{S^{ref}, p_i \oplus q_k\right\}\right]\right)^2 \tag{8}$$

The steps are repeated until convergence or termination of the cycle. It has been proved that the above method is converges, and it converges better than average ICP. PLICP can reduce the matching times and much closer to the real distance.

Two coordinates are defined in this paper. They are the local system (x_l, y_l, z_l) which the origin is lidcoordinated and the world coordinate system(X_w, Y_w, Z_w) with the artificially set origin.

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = \mathrm{R}_{3\times3} \times \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} + t_{3\times1} \tag{9}$$

First, according to the current guess$q_k = \ (t_k, \ \theta_k)$, the coordinate transformation between the second sets of scan data to the first scan data is completed, the point

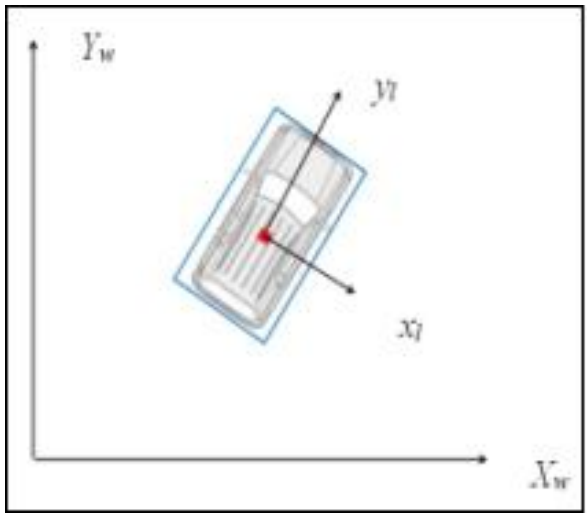

Fig. 3. Coordinate explanation of the SLAM system.

$p_i$is converted to pointp_i^w:

$$p_i^w \triangleq p_i \oplus q_k = R(\theta_k)\, p_i + t_k \tag{10}$$

For each point p_i^w, find the closest two points in the first set of data, j_i^1 and j_i^2 respectively. The corresponding relation is called C_k for all points to line segments in step k. Consisting of a series of arrays $< i, j_i^1, j_i^2 >$, indicating that point i matches line segments $j_i^1 - j_i^2$.

And the error function is:

$$J(q_{k+1}, C_k) = \sum_i \left(n_i^T \left[R(\theta_{k+1})\, p_i + t_{k+1} - p_{j_1^i} \right] \right)^2 \tag{11}$$

This is the sum of the distances between the lines from the point i to the containing segment.

Getting the q_{k+1} method is similar to that in the paper [10] and is no longer repeated.

After finding the minimum value of q_{k+1}, and then the rotation matrix and the translation matrix[17] M will be found.

$$M_{t-1}^t = \begin{bmatrix} R & & & t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

Then the position of the autonomous vehicle at the t moment x_t^Dwould be got.

$$x_t^{(i)} = f(x^{t-1}, R, t) + \theta_t^{(i)} \tag{13}$$

(2) Particle update to complete pose optimization

Because of the existence of various noises and errors, the rough estimate obtained by the PLICP algorithm is only the high probability region of the pose of the autonomous vehicle. Therefore, this paper proposes optimizing the pose of the autonomous vehicle. In this paper, particle scattering is carried out near the coarse estimate, and the particle swarm optimization algorithm is used to optimize the position of particles.

The position of each particle is optimized, and the position and attitude of each vehicle represented by each particle are updated by using the speed and position equations in the PSO algorithm:

$$V_t^D = \omega \bullet V_{t-1}^i + c_1 \bullet r_1 \bullet \left(x_{pbest}^D - x_t^D\right) + c_2 \bullet r_2 \bullet \left(x_{gbest} - x_t^D\right) \tag{14}$$

$$x_t^{D*} = x_t^D + V_t^D \tag{15}$$

$x_{pbest}^D$and x_{gbest}respectively represent the local optimum and global optimum of the predicted value of the autonomous parking space. x_t^{D*} is the updated pose.

(3)Niche technique

Then, by using the appropriate niche radius, the Gauss mutation operator is used to make the random disturbance, which ensures that each niche has only one fine particle, In this way, the diversity of the population is maintained, and the searching ability of the particles is also improved. This paper uses niche technology based on crowding out mechanism. Calculate the distance between two particles:

$$D(x_t^{n*}, x_t^{m*}) = |x_t^{n*} - x_t^{m*}| \tag{16}$$

$n, m = 1, 2, ..., N$ $(n \neq m)$ N is the number of particles.

If the distance between two particles is less than the setting threshold, random disturbance by Gaussian mutation operator is implemented, until reach the testing number of loop, or the distance between the two is greater than the niche radius.

$$x_t^{n*} = x_{tSmaller}^{n*} \bullet (1 + c_3 \bullet d_3) \tag{17}$$

x_t^{n*} stands the pose after the disturb, c_3 is disturbance coefficient, d_3 is an matrix of 1*3, whose elements are random numbers that obeys the standard.

4.2. Mapping

In this article, we use a grid map representation to represent the map. Scanned cloud information is divided into grids. Each of which has a size of 20*20*20(cm), 0 indicates that the grid is not occupied, and 1 indicates that the grid is occupied. In this paper, the prior scanned map is used as the initial global map. When the autonomous vehicle is moving, the position and pose are determined by the positioning algorithm mentioned above. When the PLICP algorithm and the NTPSO algorithm[18] are used to obtain the particles which represent the vehicle posed, and we also obtain the local point cloud map of the particles at the same time.

4.2.1. Particle Weight Calculation For all of the above particles$N$, Firstly, the raster map of the local point cloud carried by each particle is converted to the global coordinate system, and then a match[19] between the two maps will be made.

The weight calculation formula can be expressed as:

$$w_t^{n*} = \frac{local\ raster\ map}{\text{global}\ raster\ map} \tag{18}$$

The matching degree of the two maps will be used as the importance weight of the particles.The better the match degree, the greater the weight, and the worse the matching degree, the smaller the weight.

4.2.2. Resampling Set degradation detection formula:

$$N_{eff} = 1/\sum_{n=1}^{N} (w_t^{n*})^2 \tag{19}$$

If N_{eff} is less than the default threshold, then the resampling[12] is performed.

After finding in the optimized particles, the local coordinates of the local cloud points obtained by the optimized particles are converted into global coordinate representation, and the local map is added to the global map. Thus, completing the update of the global map.

5. Experiments and results

5.1. System Overview

The experiment was carried out on the autonomous vehicle platform at Beijing Union University. As showed in Figure 4, the 64-line LIDAR is placed on the roof of the autonomous vehicle.

One of the advantages of the proposed algorithm is that only the data of a single sensor[21] is used and this avoids data calibration and fusion of different data sources. The sensor is a Velodyne HDL-64 LIDAR. We will use data from this sensor to illustrate the proposed method. The laser scanner has 360° field views and 64 lines/s scanning rate. It returns 133333 points per second. Here, note that the coordinates of the initially returned point cloud is in the local coordinate system mentioned above, and it will project into the world coordinate after the calculation of the PLICP algorithm.

5.2. Experiment Environments and Procedures

In this paper, environment information will be scanned by human as priori map beforehand, and the initial pose of the vehicle is recorded. The 3D laser point cloud collected from the experimental platform of autonomous vehicle is processed by the improved SLAM algorithm proposed in this paper. The processing of the initial

Fig. 4. Autonomous vehicle platform of Beijing Union University

Fig. 5. Velodyne LIDAR

point cloud includes the aforementioned remove the sparse points, Voxel grid and point cloud separation. Next, the PLICP algorithm is the application to do point cloud registration of the cloud point after pretreatment, and in order to reduce the time consumption, the KD-tree is also applied in the algorithm. After that,

the transformation matrix R and t of the autonomous vehicle are obtained. As the experimental environment and site are too large, this paper intercepted part of the experimental site for experimental explanation, and the registration results are shown below.

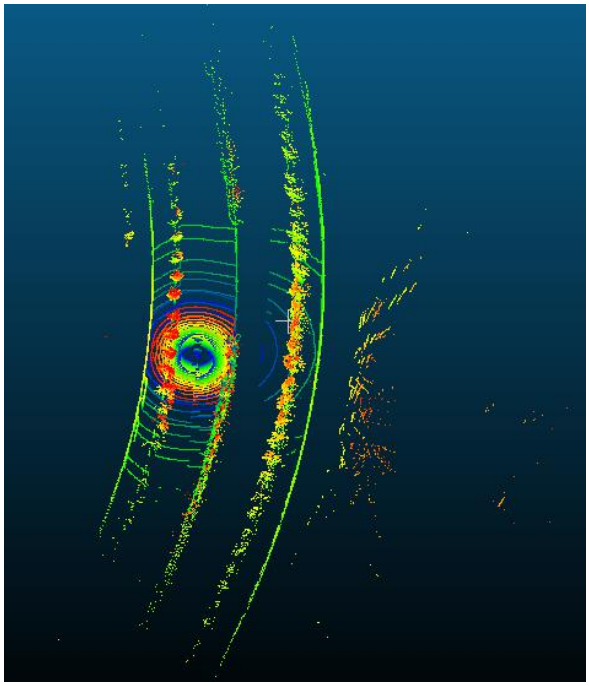

Fig. 6. Point cloud before preprocessing

The figure above shows two local point cloud maps and their application before and after the PLICP algorithm matches. Figure 6 shows the point cloud of the two local frames before the algorithm is processed. Figure 7 shows the results after two frames of point cloud matching. The red points represent the local point cloud of the y_{t-1}th scan, and the green points represents the local cloud of the y_tth scan.

In this paper, the experimental environment is the real suburban road lane. District Garden Expo test field in Fengtai of Beijing City, there are pedestrians, vehicles and non pedestrian vehicle, the Garden Expo satellite map is shown in Figure 8.

In the environment mentioned above, the laser SLAM algorithm proposed in this paper enables the intelligent vehicle to navigate autonomously in the environment, and simultaneously records and outputs the real-time navigation trajectory and the environment map. In order to make a comparison about the experiment, three groups of experiments were carried out in this paper, namely, manual acquisition of map trajectories under the GPS positioning function, autonomous driving under the proposed algorithm and autonomous driving under the FastSLAM algorithm. Among them, the trajectory of GPS is taken as the standard route, and the effec-

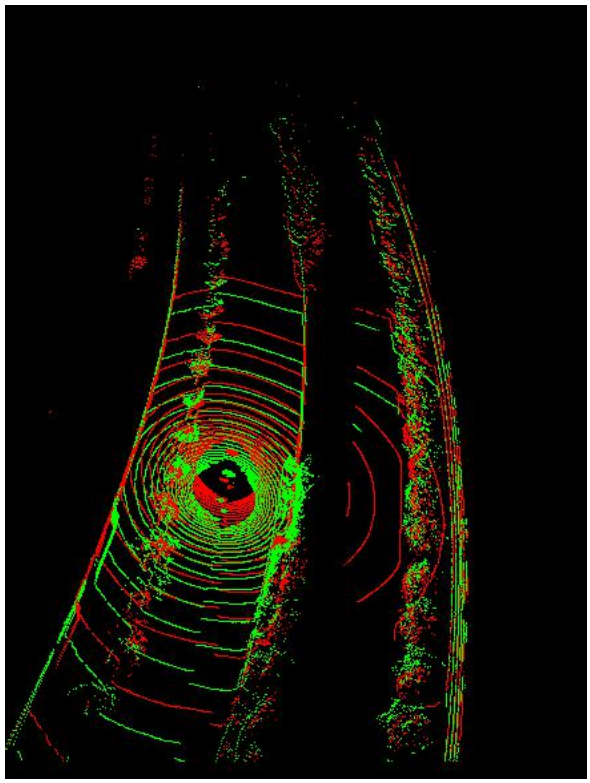

Fig. 7. Point cloud after preprocessing

Fig. 8. District Garden Expo test field in Fengtai of Beijing City

tiveness of the proposed algorithm is illustrated by comparing the trajectory of the two SLAM algorithms with the GPS trajectory.

As we can see from the above picture, the point cloud map of the environment was established, and the red arrow stands for the pose of the autonomous vehicle

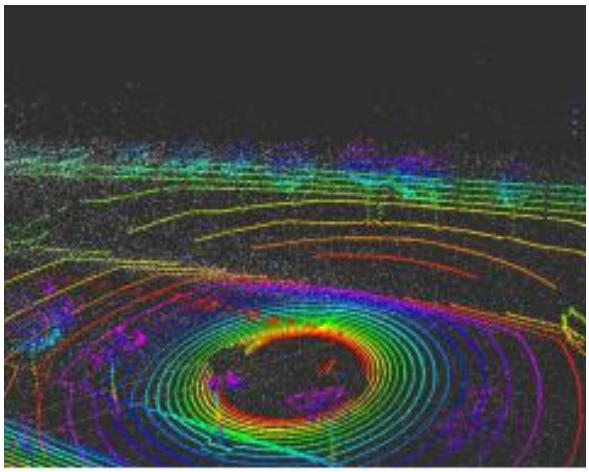

Fig. 9. Description of the localization and mapping process

of each moment and it is represented by particles in the figure. These positions are connected together to become the trajectory of the autonomous vehicle.

5.3. Experiment Results and Analyse

As showed in Figure 10, the real-time trajectory plots for three different localization algorithms are obtained.

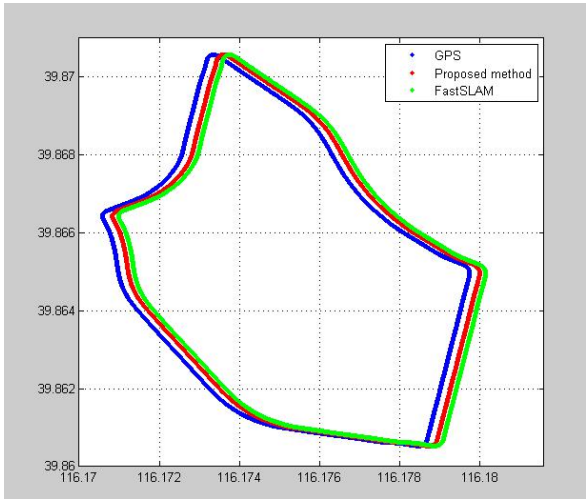

Fig. 10. Localization results comparison of GPS, the proposed SLAM method and FastSLAM

The localization results are shown in Figure 10 where the blue line represents the standard trajectory of GPS which is formed by an experienced driver, and the green

line represents the trajectory of FastSLAM algorithm, and the red line represents the trajectory of proposed SLAM method of this paper. It is obvious that the result of the proposed method is better than FastSLAM.

In order to further analyze the positioning accuracy of the two algorithms, 6 experiments were carried out respectively for each algorithm, and the error and variance of per meter of the location were analyzed as follows table.

Table 1. Error of real time localization

	Error mean x-axis(m)	**Error variance x-axis(m)**	**Error mean y-axis(m)**	**Error variance y-axis(m)**
FastSLAM	0.4523	0.0875	0.4468	0.7776
Proposed method	0.2506	0.0748	0.2483	0.5998

Comparing the error mean of the two different SLAM method, the positioning error has been reduced from 44.9cm of FastSLAM to 24.94cm of the proposed method.

In addition, one of the advantages of the improved FastSLAM algorithm is the reduction in the number of particles used, we have counted the particle number of the particle filter used in the experiment of the two SLAM algorithms to complete the process of SLAM technology, and the time execution result is showed in the following table.

Table 2. Numbers of particles of the two SLAM algorithm

	Numbers of particles	**Environment**	**Execution time/s**
FastSLAM2	25	3km	1262.36
Proposed method	25	3km	841.67

It is obvious that the proposed method needs fewer particles than FastSLAM algorithm. Besides, the proposed method reduces the time consumption.

At the same time, the PLICP algorithm is used to align each pair of the local point cloud maps, and finally the global map of the environment will be successfully built. Figure 11 shows the mapping results of the proposed SLAM method, and the red line stands the localization results of the algorithm, namely the trajectory of the autonomous vehicle.

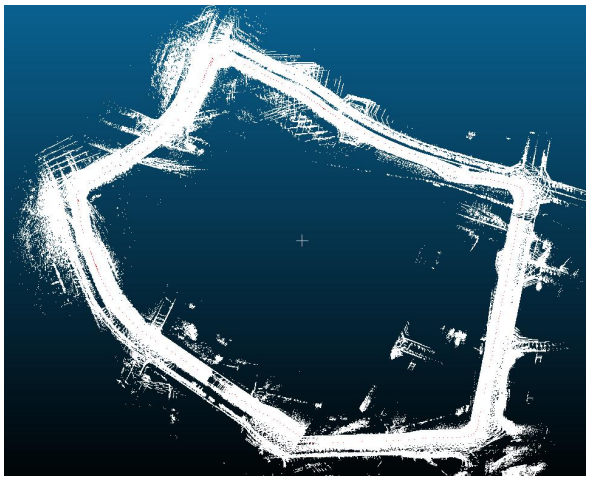

Fig. 11. Global Point cloud map of Garden Expo area building by the algorithm of this paper and the real-time trajectory of the autonomous vehicle

6. Conclusion

The algorithm proposed in this paper is mainly used to solve the autonomous vehicle's safety localization and navigation. As we can see from the experiment, it can successfully complete the localization and build the map of the environment simultaneously. The algorithm uses fewer particles to achieve more accurate position with the combine of PLICP and NTPSO function, and it improves the efficiency of the SLAM process. Besides, the proposed method has been tested on the autonomous vehicle platform of BUU. The results of the experiment demonstrate the effectiveness of the method, and as a contrast. It also shows that the method has the advantage over FastSLAM within a shorter time. With the continuous exploration of autonomous vehicle laser SLAM technology and its mature development, laser SLAM technology will be extended to more complex unstructured environment like dynamic and three-dimensional application. In the near future, the proposed technique will be applied to the autonomous vehicles application along with the combination of GPS, visual SLAM, or even with IMU, odometer for auxiliary at the same time. It makes the vehicle to get more rich source information and more comprehensive environmental information, so that it can realize fully safety self-driving of the autonomous vehicle.

References

[1] W. Burgard, O. Brock, C. Stachniss: *Map-Based Precision Vehicle Localization in Urban Environments.* Robotics: Science & Systems Iii, June, Georgia Institute of Technology, Atlanta, Georgia, Usa. DBLP (2008),121–128.

[2] H. Durrantwhyte, T. Bailey: *Simultaneous localization and mapping: part I.* IEEE Robotics & Amp Amp Automation.

[3] A. Censi: *An ICP variant using a point-to-line metric.* IEEE International Conference on Robotics and Automation. IEEE (2008), 19–25.

[4] H. Alismail, L. D. Baker, B. Browning: *Continuous trajectory estimation for 3D SLAM from actuated LIDAR.* IEEE International Conference on Robotics and Automation. IEEE (2014), 6096–6101.

[5] W. Liu: *LIDAR-IMU Time Delay Calibration Based on Iterative Closest Point and Iterated Sigma Point Kalman Filter.* Sensors *17* (2017), No. 3, 539.

[6] W. Hess, D. Kohler, H. Rapp: *Real-time loop closure in 2D LIDAR SLAM.*IEEE International Conference on Robotics and Automation. IEEE (2016) 1271–1278.

[7] J. Zhang, S. Singh: *Low-drift and real-time LIDAR odometry and mapping.* Autonomous Robots (2016), 1–16.

[8] G. A. Kumar, A. K. Patil, R. Patil: *A LIDAR and IMU Integrated Indoor Navigation System for UAVs and Its Application in Real-Time Pipeline Classification.* Sensors *17* (2017), 6.

[9] E. López, S. García, R. Barea: *A Multi-Sensorial Simultaneous Localization and Mapping (SLAM) System for Low-Cost Micro Aerial Vehicles in GPS-Denied Environments.* Sensors *17* (2017), 4.

[10] R. POLI, J. KENNEDY, T. BLACKWELL: *Swarm intelligence, New York.* USA: Springer US (2017),3–57.

[11] F. Shabbir, P. Omenzetter: *Particle Swarm Optimization with Sequential Niche Technique for Dynamic Finite Element Model Updating.*Computer-Aided Civil and Infrastructure Engineering *30* (2015), No. 5, 359–375.

[12] G. Zhao, X. Xiao, J. Yuan: *Fusion of Velodyne and camera data for scene parsing.* International Conference on Information Fusion. IEEE (2012), 1172–1179.

[13] J. Zhang, S. Singh: *LOAM: LIDAR Odometry and Mapping in Real-time.* Robotics: Science and Systems Conference (2014).

[14] J. M. Będkowski, T. Röhling: *Online 3D LIDAR Monte Carlo localization with GPU acceleration.* Industrial Robot *44* (2017),4.

[15] X. Zhu, C. Qiu, A. Mark: *Terrain Inclination Aided Three- Dimensional Localization and Mapping for an Outdoor Mobile Robot.* International Journal of Advanced Robotic Systems *10* (2013), No. 1, 1.

[16] F. Pomerleau, F. Colas, R. Siegwart: *Comparing ICP variants on real-world data sets.* Autonomous Robots *34* (2013), No. 3, 133–148.

[17] M. L. Wang, M. L., Y. Li, Y. Yang: *Localization and mapping in urban area based on 3D point cloud of autonomous vehicles.*Journal of Beijing Institute of Technology (English Edition)*25* (2016), No. 4, 473–482.

[18] R. Havangi: *A mutated FastSLAM using soft computing.* Industrial Robot.(2017).

[19] X. Wang, C. Zhang, F. Liu: *Exponentially Weighted Particle Filter for Simultaneous Localization and Mapping Based on Magnetic Field Measurements.* IEEE Transactions on Instrumentation & Measurement *66* (2017), No. 7, 1658–1667.

[20] H. Xiong, Y. Chen, X. Li: *A Scan Matching Simultaneous Localization and Mapping Algorithm Based on Particle Filter.*Industrial Robot *43* (2016), No. 6, 607–616.

[21] R. Bogue: *Sensors for robotic perception. Part two: positional and environmental awareness.* Industrial Robot *42* (2015), No. 6, 502–507.

Received November 16, 2017