

<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

Lidar Scan-Matching for Mobile Robot Localization

Xia Yuan, Chun-Xia Zhao and Zhen-Min Tang
School of Computer Science and Technology,
Nanjing University of Science and Technology, Nanjing 210094, China

Abstract: Problem of mobile robot localization is usually solved by using GPS and INS system, but the system error of this kind of system has to be corrected by other sensors such as lidar. This study proposes an algorithm to do lidar scan-matching. The method employs a fuzzy clustering algorithm to segment points of lidar scans first and then do weight least-square linear fitting for each segment. Segments that satisfy linear distributed are picked out to calculate rotation between two lidar scans. Then the algorithm computes translation by calculating shifting of matched points. A principle called matching-range-point rule is used to find matching points belong to two scans. The characteristic of this proposed method it abandons iteration when calculate rotation and translation. It works fast and reliably and adapts to correct the error of GPS and INS system to localize a mobile robot accurately.

Key words: Robotics, localization, lidar, scan-matching

INTRODUCTION

Ground unmanned mobile robots should navigate in different environment (Meyrowitz *et al.*, 1996). Simultaneous Localization and Mapping (SLAM) is a basic problem for a mobile robot and it attracts lots of attentions since nineties of last century (Castellanos *et al.*, 2000; Dissanayake *et al.*, 2000; Bailey, 2002; Luo and Hong, 2004).

The robot understands the world outside depending on sensors it carries. When an unmanned robot reaches an unknown place two basic questions it has to answer: Where am I? and Where to go? A robot usually uses GPS combining INS system to answer the first question. Algorithms proposed by researchers to deal with the problem of localization including KF, EKF, UKF, MKF, MCL and MM, etc. The error of GPS and INS system is unavoidable. One more important thing is that in many cases there is no GPS signal and the error of INS system can be too large to be corrected without GPS. We use other sensors to correct result of localization in this situation. Matching points of lidar scans is a fast and good way to do localization accurately and we call this scan-matching in this paper. Iterative Closest Point (ICP) algorithm (Lu, 1995; Nieto *et al.*, 2007) is a common scan-matching method. The resolution of lidar is limited and it is impossible for us to match points of two scans one-to-one so the convergent speed or even if the ICP algorithm is convergent is uncertain sometimes.

We proposed a scan-matching algorithm based on geometry-matching which abandon iteration totally. It has

three mainly differences comparing to ICP: firstly, there is no iterative steps during computing translation and rotation between two lidar scans in our algorithm; secondly, our matching algorithm is based on geometry-matching which means we extract geometrical features from sampling points first and these features have less data quantity and will get more reliable result than finding corresponding points one-to-one from two scans; thirdly, we use fuzzy clustering method based on maximum entropy theory to do data segmentation.

APPROACH OF THE ALGORITHM

We give an overview of the algorithm first:

- (1) Clustering points of a lidar scan. Then the points will be split into some segments
- (2) For each segment of points we do weight least-square linear fitting to find linear distributed ones
- (3) Find matched linear segments from two lidar scans and calculate rotation angle between them
- (4) Find corresponding points from two lidar scans according to matching-range-point rule and calculate translation between these two scans
- (5) Do scan-matching of two lidar scans using rotation and translation value computed by step 3 and 4

Clustering of pints: We employ a fuzzy cluster method which based on Maximum Entropy Theory (MEP) to cluster points in a lidar scan (Liu and Meng, 2004; Yuan *et al.*, 2008). It is assumed the threshold of

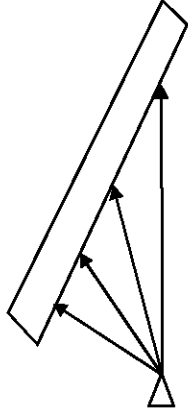


Fig. 1: Sampling points of lidar are asymmetrical

prediction error is δ , the next predicted lidar point's position from a same laser detector is (\hat{x}_i, \hat{y}_i) and the real measured position is (x_i, y_i) . The Euclidean distance between (\hat{x}_i, \hat{y}_i) and (x_i, y_i) is defined as:

$$e_i = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \quad (1)$$

The point will be considered to be the start of another trend if $e_i > \delta$ and it means this point will be a critical point between different clusters. Since sample points collected by lidar are asymmetrical as shown in Fig. 1 the parameter δ should be determined according to the distance between lidar and the target. We use Eq. 2 to settle this problem.

$$\begin{aligned} \delta &= \max \{\delta_i\} \\ \text{where } \delta_i &= 2a\sqrt{x_i^2 + y_i^2} \sin \theta \quad t = i-1, i, i+1 \end{aligned} \quad (2)$$

Parameter a in Eq. 2 is a scaling factor to counteract noise. Parameter θ is half of angular resolution of azimuth of HDL64-E. (x_i, y_i) is the coordinate of a point p_i .

Suppose there are N points in a lidar data frame and the system model can be characterized by M cluster centers $\bar{C}_1, \bar{C}_2, \dots, \bar{C}_M$ in an N -dimension space. Each cluster center \bar{C}_i can be represented by a vector that is composed of a pair of component vectors: the input vector \bar{Q}_i and the output vector \bar{Y}_i . For \bar{C}_i let the first p dimensions correspond to p input variables that constitute the input vector \bar{Q}_i and the other $m-p$ dimensions correspond to $m-p$ output variables that form the output vector \bar{Y}_i .

There will be a prediction error at each time when predict the position of the next point and the accumulated error problem is on the table if we ignore this error. In our algorithm, each vector in input vector \bar{Q}_i is presented as

$\bar{X}_i = [(x_i, y_i), e_i]$, where error item e_i take part in prediction process too and then minus the last predicted error in next prediction to reduce the effect of accumulate errors. Then our cluster centers are formed as follows:

$$\begin{aligned} \bar{C}_1 : \bar{Q}_1 &= [\bar{X}_{j-p-M+1}, \bar{X}_{j-p-M+2}, \dots, \bar{X}_{j-M}]^T \\ \bar{Y}_1 &= [\bar{X}_{j-M+1}]^T \\ \bar{C}_2 : \bar{Q}_2 &= [\bar{X}_{j-p-M+2}, \bar{X}_{j-p-M+3}, \dots, \bar{X}_{j-M+1}]^T \\ \bar{Y}_2 &= [\bar{X}_{j-M+2}]^T \\ &\vdots \\ \bar{C}_M : \bar{Q}_M &= [\bar{X}_{j-p}, \bar{X}_{j-p+1}, \dots, \bar{X}_{j-1}]^T \\ \bar{Y}_M &= [\bar{X}_j]^T \end{aligned} \quad (3)$$

Data acquired by lidar is presented in polar coordinates as (ρ, θ) . We converse it into grid coordinates first:

$$\begin{aligned} x_i &= \rho_i \cos \theta_i \\ y_i &= \rho_i \sin \theta_i \end{aligned} \quad (4)$$

For data point measured at time t , the probability of it belongs to the cluster centers can be viewed as its fuzzy membership μ_i , where, $\mu_i \in [0, 1]$ and $i = 1, 2, \dots, M$. The summation of all μ_i 's is equal to one, i.e.,

$$\sum_{i=1}^M \mu_i = 1 \quad \forall \mu_i \in [0, 1] \quad (5)$$

The clustering process can be formulated as an optimization problem and the corresponding cost function to be minimized is defined as:

$$E = \sum_{i=1}^M \mu_i \|P - \bar{X}_i\|^2 \quad (6)$$

where, \bar{X}_i is the input vector of fuzzy centers.

The distribution of μ_i is unknown in our problem. According to the information theory, the MEP is the most unbiased prescription to choose the values of the membership, μ_i , for which Shannon entropy, i.e., the expression:

$$H = - \sum_{i=1}^M \mu_i \ln \mu_i \quad (7)$$

is maximal under the constrains 7. This optimization problem can be reformulated as the maximization of the Lagrange:

$$L = H - \lambda E \quad (8)$$

where, λ is the Lagrange multiplier.

The final solved probabilities are of the Gibbs distribution (Rose *et al.*, 1990; Rose, 1998), i.e.,

$$\mu_i = \frac{\exp(-\lambda \|\bar{P} - \bar{X}_i\|^2)}{\sum_{i=1}^M \exp(-\lambda \|\bar{P} - \bar{X}_i\|^2)} \quad (9)$$

Take Eq. 10 into Eq. 6 and versus λ differentiate and then take the exponential parameter we get:

$$\lambda = 100M / \min(\|\bar{P} - \bar{X}_i\|^2) \quad (10)$$

Weighting reasoning mechanism is used to compute the out put vector as our system present a linear trend. The following formula will account for that trend well:

$$\bar{Y} = \sum_{i=1}^M w_i \mu_i \bar{Y}_i + \frac{1}{M-1} \sum_{i=1}^{M-1} (\bar{Y}_{i+1} - \bar{Y}_i) \quad (11)$$

where, w_i is the weighting term:

$$w_i = 1M / \sum_{i=1}^M 1 \quad (12)$$

Here, we have introduced the clustering algorithm we used in this study. This method does not require training which enables it to work online completely. It has the ability to trace the trend of points. From the experiment we can see fuzzy clustering gets better result than some simple clustering algorithms (Liu and Mang, 2004; Nieto *et al.*, 2007). After clustering we do linear fitting to find linear distributed segments.

Find linear distributed segment of points: Usually, we decide points in a segment are linear distributed or not by considering the value of mean μ and variance σ of the distance between points and the fitted line. But points sampled by lidar are asymmetrical so it is hard to decide the threshold of μ and σ to differentiate linear or nonlinear distribution.

We use a multi-time weight least-square linear fitting algorithm to find linear distributed segments. For each segment the fitting process contains three steps.

Firstly, we execute a regular least-square linear fitting algorithm on a segment pf points and get initial fitting line L_1 .

Secondly, the algorithm do weight least-square linear fitting based on L_1 . Suppose the function of a line is:

$$y = \beta_0 + \beta_1 x + \varepsilon \quad (13)$$

and the sum of square of least-square dispersion of its simple linear regression is:

$$Q_w(\beta_0, \beta_1) = \sum_{i=1}^n w_i (y_i - \beta_0 - \beta_1 x_i)^2 \quad (14)$$

it has been proved (He and Liu, 2007) that the weight least squares estimation of (β_0, β_1) will be:

$$\begin{cases} \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \\ \hat{\beta}_1 = \frac{\sum_{i=1}^n w_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n w_i (x_i - \bar{x})^2} \end{cases} \quad (15)$$

\bar{x} and \bar{y} in Eq. 15 is defined as:

$$\begin{cases} \bar{x} = \frac{1}{\sum w_i} \sum w_i x_i \\ \bar{y} = \frac{1}{\sum w_i} \sum w_i y_i \end{cases} \quad (16)$$

We compute the weight of a point as follow steps. Suppose there are two adjacent points P_i and P_{i+1} . The weight of P_i equals to $\overline{P_i P_{i+1}}$'s projection on L_1 as showing in Fig. 2. We normalize the weight of every point w_i that belongs to the same segment using (17):

$$w_i = \frac{\left| \overline{P_i P_{i+1}} \cdot \text{direction}(L_1) \right|}{\sum_{i=1}^n \left| \overline{P_i P_{i+1}} \cdot \text{direction}(L_1) \right|} \quad (17)$$

Symbol (L_1) in Eq. 17 denotes the direction of L_1 . Weight least-square linear fitting was executed after weight computing and then we get fitting line L_2 .

Thirdly, we recomputed the weight of each point. The new weight of P_i equals to $\overline{P_i P_{i+1}}$'s projection on L_2 . Then the algorithm take 80% points whose weight are bigger and dose the third times weight least-square linear fitting to get L_3 .

This linear fitting method has considered the direction between the fitting line and the line liked by two

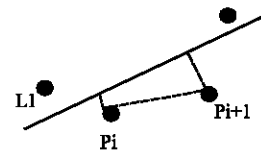


Fig. 2: Principle of computing weight of a point

neighbors. A point gets bigger weight if its direction is along the main distributed direction of the points in a segment.

Our experiment show values of slope of L_1 and L_3 are close if points in a segment satisfy linear distribution, otherwise the include angle of L_1 and L_3 is bigger than a threshold θ_{line} .

Matching linear distributed segments: We do scan-matching based on matching geometrical feature of linear distributed segments that extracted from two lidar scans. It is easy to calculate the include angle between two matched linear segments since linear distributed segments have been extracted. The angle of rotation between two scans can be calculated according to include angles of matched segments. To find matched linear segments one-to-one is easier than to find matched points one-to-one. Then the rotate angle can be computed once without iteration.

The key of the idea is how to find matched segments one-to-one reliably. The weight least-square linear fitting algorithm can ensure linear distributed segments are picked out and then we propose some rules to find matched linear distributed segments one-to-one.

There are little differences between two lidar scans as the data updating rate is so, fast that the robot moves a very short distance during two data scans. We believe two matched segments are close in position according to their own coordinate so we compute the middle point p_n ($n = 1, \dots, k$) of every linear distributed segments s_1, s_2, \dots, s_k . A threshold of distance $Dist$ is chosen. There are two matching rules should be satisfied simultaneously:

- Two linear distributed segments that belong to two scans are matched if the distance of their middle points is less than $Dist$
- The discrepancy of number of points contained in two segments is less than 20%:

$$Q = \sum_{m=1}^r (\alpha - \beta_m)^2 \quad (18)$$

Suppose there are r matched segments and the include angle of each pair of fitting line of matched segments is β_m ($m = 1, \dots, r$), then the rotation angle of two scans α can be calculated by minimizing Q in Eq. 18:

Match points between two scans: We introduce how to calculate translation of two lidar scans in this section. The method we used is called Matching-range-point rule (Lu, 1995) which is illustrated by Fig. 3.

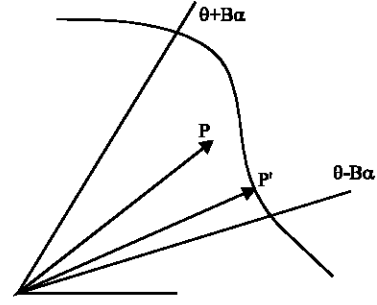


Fig. 3: Matching-range-point rule

Consider a data point P and its corresponding point $P = P\alpha + T$. If we ignore the translation, we have $P \approx P'$. On the other hand, the polar angle θ of P and the polar angle $\hat{\theta}$ of P' are related by:

$$\hat{\theta} \approx \theta + \alpha$$

This implies that the correspondence of P under a rotation is a point which has the same polar range as that of P and the polar angles of the corresponding point differ by the rotation angle α . Now in the presence of a small translation, we can still expect that the point P with the same range as that of P is possibly a good approximation to the true correspondence of P , and this approximate correspondence provides rich information about the rotation angle α .

To ensure that the rule finds a unique and reliable correspondence, we only search for the matching-range point within a local region of the model near P . Suppose that we can estimate a bound B_α for the rotation α , i.e., $|\alpha| < B_\alpha$. We have:

$$\hat{\theta} \in [\theta - B_\alpha, \theta + B_\alpha]$$

This means that P' should lie within the sector bounded by $\theta \pm B_\alpha$.

Therefore, the matching-range-point rule is expressed as follows: For a data point P , its corresponding point is P' on the model where, P' satisfies:

$$|\hat{\theta} - \theta| < B_\alpha$$

and

$$|P'| \approx |P|$$

$$O = \sum_{q=1}^g (T - t_q)^2 \quad (19)$$

Suppose there are g pair corresponding points between two lidar frames and the translation of each pair of points is t_q ($q = 1, \dots, g$), then the translation of two frames T can be calculated by minimizing O in Eq. 19.

Scan-matching: We can do scan-matching by rotate and transfer the coordinate between two scans. Suppose the data of k -th lidar scan is F_k , then the $(k+1)$ -th scan F_{k+1} will be:

$$F_{k+1} = F_k R + T, \text{ where, } R = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \quad (20)$$

RESULTS AND DISCUSSION

We use SICK LMS-291 lidar and its angle resolution is set to 1 degree. A GPS and INS system is used to locate the robot cursorily.

Discussion Results of clustering and linear fitting: Our clustering algorithm is better than simple ones. A simple algorithm may choose a distance threshold and points belong to two different segments when distance between two points is bigger than the threshold.

Figure 4a is the result of a simple clustering algorithm and Fig. 4b is followed by our method based on fuzzy

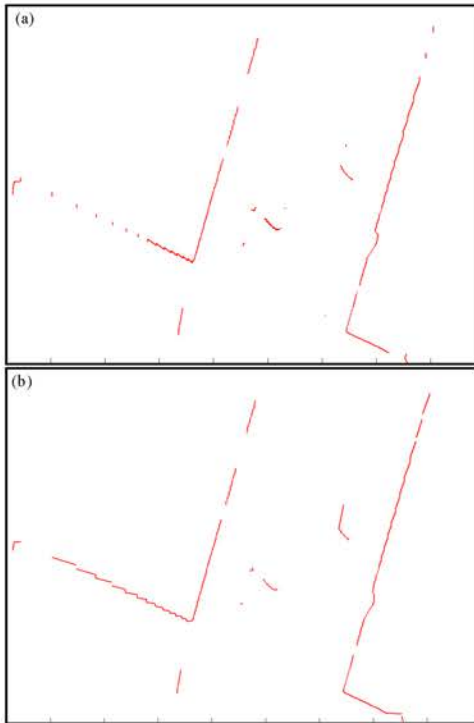


Fig. 4: Result of data clustering: (a) simple method and (b) proposed method

The result of weight least-square linear fitting is theory. We can see our method get better result than simple ones since it is better in tracing the trend of points. The result of simple method get more fragments than ours. illustrated by Fig. 5a. L_1 is shown in blue and L_3 in red. For linear distributed segments L_1 and L_3 are almost overlapped but there exist obvious include angle when points of the segment are nonlinear distributed. We choose $\theta_{line} = 1$ and the picked out linear distributed segments is shown in Fig. 5b.

Discussion Results of scan-matching: From Fig. 6a and 7a we can see the results of localization of GPS and INS system has obvious errors. By using our algorithm to refine the result of GPS and INS the error has been corrected a lot (Fig. 6c, 7b). When we calculate translation between two lidar scans the X coordinate is aligned with the moving direction of the robot.

We also tried an iterative based method to do scan-matching as shown in Fig. 8. In this method we find the nearest two points in two scans as matched points and then calculate the rotation and translation matrix by using least-square algorithm. The result is not well. We think one reason is value of rotation and translation is small and another reason is it is hard to find reliable corresponding points according to the nearest distance.

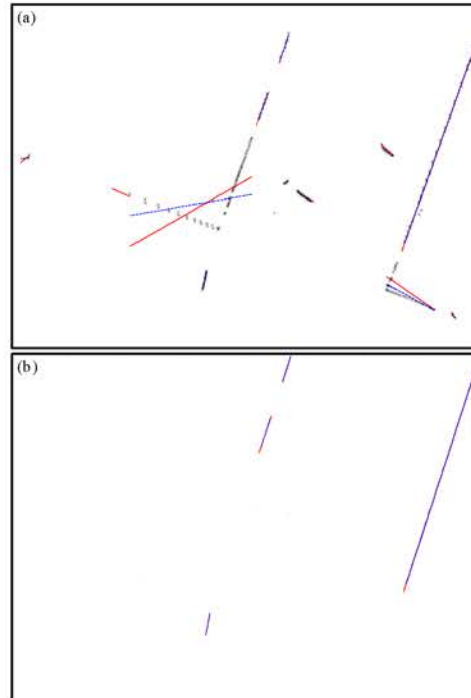


Fig. 5: (a) Result of weight least-square linear fitting. L_1 is Red and L_3 is Blue and (b) linear distributed segments picked out from a. Laser point is black dot

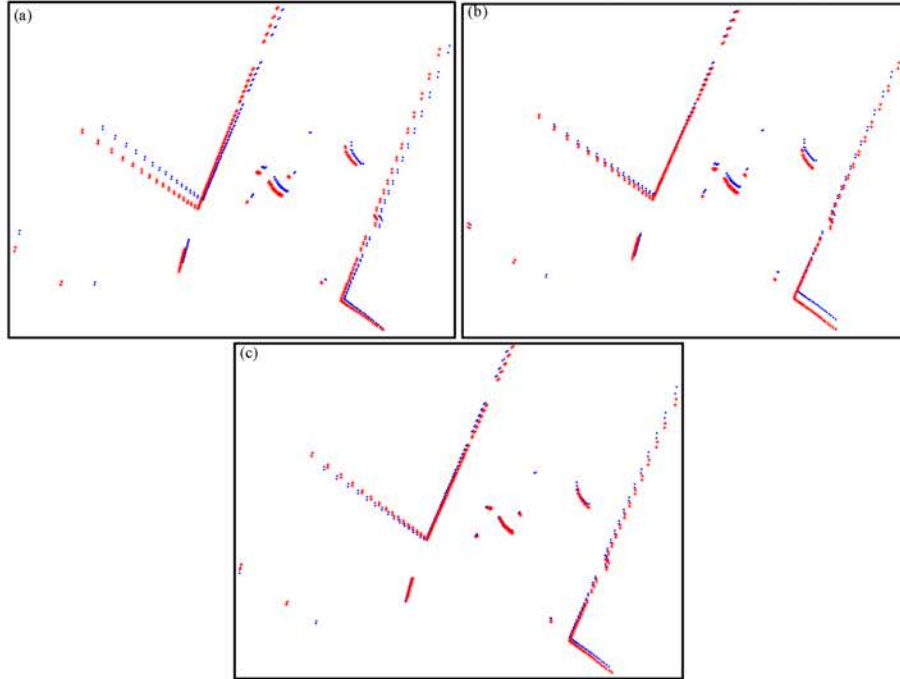


Fig. 6: Result of scan-matching around crossroad: (a) localization by GPS and INS, (b) result of localization after rotating based on (a), $\alpha = 2.4$, (c) result of localization after rotating and transferring based on (a), $\alpha = 2.4$, $T_x = 28.3$ cm, $T_y = 23.3$ cm (d) scan-matching using iteration method

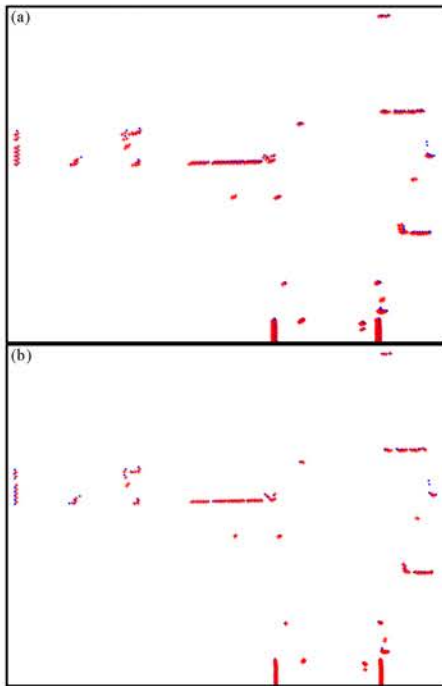


Fig. 7: Result of scan-matching on straight road: (a) localization by GPS and INS, (b) result of localization by our algorithm, $\alpha = 0.1$, $T_x = 33.2$ cm, $T_y = 10.2$ cm

We know the error is small when the robot moves straightly. Error becomes large when the curvature of the path is big like in crossroad area. Our experiments proved this situation. We can see the rotation angle is bigger in Fig. 6b than in Fig. 7b.

It is proved by Fig. 6c and 7b that our algorithm gets good result without iteration. The algorithm decouples the computation of rotation and translation. We use a reliable method to calculate the rotation. Traditional methods find corresponding points between two scan frames to calculate the rotation. But corresponding pair of points found by this way is unreliable so it has to do it as an iterative procedure to refine the result repeat. We extract lines from a lidar scan and calculate rotation by matching these lines between two scans. Linear feature is more reliable than laser sampling points so we can compute rotation without iteration and the result is exact enough for robot navigation task. But theoretically to compute rotation and translation is an iterative process. We try to compute translation by 5 times iterating and we can find in Table 1 that the result of the first time is good enough for robot navigation. So, there is no iteration in our algorithm.

To compare the cost factor of our method and iterative one we test run time of these. Our method cost 0.66 sec to segment points and do weight least-square

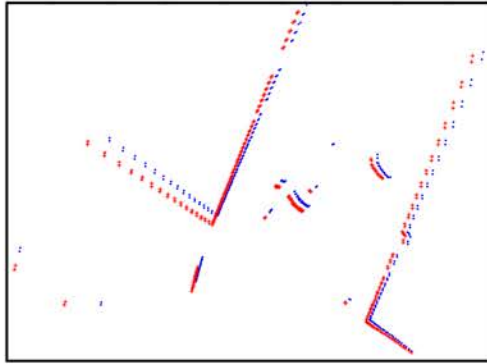


Fig. 8: Result of scan-matching using iterative method after 1000 iterations

Table 1: Compute translation by iterating 5 times

Iterating time	1	2	3	4	5
Tx (cm)	10.2	-0.7	-4.7e-15	-3.5e-15	-4.4e-15
Ty (cm)	33.2	-2.2	3.1e-14	2.7e-14	2.7e-14

linear fitting of a lidar scan and 0.09 sec to calculate rotation and translation. The iterative method showed in Fig. 8 costs 5.69 sec to calculate rotation and translation by 1000 iterations. The cost time is get in platform of matlab 7.1 without any code optimization so, it just for comparing these two kind of methods.

CONCLUSIONS

Traditional localization algorithms employ plenty filters to correct the error of GPS and INS system, but intrinsic system error can not be corrected by data filtering. So, we propose this lidar scan-matching method to help to correct the error of GPS and INS system. In actual implementation we may do scan-matching like every ten scans to counteract the cumulate error of system. This is helpful to improve efficiency.

We separate the computation of rotation and translation to avoid iteration to satisfy the requirement of fast computing of unmanned mobile robot navigation. The algorithm increase the reliability of the result of finding correspondence between two scans by matching geometrical feature instead of matching points directly to calculate the rotation.

Now we only use linear feature to match frames. There are not so many linear features in cross-country area so we will extract more geometrical features in our future work to improve the environmental adaptability of the algorithm.

REFERENCES

- Bailey, T.A., 2002. Mobile robot localization and mapping in extensive outdoor environments. Ph.D. Thesis, Australian Center for Field Robotics, Department of Aerospace, Mechanical and Mechatronic, Engineering, University of Sydney.
- Castellanos, J.A., J.M.M. Montiel, J. Neira and J.D. Tardós, 2000. Sensor influence in the performance of simultaneous mobile robot localization and map building. *Lecture Notes Control Inform. Sci.*, 250: 287-296.
- Dissanayake, W.M.G., P. Newman, H.F. Durrant-Whyte, S. Clark and M. Csorba, 2000. An experimental and theoretical investigation into simultaneous localization and map building. *Lecture Notes Control Inform. Sci.*, 250: 265-274.
- He, X.Q. and W.Q. Liu, 2007. Application of Regression Analysis. 2nd Edn., China Renmin University Press, China, ISBN: 9787300082356.
- Liu, P.X. and M.Q.H. Meng, 2004. Online data-driven fuzzy clustering with applications to real-time robotic tracking. *IEEE Trans. Fuzzy Syst.*, 12: 516-523.
- Lu, F., 1995. Shape registration using optimization for mobile robot navigation. Ph.D. Thesis, Graduate Department of Computer Science, University of Toronto.
- Luo, R.H. and B.M. Hong, 2004. The progress of simultaneous localization and mapping for mobile robot. *Robot*, 26: 182-186.
- Meyrowitz, A.L., D.R. Blidberg and R.C. Michelson, 1996. Autonomous vehicles. *Proc. IEEE*, 84: 1147-1164.
- Nieto, J., T. Bailey and E. Nebot, 2007. Recursive scan-matching slam. *Robotics Autonomous Syst.*, 55: 39-49.
- Rose, K., 1998. Deterministic annealing for clustering, compression, classification, regressions and related optimization problems. *Proc. IEEE*, 86: 2210-2239.
- Rose, K., E. Gurewitz and G.C. Fox, 1990. Statistical mechanics and phase transitions in clustering. *Phys. Rev. Lett.*, 65: 945-948.
- Yuan, X., C.X. Zhao, Y.F. Cai, H. Zhang and D.B. Chen, 2008. Road-surface abstraction using lidar sensing. *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision*, Dec. 17-20, Hanoi, Vietnam, pp: 1097-1102.