Parallel registration algorithm with arbitrary affine transformation

Qin Shu (舒 勤)[1], Xiuli He (何秀丽)[1], Chang Wang (王 畅)[1,*], and Yunxiu Yang (杨赟秀)[2]

[1]*College of Electrical Engineering, Sichuan University, Chengdu 610065, China*

[2]*Southwest Institute of Technical Physics, Chengdu 610041, China*

**Corresponding author: 745472485@qq.com*

Received April 11, 2020; accepted April 30, 2020; posted online June 15, 2020

The point clouds scanned by a 3D laser scanner may be affine transformed when the size and posture of the objects being scanned are different. This type of problem is common, but few algorithms can solve it. Therefore, this Letter proposes a parallel registration algorithm. The algorithm eliminates the effects of the affine matrix in the point cloud, based on a simple whitening operation. Moreover, it also has strong anti-noise performance. The algorithm proposed in this Letter is not only simple in structure, but also shows excellent effects in practical applications and simulations.

Keywords: point cloud; affinity; parallel registration.
doi: 10.3788/COL202018.071001.

Registration technology is an important digital detection technology that is used in many fields such as NDT (nondestructive testing), pattern recognition, virtual reality, robots, and related fields[1,2]. At present, global registration[3–6] and local registration[7–9] are the two main research directions. Since the scale of the point clouds may be inconsistent, scale registration[10–12] is proposed based on global registration. The scale registration algorithm generally introduces a scale coefficient based on the global registration algorithm. However, in practical engineering applications, due to different types of three-dimensional laser scanners, thermal expansion of the objects being scanned, and differences in the types of scanned objects, point clouds may be arbitrarily affine transformed. This problem is very common, for example, the relationship between iphone 6 and iphone 6 plus is an affine transformation. Therefore, we focus on the problem of arbitrary affine point clouds registration.

Besl *et al.*[3] proposed the classic iterative closest point (ICP) algorithm in 1992, which is commonly used for point set registration, but it is not suitable for affine registration. On the basis of ICP, many scholars are committed to improving the ICP algorithm in order to complete a more complex point set registration. Ying *et al.*[10] put forward a scale registration algorithm that introduced an affine factor and searched for seven undetermined variables to register point clouds on different scales. In Ref. [11], Makovetskii *et al.* presented an affine registration algorithm based on a point-to-plane approach, which is used to register point clouds that are affine transformed in three fixed directions. The methods[12,13] solved the same problem like in Ref. [11]. A multi-scale affine registration algorithm[14] can only be used for image registration. In Ref. [15], an affine registration algorithm of point sets using ICP and ICA (independent component analysis) was proposed. Indeed, the algorithm can solve the affine registration problem. However, both ICA and ICP require iterative operations, and the ICA algorithm requires the order of point sets. Some scholars also presented a correntropy-based method to improve the affine ICP[16,17]. Those algorithms can obtain good results on some simple affine registrations while they cannot register point clouds with an arbitrary affine transformation.

Some algorithms[18–22] based on probability estimation have been developed to solve the nonrigid registration problem. They use Gaussian mixture models (GMMs); the coherent point drift (CPD) algorithm[18] is a classic example that is widely used for point cloud registration. These algorithms can achieve rigid and nonrigid registration for point clouds; they avoid the iteration of the ICP, but the probability estimation algorithm relies on the EM (expectation maximization algorithm). The EM iteration process may consume more time than the ICP. More recently, Ma *et al.*[23] proposed the global and local structure preserving point set registration (PRGLS) algorithm, which improved CPD by obtaining a binary corresponding matrix that updates the transformation under preserving global and local structures. Zhang *et al.*[24] also used the method of the global-local structural preservation to solve a nonrigid point set registration. In addition, Ma *et al.*, who keep focusing on spatial structure preservation, proposed locality preserving matching (LPM)[25] and they put forward the MR-RPM algorithm[26] by learning nonrigid transformation between two given point sets with manifold regularization to complete the registration. It has to be said that these methods have achieved good registration results, but they take a relatively long time to complete the registration when they are applied to large-scale 3D point cloud registration.

As in engineering applications, arbitrary affine phenomena are common in point cloud registration, and given that there are currently few affine algorithms to solve such problems, and some classic nonrigid registration algorithms either run for a long time or have higher requirements for operating equipment, this Letter focuses on how to solve the relationship between two given point sets in a simple way and how to accurately and quickly estimate the affine transformation. Finally, we propose a simple structured and fast affine registration algorithm.

The contributions of this Letter include the following three aspects. First, the whitening operation is introduced to the point set registration problem to find the relationship between the two point sets, which makes the affine registration problem be a rigid registration problem. Second, in the relationship after the whitening operation, the global vector features of point clouds are introduced and combined with the least squares method to achieve a fast coarse registration. Third, fast registration is provided for the proposed method using parallel computing, especially for large-scale 3D point clouds, which means the algorithm can work on low-performance computers.

Let a target point cloud $\boldsymbol{\Omega}$ be the point set $\{p_1, p_2, ..., p_n\}$ and a source point cloud Γ be the point set $\{q_1, q_2, ..., q_n\}$ in R^3. Furthermore, assume that the relationship between points in $\boldsymbol{\Omega}$ and Γ is one-to-one mapping. Without loss of generality, a reversible operator is defined as

$$T\bullet = A\bullet + p_0, \tag{1}$$

where the affine matrix $A \in R^3$, $\text{Rank}(A) = 3$ and the translation vector $p_0 = (x_0, y_0, z_0)^T$.

The registration problem of the two point clouds is to find the reversible operator $T\bullet$ so that $T(p_k)$ is infinitely close to q_k. Then the corresponding point between p_k and q_k is as follows:

$$q_k = T(p_k). \tag{2}$$

How to obtain the reversible operator is the key point in registration, and many researchers describe the problem using a cost function

$$J(T) = \sum_{k=1}^{n} \| T(p_k) - q_k \|^2, \tag{3}$$

where $\|\cdot\|$ represents 2-norm, and further minimizes the cost function to obtain an estimate of the reversible operator

$$\hat{T} = \arg\min_T J(T). \tag{4}$$

Let the target point cloud be matrix X, and the source point cloud be matrix Y and $X = \{p_1, p_2, ..., p_n\}$, $Y = \{q_1, q_2, ..., q_m\}$. Then,

$$Y \leftrightarrow AX + p_0 I_n, \tag{5}$$

where $I_n \in R^{1\times n}$ is the vector whose elements are all 1, and $\leftrightarrow$ indicates that two point clouds have been registered. Since the relationship between point clouds does not correspond exactly, $=$ cannot be used to describe their relationship.

We first remove the mean of the point clouds and ensure that the center of the point clouds is at the origin of the coordinate space. After the mean value is removed, the point clouds are as follows:

$$\begin{cases} \tilde{Y} = Y - \left(\frac{1}{m}\sum_{k=1}^{m} q_k\right) I_m, \\ \tilde{X} = X - \left(\frac{1}{n}\sum_{k=1}^{n} p_k\right) I_n. \end{cases} \tag{6}$$

Equation (7) can be obtained by the principal component analysis (PCA) method:

$$\begin{cases} \frac{1}{m}\tilde{Y}\tilde{Y}^T = Q_Y \Lambda_Y Q_Y^T, \\ \frac{1}{n}\tilde{X}\tilde{X}^T = Q_X \Lambda_X Q_X^T. \end{cases} \tag{7}$$

In fact, $\tilde{Y} \leftrightarrow A\tilde{X}$. According to Eq. (7) and statistical principles, Eq. (8) can be obtained:

$$\frac{1}{n} A\tilde{X}\tilde{X}^T A^T = \frac{1}{m}\tilde{Y}\tilde{Y}^T. \tag{8}$$

Then, Eq. (8) can be rewritten as

$$A Q_X \Lambda_X Q_X^T A^T = Q_Y \Lambda_Y Q_Y^T. \tag{9}$$

Equation (9) can be decomposed into two parts. Since the decomposition is not unique, there must be an orthogonal matrix R that satisfies

$$A Q_X \Lambda_X^{\frac{1}{2}} = Q_Y \Lambda_Y^{\frac{1}{2}} R. \tag{10}$$

In addition, the two sets of point clouds can be whitened as follows:

$$\begin{cases} \hat{Y} = \Lambda_Y^{-\frac{1}{2}} Q_Y^T \tilde{Y}, \\ \hat{X} = \Lambda_X^{-\frac{1}{2}} Q_X^T \tilde{X}. \end{cases} \tag{11}$$

By Eqs. (5), (10), and (11),

$$R\hat{X} \leftrightarrow \hat{Y}, \tag{12}$$

where the relationship between $\hat{X}$ and $\hat{Y}$ is a rigid transformation.

At present, many algorithms[3–7,14,15] can solve the registration problem in Eq. (12). Here, we give a fast algorithm to solve it. Define the global structure feature and introduce a continuous bounded nonlinear real function cluster

$$\{g_1, g_2, ..., g_k\}, \tag{13}$$

where $0 < M_1 \le g_k \le M_2$, and M_1 and M_2 are the infimum and supremum of g_k, respectively. The global vector features of point clouds $\hat{X}$ and $\hat{Y}$ are defined as

$$\begin{cases} C_X^j = \frac{1}{n}\sum_{k=1}^{n} g_j(\|\hat{x}_k\|)\hat{x}_k, \\ C_Y^j = \frac{1}{m}\sum_{k=1}^{m} g_j(\|\hat{y}_k\|)\hat{y}_k, \end{cases} \tag{14}$$

where $\hat{x}_k$ and $\hat{y}_k$ denote the kth column vector of $\hat{X}$ and $\hat{Y}$, respectively. After taking the weighted average of the data, some features of the point cloud become more stable. According to statistical principles,

$$C_Y^j = \frac{1}{n}\sum_{k=1}^{m} g_j(\|R\hat{x}_k\|)R\hat{x}_k. \tag{15}$$

According to the linear operator theory, the orthogonal matrix R is a unitary matrix. Then,

$$\begin{cases} \|R\| = 1, \\ \|Rx\| = \|x\|, \quad \forall x \in R^3. \end{cases} \tag{16}$$

Clearly, Eq. (16) can be rewritten as

$$C_Y^j = RC_X^j. \tag{17}$$

For convenience, the characteristic matrices for two point clouds are defined as

$$\begin{cases} X_e = (C_X^1, C_X^2, ..., C_X^k) \\ Y_e = (C_Y^1, C_Y^2, ..., C_Y^k) \end{cases}, \quad k > 2. \tag{18}$$

Then, the following equation can be derived from Eqs. (17) and (18):

$$RX_e = Y_e. \tag{19}$$

According to the least squares method,

$$R = Y_e X_e{}^T (X_e X_e{}^T)^{-1}. \tag{20}$$

According to the singular value decomposition,

$$R = U\Sigma V^T. \tag{21}$$

Considering the effect of the calculation error, R may not be an orthogonal matrix. Therefore, R is corrected to

$$\hat{R} = U\Sigma^T. \tag{22}$$

Clearly,

$$R\tilde{X} \leftrightarrow \tilde{Y}. \tag{23}$$

The above algorithm runs fast in computers, and its step is very simple. Therefore, the algorithm is called a fast algorithm.

However, there may be some errors in the previous registration process. Then, to further register the point cloud, a cost function $J(R, t)$ can be built,

$$J(R, t) = \|R\hat{X} + tI_n - Z\|_F^2, \tag{24}$$

where Z represents the corresponding point of $\hat{X}$ in $\hat{Y}$; $\|\cdot\|_F$ stands for the Frobenius norm; and t is the translation vector. Minimize $J(R, t)$ and the optimal parameters can be obtained. Clearly, the registration Eq. (22) may require multiple calculations. Then

$$\begin{cases} (R^{(k+1)}, t^{(k+1)}) = \arg\min \sum_{i=1}^{n} \|R^{(k)}x_i + t^{(k)} - z_i^{(k)}\|^2, \\ R^{(0)} = \hat{R};\, t^{(0)} = (0 \quad 0 \quad 0)^T, \\ \text{st. } R^T R = RR^T = I, \end{cases} \tag{25}$$

where $z_i^{(k)}$ denotes the ith column vector of $Z^{(k)}$ and k is the number of iterations. In addition, there may be an error in the whitening process, hence R may not be an orthogonal matrix. Then, Eq. (25) can be reduced to

$$\begin{cases} (R^{(k+1)}, t^{(k+1)}) = \arg\min \sum_{i=1}^{n} \|R^{(k)}x_i + t^{(k)} - z_i^{(k)}\|^2, \\ R^{(0)} = \hat{R};\, t^{(0)} = (0 \quad 0 \quad 0)^T, \\ \text{st. } |\lambda(R)| \in (1 - \varepsilon, 1 + \varepsilon), \end{cases} \tag{26}$$

where $\varepsilon = 0.5$ in the experiment. However, it is important to choose which iteration formula to use. In order to quickly calculate the optimal value, we use a parallel computing structure. That is, two iteration formulas are simultaneously calculated for several cores. An optimal convergent solution (R^*, t^*) can be obtained with a differential optimization algorithm. Then

$$Q_Y \Lambda_Y^{\frac{1}{2}} R^* \Lambda_X^{-\frac{1}{2}} Q_X^T X + \left(Q_Y \Lambda_Y^{\frac{1}{2}} t^* + \frac{1}{m}\sum_{k=1}^{m} q_k - \frac{1}{n}\sum_{k=1}^{n} p_k\right) I_n \leftrightarrow Y. \tag{27}$$

Further,

$$\begin{cases} \hat{A} = Q_Y \Lambda_Y^{\frac{1}{2}} R^* \Lambda_X^{-\frac{1}{2}} Q_X^T, \\ \hat{p}_0 = Q_Y \Lambda_Y^{\frac{1}{2}} t^* + \frac{1}{m}\sum_{k=1}^{m} q_k - \frac{1}{n} Q_Y \Lambda_Y^{\frac{1}{2}} R^* \Lambda_X^{-\frac{1}{2}} Q_X^T \sum_{k=1}^{n} p_k. \end{cases} \tag{28}$$

For convenience, we call the algorithm proposed in this Letter as the whitening iteration closest point (Whitening-ICP) algorithm. The algorithm flow is as follows.

The time complexity of Step 1 in the Whitening-ICP algorithm is $O(1)$, the time complexity of Step 2 is $O(n)$, Step 3 cost $O(1)$ complexity, and the major cost is Step 4 that searches the nearest point in two point sets by a Delaunay irregular network; its complexity is close to $O(n \log n)$, so the time and space complexities of our

```
Whitening-ICP Algorithm
Input: Target point cloud $X$ and source point cloud $Y$;
Step 1: Remove the mean of the point clouds $X$ and $Y$ using Eq. (6);
Step 2: Whiten $X$ and $Y$ using Eqs. (7) and (11);
Step 3: Estimate the value of $R$ using Eq. (20);
Step 4: Parallel compute the following two steps:
Step 4.1: Optimize Eq. (25) until $\sum_{i=1}^{n}\|R^{(k)}x_i + t^{(k)} - z_i^{(k)}\|^2 < \sum_{i=1}^{n}\|R^{(k+1)}x_i + t^{(k+1)} - z_i^{(k+1)}\|^2$;
Step 4.2: Optimize Eq. (26) until $\sum_{i=1}^{n}\|R^{(k)}x_i + t^{(k)} - z_i^{(k)}\|^2 < \sum_{i=1}^{n}\|R^{(k+1)}x_i + t^{(k+1)} - z_i^{(k+1)}\|^2$;
Step 5: Using Eq. (28), estimate the parameters and register the point clouds.
```

method can be simply written as $O(n\log n)$ and $O(n)$, respectively.

To prove the effectiveness of the algorithm proposed in this Letter, Bunny point cloud data provided by Stanford University and Face point cloud data provided by CPD were used for registration verification. The simulation was based on MATLAB 2016a, whose environment was configured for a 2.5 GHz CPU with 8 GB RAM.

In this section we consider the case where the point cloud is disturbed by white noise. The registration result of the point cloud shown in Fig. 1 is when the signal-to-noise ratio (SNR) is 20 dB.

It can be seen from Fig. 1 that the source point cloud is very unclear because it is affine transformed and disturbed by noise. When both point clouds are whitened, the outline of the source point cloud can be seen roughly.

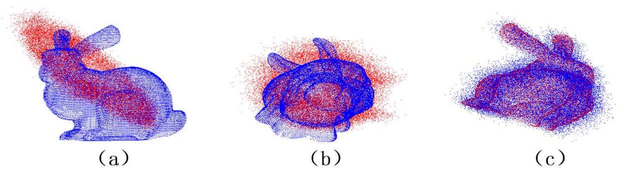

Fig. 1. Schematic diagram of the registration process: (a) initial state, (b) point clouds after being whitened, (c) point cloud after registration.

This fully proves that the whitening process is very effective. In order to verify the stability of the algorithm, experiments were carried out under different SNRs. The experimental results are shown in Table 1 and Fig. 2 in which the root mean square error (RMSE) representation is

$$\mathrm{RMSE} = \frac{1}{n}\|\hat{A}X + \hat{p}_0 I_n - Z\|_F, \tag{29}$$

where Z is the corresponding point of the point cloud $AX + p_0 I_n$ in the point cloud Y.

Taking the Bunny point cloud (approximately **31,600** points) as an example, we test the accuracy and the elapsed time of several algorithms: CPD[18], Affine-ICP[15], MDAR[13], and the algorithm proposed in this Letter. In order to prove the effectiveness of the algorithm in this Letter, we calculated the average running time and registration error on 100 trials in Table 1. In addition, to make a quantitative comparison of different methods, we analyzed the standard deviation of the errors for all the trials in Fig. 2. It can be seen from Table 1 and Fig. 2 that the Whitening-ICP algorithm has a strong anti-noise performance.

In order to better verify the effect of an arbitrary affine registration, a Face point cloud (392 points) was added for an affine registration test. At the same time, we added a state-of-the-art classic algorithm PR-GLS[23]

Table 1. Registration Effect Under Different Signal-to-Noise Ratios

Metric	SNR	CPD	Affine-ICP	MDAR	Whitening-ICP
Ave. Error (mm)	20	0.6834	1.8235	0.9731	**0.2641**
	25	0.6882	1.9111	0.9208	**0.3684**
	30	0.1825	2.3022	0.6375	**0.0788**
	35	0.6237	2.0444	0.8572	**0.0612**
Ave. Time (s)	20	1219	195.2	43.6	**40.3**
	25	1410	170.9	28.2	**52.8**
	30	1484	344.6	24.9	**19.4**
	35	1330	70.3	41.5	**13.4**

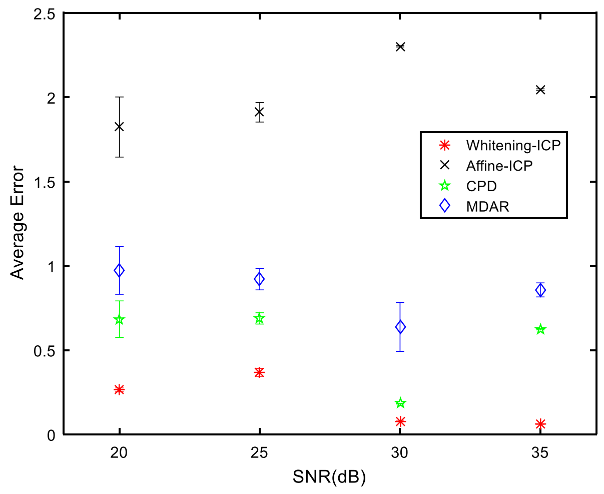

Fig. 2. Comparison of Whitening-ICP with Affine-ICP, CPD, and MDAR on different SNR for the Bunny point cloud. Error bars: registration error means and standard deviations over 100 trials.

for comparison. Since PR-GLS makes MATLAB exceed the memory limit when performing large-scale point cloud registration, here we only use the Face point cloud for testing. It can be seen from Figs. 3 and 4 that CPD and MDAR distort the point cloud, Affine-ICP cannot register the point cloud under any affine change at all, and the registration effect of PR-GLS is not ideal. By contrast, Whitening-ICP can achieve the best performance. In Tables 1 and 2, obviously, the time consumption of Whitening-ICP is relatively optimal. In Table 2, although the error in CPD is the smallest, it consumes dozens of times the time of Whitening-ICP. So, it can be concluded that Whitening-ICP has the best performance.

In this Letter, we use the portable laser scanner (HANDYSCAN 700TM portable laser scanner) to collect data from objects. Considering the existence of the ground truth affine transformation, we first label the object and then the object can be automatically located by the scanner in the scanning process so that the interference

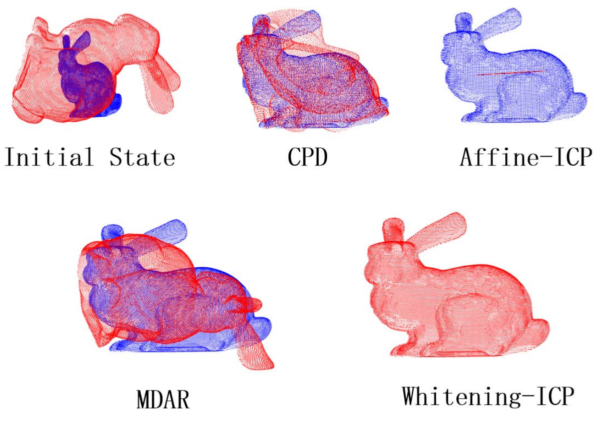

Fig. 3. Bunny point cloud registration effect of several algorithms under the condition that the point cloud is affine transformed.

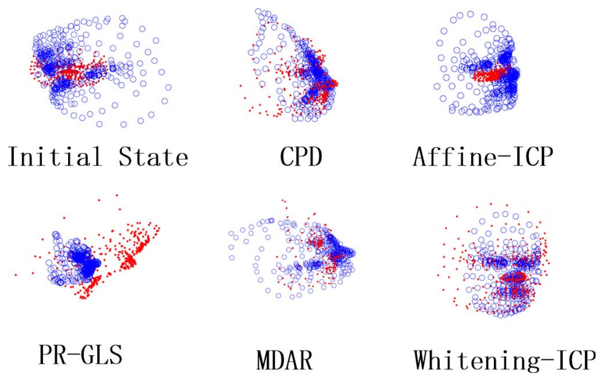

Fig. 4. Face point cloud registration effect of several algorithms under the condition that the point cloud is affine transformed.

of the ground and the background can be ignored. The data collected by the scanner was exported to the software MeshLab and the point clouds were reconstructed, which is shown in Fig. 5.

Figures 5(a) and 5(b) show the scanned objects and scanned point cloud data (Bottle: 21,400 points, Cup: 34,000 points, Banana: 16,800 points). As can be seen from Fig. 5(b), there are holes in the surface of some point clouds. The collected point cloud data is not complete due to the reflection of the object surface. Therefore, registering point clouds during the experiment is more challenging than in simulation. In order to fully demonstrate the superiority of the algorithm, this Letter uses CPD[18], Affine-ICP[15], and MDAR[13] to compare with the algorithm. Their registration results are shown in Fig. 6.

Figure 6 shows that the other methods have shown better performance except for Affine-ICP because the three objects did not undergo obvious affine deformation during actual scanning. However, it can be seen from Table 3 that Whitening-ICP has better stability and accuracy and higher efficiency.

In simulations and experiments, the Affine-ICP algorithm often has poor registration results due to the limitations of the convergence domain. For CPD, in fact, we can also adjust the parameters of the CPD algorithm so that it may have a better registration effect. However, the CPD depends on the initial state of the point clouds and inevitably distorts the point cloud. MDAR is just a multi-directional affine algorithm; it allows the point cloud to expand and contract in the three directions of X, Y, and Z. Therefore, its effect is not stable during arbitrary affine registration, which depends on the degree of affine deformation. However, Whitening-ICP shows excellent point clould registration results under arbitrary affine transformation.

In conclusion, we proposed a novel approach named Whitening-ICP for point cloud registration under arbitrary affine transformation. Our approach uses a simple whitening operation to find the relationship between two point clouds, and the global vector features of point clouds are introduced and combined with the least squares

Table 2. Registration Results on 100 Trials for Face Point Cloud

Metric	CPD	Affine-ICP	PR-GLS	MDAR	Whitening-ICP
Ave. Error (mm)	**0.0911**	0.4151	0.3803	0.2796	**0.1314**
Ave. Time (s)	**3.5**	0.1	35.1	0.6	**0.1**

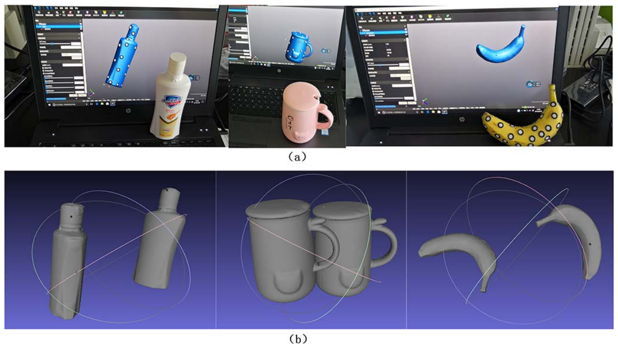

Fig. 5. Point cloud and objects.

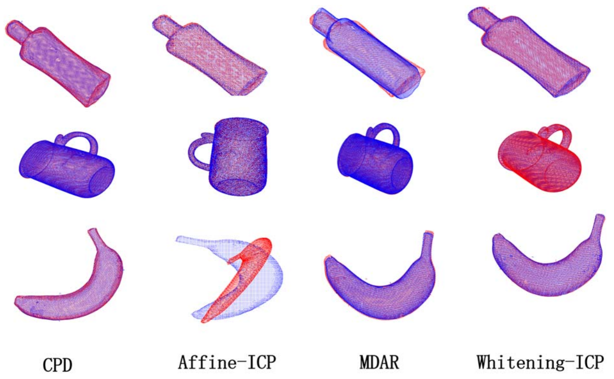

Fig. 6. Registration results of several algorithms.

method to achieve fast coarse registration. The proposed algorithm based on parallel computing can work on low-performance computers. We validated the effectiveness of our method both in simulations and experiments, and it realizes a more advanced performance compared with other state-of-the-art methods. However, Whitening-ICP can register complete point clouds. When the point clouds are incomplete, Whitening-ICP may fail to register them. We will improve this algorithm in future work and try to solve this problem.

This work was supported by the Science and Technology Supporting Program Project of Sichuan Province (Nos. 2018GZ0226 and 2017YFB0405101).

References

1. L. Li, Z. Pan, H. Cui, J. Liu, S. Yang, L. Liu, Y. Tian, and W. Wang, Chin. Opt. Lett. **17**, 061001 (2019).
2. C. T. Seo, S. W. Kang, and M. Cho, Chin. Opt. Lett. **15**, 081102 (2018).
3. P. J. Besl and N. D. McKay, IEEE Trans. Pattern Anal. **14**, 239 (1992).
4. X. Ge, ISPRS J. Photogramm. **130**, 344 (2017).
5. M. Bueno, H. Gonzalez-Jorge, J. Martinez-Sanchez, and H. Lorenzo, Automat. Constr. **81**, 134 (2017).
6. S. Ji, Y. Ren, J. Zhao, X. Liu, and H. Gao, Optik **140**, 451 (2017).
7. X. Huang, J. Zhang, L. Fan, Q. Wu, and C. Yuan, IEEE Trans. Image Process. **26**, 3261 (2017).
8. S. Byun, K. Jung, S. Im, and M. Chang, Int. J. Precis. Eng. Man. **18**, 1221 (2017).
9. Y. He, B. Liang, J. Yang, S. Li, and J. He, Sensors. **17**, 1862 (2017).
10. S. Ying, J. Peng, S. Du, and H. Qiao, IEEE Trans. Autom. Sci. Eng. **6**, 559 (2009).
11. A. Makovetskii, S. Voronin, V. Kober, and D. Tihonkih, Procedia Eng. **201**, 322 (2017).
12. J. Ho, A. Peter, A. Rangarajan, and M.-H. Yang, in *IEEE 12th International Conference on Computer Vision* (2009), p. 1335.
13. C. Wang, Q. Shu, Y. Yang, and F. Yuan, IEEE Photonics J. **10**, 6901215 (2018).
14. J. Kannala, E. Rahtu, and J. Heikkila, in *IEEE International Conference on Image Processing* (2005), paper III-1064.

Table 3. Registration Results on 50 Trials for Three Actual Objects

Metric	Algorithm	Bottle	Cup	Banana
Ave. Error (mm)	CPD	0.4846	0.3448	0.6374
	Affine-ICP	0.7682	0.3102	18.0051
	MDAR	1.3873	0.8858	**0.4394**
	Whitening-ICP	**0.4036**	**0.2964**	0.4511
Ave. Time (s)	CPD	550	902	263
	Affine-ICP	129	68.4	294
	MDAR	9.8	16.4	**7.09**
	Whitening-ICP	**145.4**	**79.0**	51.4

15. S. Du, N. Zheng, S. Ying, and J. Liu, Pattern Recognit. Lett. **31**, 791 (2010).
16. Z. Wu, H. Chen, and S. Du, in *International Joint Conference on Neural Networks (IJCNN)* (2016), p. 1415.
17. H. Chen, H. Zhang, S. Du, Z. Wu, and N. Zheng, IEEE/CAA J. Automa. Sin. **6**, 981 (2019).
18. A. Myronenko and X. Song, IEEE Trans. Pattern Anal. **32**, 2262 (2010).
19. B. Jian and B. Vemuri, IEEE Trans. Pattern Anal. **33**, 1633 (2011).
20. M. Lu, J. Zhao, Y. Guo, and Y. Ma, IEEE Trans. Geosci. Remote Sens. **13**, 162 (2016).
21. S. Du, J. Liu, C. Zhang, J. Zhu, and K. Li, Neurocomputing **157**, 187 (2015).
22. W. Tao and K. Sun, IEEE Trans. Image Process. **24**, 3754 (2015).
23. J. Ma, J. Zhao, and A. L. Yuille, IEEE Trans. Image Process. **25**, 5867 (2016).
24. S. Zhang, K. Yang, Y. Yang, Y. Luo, and Z. Wei, Pattern Recogn. **80**, 183 (2018).
25. J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, Int. J. Comput. Vision. **127**, 512 (2019).
26. J. Ma, J. Zhao, J. Jiang, H. Zhou, and Q. Z. Sheng, IEEE Trans. Neur. Netw. Learn. Syst. **30**, 3584 (2019).