Registration of Point Clouds Based on the Ratio of Bidirectional Distances

Jihua Zhu[1], Di Wang[1], Xiuxiu Bai[1], Huimin Lu[2], Congcong Jin[1], Zhongyu Li[3]
1. Xi'an Jiaotong University, China
2. Kyushu Institute of Technology, Japan
3. University of North Carolina at Charlotte, USA
zhujh@mail.xjtu.edu.cn

Abstract

Despite the fact that original Iterative Closest Point (ICP) algorithm has been widely used for registration, it cannot tackle the problem when two point clouds are partially overlapping. Accordingly, this paper proposes a robust approach for the registration of partially overlapping point clouds. Given two initially posed clouds, it firstly builds up bilateral correspondence and computes bidirectional distances for each point in the data shape. Based on the ratio of bidirectional distances, the exponential function is selected and utilized to calculate the probability value, which can indicate whether the point pair belongs to the overlapping part or not. Subsequently, the probability value can be embedded into the least square function for registration of partially overlapping point clouds and a novel variant of ICP algorithm is presented to obtain the optimal rigid transformation. The proposed approach can achieve good registration of point clouds, even when their overlap percentage is low. Experimental results tested on public data sets illustrate its superiority over previous approaches on robustness.

1. Introduction

Due to the wide application in 3D reconstruction [11], shape recognition [6] and robot mapping [16, 13], point clouds registration has attracted immense attention in computer vision, pattern recognition and robotics. It addresses the problem of building up the correspondences and calculating the optimal transformation between two given point clouds. The most popular solution for registration is the iterative closest point (ICP) algorithm[1], which can iteratively build up point correspondences and calculate the rigid transformation by the minimization of residual errors. Although the original ICP algorithm is efficient and accurate, it can not deal with the registration of partially overlapping point clouds, which remains to be a challenging problem in many practical applications.

To address this issue, a straightforward solution is to reject point pairs including distances greater than the user-specified threshold [12]. Further, Godin [3] proposed to weight of point pairs by assigning lower weights to pairs with greater point-to-point distances. This approach is simple but unable to produce enough accurate results in most cases. Moreover, Chetverikov et al. [2] proposed the trimmed ICP (TrICP) algorithm, which introduced an overlap parameter into the least square function to automatically trim the outliers and can obtain accurate registration results for partially overlapping point clouds. Nevertheless, this approach is time-consuming. Accordingly, Phillips et al. [10] proposed the fractional TrICP (FTrICP) algorithm for much faster speed, which can compute the overlap parameter and the optimal transformation simultaneously. Besides, some probabilistic approaches[4, 14, 9, 7] were also proposed for registration of partially overlapping clouds. Although these probabilistic approaches are very accurate, they are always requiring huge computational resources. Recently, the concept of bidirectional distances[18, 17] was proposed to solve the non-rigid registration of absolutely overlapping point clouds. But it has not been applied to registration of partially overlapping point clouds. More recently, the correntropy was introduced into the registration problem [5, 15], which can be solved by maximizing the correntropy between two point clouds. Although it can deal with rigid registration with noises and outliers, its robustness should be further improved for registration of cloud pair with low overlap percentage.

Therefore, this paper proposes a robust approach for registration of partially overlapping point clouds. It firstly introduces the bilateral correspondence containing bidirectional distances, which express different characters for outliers and inliers. Based on the character of bidirectional distances, a probability value can be calculated and assigned for each point pair. Then, the registration problem can be also formulated into the weighted least square(LS) function. Finally, a novel variant of the ICP algorithm is proposed to solve the weighed LS function and can obtain the optimal

978-1-5090-5407-7/16 $31.00 © 2016 IEEE
DOI 10.1109/3DV.2016.85

rigid transformation. Experimental results illustrate that the proposed approach can achieve rigid registration of partially overlapping clouds with superior performances.

The remainder of this paper is organized as follows. In Section 2, the ICP algorithm is briefly reviewed. Section 3 presents the proposed approach with implementation details. Following that is section 4, in which the proposed approach is tested and evaluated on some public datasets. Finally, some conclusions are drawn in Section 5.

2. The ICP algorithm

Given two point clouds, denoted as the data shape $D = \{\vec{d}_i\}_{i=1}^{N_d}$ and the model shape $M = \{\vec{m}_j\}_{j=1}^{N_m}$, the goal of registration is to find the optimal rigid transformation $(\mathbf{R}, \vec{t})$, with which D can be in the best alignment with M. Accordingly, it can be formulated as the following least square (LS) problem:

$$\begin{aligned} \min_{\mathbf{R},\vec{t},c(i)\in\{1,\ldots,N_m\}} \sum_{i=1}^{N_d} \left\| \mathbf{R}\vec{d}_i + \vec{t} - \vec{m}_{c(i)} \right\|_2^2 , \\ \text{s.t.} \mathbf{R}^T\mathbf{R} = \mathbf{I}_{3\times3}, \det(\mathbf{R}) = 1 \end{aligned} \quad (1)$$

where $\mathbf{R} \in \mathbb{R}^{3\times3}$ is the rotation matrix, $\vec{t} \in \mathbb{R}^3$ indicates the translation vector, $\vec{m}_{c(i)}$ represents the correspondence of the $\vec{d}_i$ in the model shape, and $\| \|_2$ denotes L_2 norm. Eq. (1) can be solved by the ICP algorithm [1], which achieves the rigid registration by iterations. Given an initial transformation $(\mathbf{R_0}, \vec{t}_0)$, two steps are included in each iteration:

1) Establish correspondence for each point $\vec{d}_i$ in the data shape:

$$c_k(i) = \underset{j\in\{1,2,\ldots,N_m\}}{\arg\min} \left\| \mathbf{R_{k-1}}\vec{d}_i + \vec{t}_{k-1} - \vec{m}_j \right\|_2 \quad (2)$$

2) Compute the optimal transformation by minimizing the LS function:

$$(\mathbf{R_k}, \vec{t}_k) = \underset{\mathbf{R},\vec{t}}{\arg\min} \sum_{i=1}^{N_d} \left\| \mathbf{R}\vec{d}_i + \vec{t} - \vec{m}_{c_k(i)} \right\|_2^2 \quad (3)$$

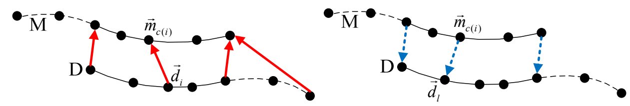
Although the original ICP algorithm has good performances for the rigid registration, it cannot achieve the registration of partially overlapping point clouds.

3. The proposed approach

In this section, an important observation is presented for point pair in the registration problem. Based on this observation, a new objective function can be designed for registration of partially overlapping clouds and then a variant of ICP algorithm is proposed to obtain the optimal solution.

3.1. An important observation

Given two partially overlapping point clouds, bilateral correspondence of one point in the data shape D can be established by applying the search method of nearest neighbour twice. As shown in Fig. 1, for each point $\vec{d}_i$ in the data shape D, its nearest neighbour $\vec{m}_c(i)$ with the forward distance can be searched from the model shape M. Then, for each of these nearest neighbours in M, its correspondence $\vec{d}_l$ with the backward distance can also be searched from D. There is an interesting phenomenon for the bilateral correspondence: in overlapping areas, the bilateral correspondence of one point in the data shape D is itself or its adjacent point, so the backward distance is equal or a little smaller than the forward one; in non-overlapping area, the bilateral correspondence of one point is far from itself, so the forward distance is much larger than the backward one.



(a) (b)

Figure 1. Bilateral correspondences between two partially overlapping shapes, where solid curve indicates the overlapping areas, real and dashed line arrows denote the forward and backward distances, respectively. (a) Nearest neighbours with forward distances for some points in D. (b) Correspondences with backward distances for some points in M, which are nearest neighbours of some points in D.

Therefore, the relationship of bidirectional distances can be viewed as an index to indicate whether one point is inlier or not. If bidirectional distances of one point are almost equal, then this point can be viewed as the inlier with a higher probability. Otherwise if the difference of bidirectional distances becomes larger, the probability should accordingly decrease.

3.2. The ICP algorithm based on the ratio of bidirectional distances

Based on the above observation, a new objective function can be proposed for the registration of partially overlapping point clouds as follows:

$$\min_{\mathbf{R},\vec{t}} \sum_{i=1}^{N_d} \left\| \mathbf{R}\vec{d}_i + \vec{t} - \vec{m}_{c(i)} \right\|_2^2 \cdot p_i, \quad (4)$$

where p_i denotes the value of probability, which is related to the ratio ρ_i of bidirectional distances and can be determined by the exponential function as:

$$p_i = e^{-\lambda(\rho_i - 1)}, \quad (5)$$

where λ is a preset parameter. As shown in Eq. (5), if bidirectional distances of one point are equal, it can be viewed as inlier with the highest probability. With the increase of the ratio ρ_i, the probability would be gradually decreased and its decreasing rate is related the value of λ. As this function is very similar to the function displayed in Eq. (1), it can be solved by a variant of the ICP algorithm, which includes three steps in each iteration:

1) Establish bilateral correspondence for each point $\vec{d}_i$ in the data shape:

$$c_k(i) = \underset{j\in\{1,..,N_m\}}{\arg\min} \left\|(\mathbf{R}_{k-1}\vec{d}_i + \vec{t}_{k-1}) - \vec{m}_j\right\|_2 \tag{6}$$

$$l = \underset{j\in\{1,..,N_d\}}{\arg\min} \left\|\vec{m}_{c_k(i)} - (\mathbf{R}_{k-1}\vec{d}_j + \vec{t}_{k-1})\right\|_2 \tag{7}$$

2) Calculate the ratio of bidirectional distances with its corresponding probability for each point pair:

$$\rho_{i,k} = \frac{\left\|\vec{d}_{i,k} - \vec{m}_{c_k(i)}\right\|_2}{\left\|\vec{m}_{c_k(i)} - \vec{d}_{l,k}\right\|_2} \tag{8}$$

$$p_{i,k} = \mathrm{e}^{-\lambda(\rho_{i,k}-1)} \tag{9}$$

where $\vec{d}_{i,k} = \mathbf{R}_{k-1}\vec{d}_i + \vec{t}_{k-1}$ and $\vec{d}_{l,k} = \mathbf{R}_{k-1}\vec{d}_l + \vec{t}_{k-1}$.

3) Compute the optimal transformation by minimizing the weighted LS function:

$$(\mathbf{R}_k, \vec{t}_k) = \underset{\mathbf{R},\vec{t}}{\arg\min}\left(\sum_{i=1}^{N_d}\left\|\mathbf{R}\vec{d}_i + \vec{t} - \vec{m}_{c_k(i)}\right\|_2^2 \cdot p_{i,k}\right). \tag{10}$$

Given good initial transformation, the optimal solution can be obtained by iteratively performing the above three steps until convergence criteria are satisfied.

Obviously, Step 2 can be solved directly and Step 1 denotes the nearest neighbour problem, which can be efficiently solved by the search method based on $k-d$ tree. Therefore, the solution to the problem in Step 3 will be demonstrated in the following section.

3.3. Computation of the rigid transformation

In this section, the closed-form solution will be presented for the computation of rigid transformation $(\mathbf{R}, \vec{t})$.

The objective function depicted in Eq. (10) is a quadratic form and can be extended as following:

$$\begin{aligned} J(\mathbf{R},\vec{t}) = \sum_{i=1}^{N_d} &[(\mathbf{R}\vec{d}_i - \vec{m}_{c_k(i)})^T(\mathbf{R}\vec{d}_i - \vec{m}_{c_k(i)})p_{i,k} \\ &+2\vec{t}^T(\mathbf{R}\vec{d}_i - \vec{m}_{c_k(i)})p_{i,k} + \vec{t}^{\mathrm{T}}\cdot\vec{t}\cdot p_{i,k}] \end{aligned} \tag{11}$$

Taking the derivative of J with respective to $\vec{t}$, we can get the following result:

$$\frac{\partial J}{\partial \vec{t}} = \sum_{i=1}^{N_d}\left[2(\mathbf{R}\vec{d}_i - \vec{m}_{c_k(i)})p_{i,k} + 2\vec{t}\cdot p_{i,k}\right] \tag{12}$$

Let $\frac{\partial J}{\partial \vec{t}} = 0$, the translate vector can be obtained as:

$$\vec{t}_k = -\frac{\sum\limits_{i=1}^{N_d}(\mathbf{R}\vec{d}_i - \vec{m}_{c_k(i)})p_{i,k}}{\sum\limits_{i=0}^{N_d}p_{i,k}} \tag{13}$$

In Eq. (11), $\vec{t}$ can be replaced by Eq. (13) and the objective function can be simplified as:

$$J(\mathbf{R}) = \sum_{i=1}^{N_d}\|\mathbf{R}\vec{x}_i - \vec{y}_i\|_2^2 \cdot p_{i,k} \tag{14}$$

where $\vec{x}_i = \vec{d}_i - \sum\limits_{i=1}^{N_d}p_{i,k}\vec{d}_i \Big/ \sum\limits_{i=1}^{N_d}p_{i,k}$ and $\vec{y}_i = \vec{m}_{c_k(i)} - \sum\limits_{i=1}^{N_d}p_{i,k}\vec{m}_{c_k(i)} \Big/ \sum\limits_{i=1}^{N_d}p_{i,k}$. Subsequently, the rotation matrix can be calculated by minimizing the function $J(\mathbf{R})$, which can be expanded as:

$$\begin{aligned} \mathbf{R} &= \underset{\mathbf{R}}{\arg\min}\sum_{i=1}^{N_d}(\vec{x}_i^T\vec{x}_i + \vec{y}_i^T\vec{y}_i - 2\vec{x}_i^T\mathbf{R}\vec{y}_i)\cdot p_{i,k} \\ &= \underset{\mathbf{R}}{\operatorname{argmax}}\sum_{i=1}^{N_d}(\vec{x}_i^T\mathbf{R}\vec{y}_i p_{i,k}) \end{aligned} \tag{15}$$

The optimization problem depicted in Eq. (15) has been solved by Myronenko [8], so the following conclusion can be directly presented as follows:

1) Compute the 3×3 matrix $\mathbf{H}$ and its singular value decomposition (SVD) results:

$$\mathbf{H} = \sum_{i=1}^{N_d}\vec{y}_i p_{i,k}\vec{x}_i^T \tag{16}$$

$$[\mathbf{U}, \Lambda, \mathbf{V}] = \mathrm{SVD}(\mathbf{H}) \tag{17}$$

2) Calculate the rotation matrix:

$$\mathbf{R}_k = \mathbf{V}\mathbf{D}\mathbf{U}^T \tag{18}$$

where $\mathbf{D} = diag(1, 1, \det(\mathbf{V}\mathbf{U}^T))$.

3.4. Implementation details

To establish the correspondence, the proposed approach adopts the search method of nearest neighbour based on the $k-d$ tree. For the transformed point $(\mathbf{R}_{k-1}\vec{d}_i + \vec{t}_{k-1})$, as shown in Eq. (6), its nearest neighbour can be searched from the model shape M in each iteration. Therefore, the $k-d$ tree for M can be built before starting iteration and it only requires to be built once. Meanwhile, for the point $\vec{m}_{c_k(i)}$, as shown in Eq.(7), its nearest neighbour should also be searched from the transformed data shape $T_{k-1}(D)$ in each iteration. Since the rigid transformation $T_{k-1} = (\mathbf{R}_{k-1}, \vec{t}_{k-1})$ is changed in each iteration, the $k-d$ tree for $T_{k-1}(D)$ requires be continuously rebuilt in each iteration. This is time-consuming. To improve efficiency, Eq. (7) can be modified as follows:

$$l = \underset{i\in\{1,\ldots,N_d\}}{\arg\min} \left\| (\mathbf{R}_{k-1}^{-1}\vec{m}_{c_k(i)} - \mathbf{R}_{k-1}^{-1}\vec{t}_{k-1}) - \vec{d}_i \right\|_2 \quad (19)$$

Obviously, Eq. (7) and (19) can obtain the same result. In each iteration, the search method is firstly utilized to find the nearest neighbour $\vec{m}_{c_k(i)}$ for each point $\vec{d}_i$ in the data shape D. Subsequently, the rigid transformation $(\mathbf{R}_{k-1}^{-1}, -\mathbf{R}_{k-1}^{-1}\vec{t})$ can be imposed to all nearest neighbours $\vec{m}_{c_k(i)}$. Then, for the transformed point $(\mathbf{R}_{k-1}^{-1}\vec{m}_{c_k(i)} - \mathbf{R}_{k-1}^{-1}\vec{t}_{k-1})$, as shown in Eq.(19), its nearest neighbour should only be searched from the raw data shape D in each iteration. Therefore, the $k-d$ tree of D can be built before starting iteration and it only requires to be built once.

Usually, the ratio of bidirectional distances is expected to meet the condition $\rho_{i,k} \in [1, +\infty)$. Sometimes, the point pair established by Eq. (6) or Eq. (19) may be coincident due to the round-off effect. In that case, $\rho_{i,k}$ will be illegal. Therefore, Eq. (8) should be modified as follows:

$$\rho_{i,k} = \frac{\left\| \vec{d}_{i,k} - \vec{m}_{c_k(i)} \right\|_2 + \delta}{\left\| \vec{m}_{c_k(i)} - \vec{d}_{l,k} \right\|_2 + \delta} \quad (20)$$

where δ represents a small positive constant.

4. Experimental results

To verify its superior performances, the proposed approach was compared with the ICP algorithm with weighting of pairs (wICP)[12], the fractional TrICP algorithm (FTrICP) [10] and the correntropy ICP algorithm (CICP)[15]. Experiments were tested on the Stanford Repository [1]. During experiment, some parameters were set as follows : $\lambda = 6$ and $\delta = 10^{-6}$. All competed approaches used the search method of nearest neighbour based on k-d tree to establish correspondences and were implemented in Matlab. All experiments were performed on the same computer.

4.1. Data simulation

For simulation comparison, the Stanford Bunny was selected and displayed in Fig. 2. To generate the simulated datasets, 5% points in the original shape were randomly deleted and the shape was cut down by one part, which includes n range points. Then the remaining shape was added with Gaussian zero mean noise and assigned to be the data shape. Further, 5% points in the original shape were randomly deleted and a small transformation $(\mathbf{R}_r, \vec{t}_r)$ was randomly generated and applied to the remaining shape, which became a transformed one. Finally, the transformed shape was cut down by other n range points and turned to be the model shape. In that case, the overlap percentage of these two shapes can be calculated as:

$$\xi = 0.95 \times (0.95N - 2n)/(0.95N - n), \quad (21)$$

where N denotes the point number of the original shape. With different value of n, the overlap percentage can be changed and simulated datasets can be aligned by all competed approaches, which can get corresponding results for comparison. For comparison, relative errors can be defined as $\varepsilon_{\mathbf{R}} = \|\mathbf{R}_m - \mathbf{R}_r\|_F$ and $\varepsilon_{\vec{t}} = \left\|\vec{t}_m - \vec{t}_r\right\|_2$, where $(\mathbf{R}_m, \vec{t}_m)$ represents the estimated transformation, and $\|\|_F$ denotes Frobenius norm. Denote d as the average point res-

[1] The stanford 3D scanning repository. http://graphics.stanford.edu/data/3dscanrep/.

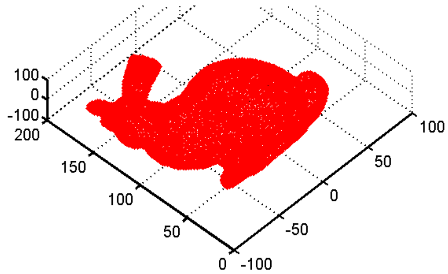
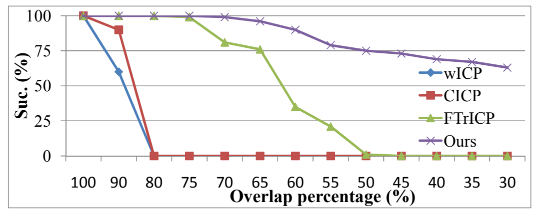
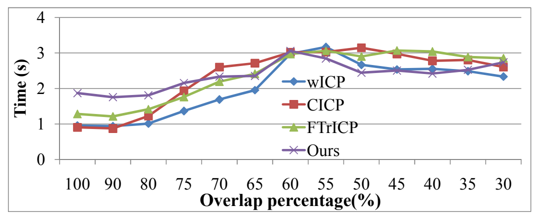


Figure 2. The original Bunny shape with 8171 range points.



(a)



(b)

Figure 3. Performance comparison under varied overlap percentages. (a)Success rate. (b)Average runtime.

Table 1. Performance comparison of all competed approaches tested on three shapes

Dataset	Overlap	N_m	$d(\times 10^{-3})$	N_d	Term	wICP	CICP	FTrICP	Ours
Bunny	46.6%	8052	1.0030	7548	$\varepsilon_{\mathbf{R}}$	0.4333	0.5202	0.3605	0.0413
					$\varepsilon_{\vec{t}}$	32.9218	40.3512	27.9069	2.9847
					$T(s)$	1.885	1.362	1.740	1.764
Dragon	76.3%	7740	0.8244	10323	$\varepsilon_{\mathbf{R}}$	0.0040	0.1639	0.0312	0.0031
					$\varepsilon_{\vec{t}}$	0.6770	9.6742	3.4659	0.6583
					$T(s)$	2.4031	1.8573	1.5440	2.6303
Buddha	86.9%	17008	0.5473	16940	$\varepsilon_{\mathbf{R}}$	0.0035	0.0484	0.0702	0.0028
					$\varepsilon_{\vec{t}}$	0.6983	1.6135	1.5008	0.5537
					$T(s)$	3.1062	3.1628	4.7351	4.7017

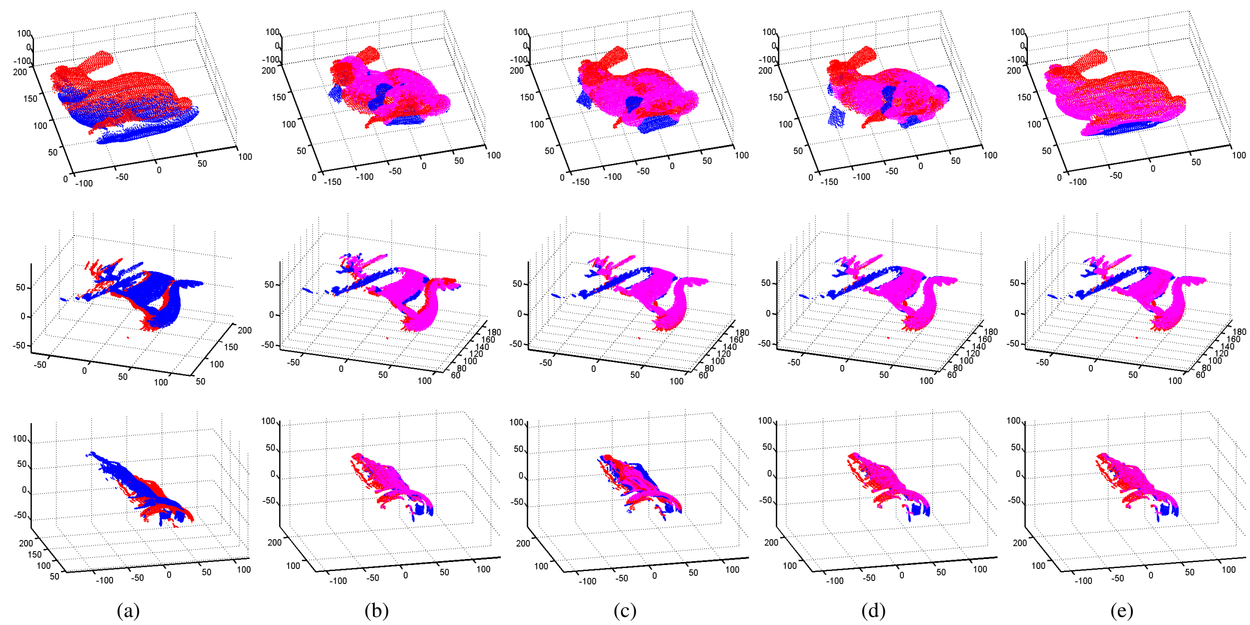


Figure 4. Registration results of all competed approaches tested on three shapes, where overlapping areas are denoted in the magenta. (a) The model shape (red) and data shape (blue) to be registered. (b) wICP results. (c) CICP results. (d) TrICP results. (c) Our results.

olution of model shape, then the registration process can be viewed as successful if and only if $(\varepsilon_{\mathbf{R}} \leq 0.01\ and \varepsilon_{\vec{t}} \leq d)$. To eliminate randomness, all approaches were tested on each shape pairs with varied overlap percentages for 100 Monte Carlo (MC) trials. Subsequently, registration performance for all competed approaches are displayed in Fig. 3.

As shown in Fig. 3(a), both the wICP algorithm and the CICP algorithm can only obtain good results for registration of point clouds with high overlap percentage. Besides, the FTrICP algorithm can achieve registration of partially overlapping point clouds, but its robustness decreases quickly with the decrease of overlap percentage. However, our approach can achieve good registration of partially overlapping point clouds and its robustness fades slowly with the decrease of overlap percentage. Therefore, our approach is the most robust one among these competed approaches. In terms of efficiency, as presented in Fig. 3(b), our approach is comparable to other registration approaches.

4.2. Experiment comparison

In this section, all competed approaches were tested on three real datasets, whose details are depicted in Table 1. In addition to point clouds, these datasets also include the ground truth of rigid transformation $(\mathbf{R}_g, \vec{t}_g)$ between a pair of point clouds to be registered.

Before registration, the ground truth of rigid transformation was imposed by a perturbed rigid transformation $(\mathbf{R}_r, \vec{t}_r)$, where the Euler angle and translate component were randomly drawn from the uniform distribution $\mathcal{U}[-10\mathbf{deg}, 10\mathbf{deg}]$ and $\mathcal{U}[-10d, 10d]$, respectively. Here, the symbol d denotes the average point resolution of the model shape. Then, all competed approaches can take the rigid transformation $(\mathbf{R}_r \cdot \mathbf{R}_g, \vec{t}_r + \vec{t}_g)$ as initial parameters for registration of point clouds and obtain the estimated transformation $(\mathbf{R}_m, \vec{t}_m)$. For comparison, relative errors can be defined as $\varepsilon_{\mathbf{R}} = \|\mathbf{R}_m - \mathbf{R}_g\|_F$ and $\varepsilon_{\vec{t}} = \frac{\|\vec{t}_m - \vec{t}_g\|_2}{d}$. To eliminate randomness, all approaches were tested on each datasets with 20 MC trials and their registration results were recorded. Subsequently, the average error and runtime for all competed approaches are displayed in Table 1. To view results in a more intuitive way, Fig. 4 displays registration results for all competed approaches tested on three datasets.

As shown in Table 1 and Fig. 4, all approaches can achieve good registration of the Stanford Buddha, which has high overlap percentage. Since our approach requires the application of the search method twice in each iteration, it is less efficient than other three related approaches. For the Stanford Dragon with moderate percentage, both the wICP algorithm and the CICP algorithm fail to obtain good registration results, but the proposed approach can obtain accurate result comparable with the FTrICP algorithm. For the Stanford Bunny with low overlap percentage, all the other three algorithms converge to local minimum and only the proposed approach can achieve good registration. Therefore, the proposed approach is the most robust one among all these competed approaches.

5. Conclusions

This paper proposes a novel approach for registration of partially overlapping point clouds. The novelty of this approach is the introduction of bilateral correspondence, which includes both the forward distance and the backward distance. By the consideration of bidirectional distances, each point pair can be assigned a probability, which can be viewed as an index to indicate whether this point pair is inlier or not. Subsequently, a new objective function can be designed for registration of partially overlapping point clouds and a variant of the ICP algorithm is proposed to calculate the optimal rigid transformation. The proposed approach has been tested on public datasets and experimental results illustrate the proposed approach can achieve the registration of partially overlapping clouds with good robustness and without losing too much efficiency.

6. Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant nos. 61573273, 61573280 and 61503300.

References

[1] P. J. Besl and N. D. McKay. A method for registration of 3d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[2] D. Chetverikov, D. Stepanov, and P. Krsek. Robust euclidean alignment of 3d point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing*, 23(3):299 – 309, 2005.

[3] G. Godin, M. Rioux, and R. Baribeau. Three-dimensional registration using range and intensity information. In *Proc. SPIE*, volume 2350, pages 279–290, 1994.

[4] S. Granger and X. Pennec. Multi-scale EM-ICP: A fast and robust approach for surface registration. In *Proc. of European Conf. on Computer Vision(ECCV)*, pages 418–432, Copenhagen, Denmark, 2002.

[5] E. Hasanbelliu, L. S. Giraldo, and J. C. Principe. Information theoretic shape matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(12):2436–2451, Dec 2014.

[6] D. Held, S. Thrun, and S. Savarese. Robust single-view instance recognition. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016.

[7] B. Jian and B. C. Vemuri. Robust point set registration using gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1633–1645, Aug 2011.

[8] A. Myronenko and X. Song. On the closed-form solution of the rotation matrix arising in computer vision problems. *ArXiv e-prints*, Apr. 2009.

[9] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275, 2010.

[10] J. M. Phillips, R. Liu, and C. Tomasi. Outlier robust icp for minimizing fractional rmsd. In *Proc. 3DIM*, pages 427–434, Aug 2007.

[11] N. Richard, F. Dieter, and S. Steve. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 343–352. IEEE, 2015.

[12] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proc. 3DIM*, pages 145–152, Quebec, Canada, 2001.

[13] T. Shiratori, J. Berclaz, M. Harville, C. Shah, T. Li, Y. Matsushita, and S. Shiller. Efficient large-scale point cloud registration using loop closures. In *3D Vision (3DV), 2015 International Conference on*, pages 232–240. IEEE, 2015.

[14] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *Proc. of European Conf. on Computer Vision(ECCV)*, pages 558–569, 2004.

[15] G. Xu, S. Du, and J. Xue. Precise 2d point set registration using iterative closest algorithm and correntropy. In *Int'l Joint Conf. on Neural Networks(IJCNN)*, July 2016.

[16] F. Yu, J. Xiao, and T. Funkhouser. Semantic alignment of lidar data at city scale. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1722–1731. IEEE, 2015.

[17] J. Zhu, S. Du, Z. Yuan, Y. Liu, and L. Ma. Robust affine iterative closest point algorithm with bidirectional distance. *IET Computer Vision*, 6(3):252–261, May 2012.

[18] J. H. Zhu, N. N. Zheng, Z. J. Yuan, S. Y. Du, and L. Ma. Robust scaling iterative closest point algorithm with bidirectional distance measurement. *Electronics Letters*, 46(24):1604–1605, November 2010.