Vis Comput (2011) 27: 543–553
DOI 10.1007/s00371-011-0580-0

ORIGINAL ARTICLE

Registration of point clouds using sample-sphere and adaptive distance restriction

Yu Meng · Hui Zhang

Published online: 19 April 2011

Abstract Registration of point clouds is a fundamental problem in shape acquisition and shape modeling. In this paper, a novel technique, the sample-sphere method, is proposed to register a pair of point clouds in arbitrary initial positions. This method roughly aligns point clouds by matching pairs of triplets of points, which are approximately congruent under rigid transformation. For a given triplet of points, this method can find all its approximately congruent triplets in $O(kn \log n)$ time, where n is the number of points in the point cloud, and k is a constant depending only on a given tolerance to the rotation error. By employing the techniques of wide bases and largest common point set (LCP), our method is resilient to noise and outliers. Another contribution of this paper is proposing an adaptive distance restriction to improve ICP (iterative closest point) algorithm, which is a classical method to refine rough alignments. With this restriction, the improved ICP is able to reject unreasonable corresponding point pairs during each iteration, so it can precisely align the point clouds which have large non-overlapping regions.

Keywords Point cloud · Pairwise rigid registration · Range image alignment · Iterative closest point · Random sample consensus

Y. Meng (✉) · H. Zhang
School of Software, Tsinghua University, Beijing 100084, P.R. China
e-mail: meng-y08@mails.tsinghua.edu.cn

Y. Meng
e-mail: amngy09@gmail.com

H. Zhang
e-mail: huizhang@mail.tsinghua.edu.cn

Y. Meng · H. Zhang
Key Laboratory for Information System Security, Ministry of Education, Beijing 100084, P.R. China

Y. Meng · H. Zhang
Tsinghua National Laboratory for Information Science and Technology, Beijing 100084, P.R. China

1 Introduction

In recent years, 3D digital scanners and laser range scanners are more and more widely used in shape acquisition and shape modeling. Since one of the most popular techniques to acquire surface is to align multiple scans called point clouds, which are achieved from different view points, into a coherent coordinate system, point cloud registration becomes a fundamental problem. This problem is difficult because of three reasons. First, the point clouds only partially overlap, and which parts are overlapping is unknown. Second, because each point cloud is represented in its own coordinate system, no relationship is available to estimate their initial positions. Third, except a set of discrete points, no other useful information is provided, such as normals, geometrical shapes and so on.

Usually, the whole process of registration can be divided into two steps: rough alignment and refinement. The first step is to put the point clouds from different initial coordinate systems into the same coordinate system, and transform them to their correct positions approximately. For rough alignment, there are generally two classes of solutions. One is known as feature-based method [4, 6, 7, 10, 15, 16, 22, 23]; the other is known as RANSAC (random sample consensus) method or try-and-vote method [1, 9, 14]. The second step is to make the corresponding parts overlap precisely. For refinement, ICP (iterative closest point) algorithm

[5] and its variants [2, 7, 12, 17–21, 24, 26] are dominant techniques and widely used.

The feature-based method generally solves three subproblems: first, calculating a geometric descriptor for each point in the point cloud; second, selecting a subset of points as feature points according to the descriptors [15, 16] or the histogram of descriptor values [4, 6, 10, 22, 23]; third, matching these feature points to generate a transformation that can align point clouds. Many feature-based methods focus on how to solve the first and second subproblems, but the third subproblem is still challenging, since it naturally has a $O(n_p^3 n_q^3)$ complexity, where n_p and n_q are respectively the number of feature points in the two point clouds to be registered. Gelfand et al. [10] proposed a branch-and-bound algorithm with greedy-strategy to match the feature points, although it is much faster, the complexity of the stage of greedy selection, $O(n_p^2)$, is still overly high especially for large point clouds. Another shortcoming of this algorithm is that it cannot guarantee the greedy selection is safe. For all feature-based methods, one common limitation is that they will fail when the point clouds do not have strong features.

RANSAC or try-and-vote methods can be used safely no matter whether point clouds have strong features or not. When there are strong features, RANSAC can make use of the feature information to accelerate, or it can be employed by feature-based methods to solve the third subproblem [6]. However, RANSAC has a disadvantage of time consumption [10]. For example, for a pair of point clouds P and Q, a naive idea of RANSAC is to first pick three points as a base from P randomly, then extract all the triplets of points approximately congruent with the base from Q, and finally select the best one. Given a triplet base, extracting all the approximately congruent triplets costs $O(n_Q^3)$ time. Dror Aiger et al proposed a much faster RANSAC method called 4PCS (4 points congruent sets) [1], which uses 4 approximately coplanar points as a base to reduce the time cost to $O(n_Q^2)$. However, it is still time expensive.

ICP algorithm [5] is an effective technique to register point clouds with high precision. However, the original ICP algorithm has two main flaws. One is that it requires the point clouds have good initial positions (i.e., have been aligned approximately), otherwise it will converge to a local optimum solution. The other flaw is that it may fail to get the correct result, if the point clouds are only partially overlapping. A number of strategies are employed to improve the original ICP. [26] gives a good summary of the improvements appeared before 2001. However, new techniques of improvements are still being proposed continuously, such as comprehensive look-up matrix [2], invariant features [24], collinearity constraint and closeness constraint [17], approximate k-d tree [12] and so on. Actually, the fundamental reason causing the two flaws is that it fails to find the real corresponding point pairs (or fails to exclude the wrong pairs). Generally there are two types of methods to distinguish the wrong corresponding pairs of points from the wrong pairs. The first type is assigning a weight to each corresponding pair to evaluate the reliability that it is a real one [19, 20]. The second type is directly judging a corresponding pair, asking whether it is reasonable or not by some criterion [17, 21]. The former is a challenging task because of the difficulty to estimate the weights, and it cannot guarantee all the weights of wrong pairs are zero when the algorithm ends, which will lead to an inaccurate result. The latter is able to avoid all the false pairs at the end of the algorithm, however, since almost all of the nearest points are not real corresponding ones at the start phase, it tends to bring a local optimal solution.

In this paper, we propose the sample-sphere method to roughly align point clouds, and design an improved ICP algorithm equipped with an adaptive distance restriction to refine the results. These techniques overcome the flaws and limitations of RANSAC frame and ICP algorithm discussed above. The main contributions of this paper are as follows:

- The sample-sphere method reduces the complexity of finding all the approximately congruent point sets of a given base. This technique uses a triplet of points as a base, and can find all the triplets approximately congruent with the base in $O(kn \log n)$ time, where n is the number of points, and k is a constant depending only on a given tolerance to the rotation error.
- The adaptive distance restriction is used to judge whether a pair of corresponding points is reasonable or not in each iteration of ICP algorithm. With this restriction, our algorithm is able to gradually exclude all the unreasonable corresponding pairs of points.

The rest of this paper is organized as follows. In Sect. 2, the model of sample-sphere is presented, and how to use it to find approximately congruent triples is exhibited. In Sect. 3, a novel method to roughly align point clouds with sample-sphere is given. In Sect. 4, the adaptive distance restriction and the ICP algorithm equipped with it are presented. Then a group of experiments and comparisons are shown in Sect. 5, and finally the conclusion is given in Sect. 6.

2 Sample-sphere

2.1 Model of sample-sphere

In order to compute a transformation matrix to align point clouds P and Q in 3D space, at least three couples of corresponding points are needed. Given a triplet of points in P as a base, our goal is to find all the triplets which can be approximately congruent with the base under rigid transformation

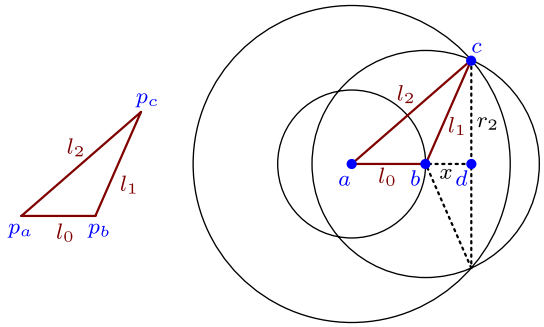

Fig. 1 The mathematical model for developing sample-sphere. $\triangle p_a p_b p_c$ is a base triangle. l_0, l_1 and l_2 are three sides of the triangle

in Q. Given three points p_a, p_b, p_c as a base, the triangle $\triangle p_a p_b p_c$ is called base triangle (see Fig. 1). A naive method to find approximately congruent triplets is as follows: for every point $q \in Q$, first a group of points are extracted from Q to constitute two sets $Q_b = \{q_b | \|q - q_b\| \approx l_0\}$ and $Q_c = \{q_c | \|q - q_c\| \approx l_2\}$, then each combination of $q_b \in Q_b$ and $q_c \in Q_c$ is tested for whether it satisfies $\|q_b - q_c\| \approx l_1$. This method has a complexity of $O(n_Q(n_Q + n_{Q_b} n_{Q_c}))$. It does not sufficiently make use of the geometrical relationships among points, and has a lot of unnecessary computation. For example, to find Q_b and Q_c, it is unnecessary to search the whole point cloud, and there is no need to try all the combinations when checking the distance between q_b and q_c. The technique of sample-sphere is able to avoid these redundant operations.

In Fig. 1, for a point a, all the points, from which the distances to a are equal to l_0, are on the surface of $Sphere(a, l_0)$, where a and l_0 are the center and radius respectively. Similarly, all the points, from which the distances to a are equal to l_2, are on the surface of $Sphere(a, l_2)$. For a point b on $Sphere(a, l_0)$, we can make a $Sphere(b, l_1)$ whose radius is equal to l_1. Assuming that $Sphere(a, l_2)$ and $Sphere(b, l_1)$ intersect on a $Circle(d, r_2)$, any point c on $Circle(d, r_2)$ is able to combine with b and a to form a triangle $\triangle abc$ that is congruent with $\triangle p_a p_b p_c$.

We now discretize the above mathematical model by sampling a group of points on $Sphere(a, l_0)$ and $Circle(d, r_2)$. The foundation of discretization is that in rough alignment it is acceptable for the results to have errors within some tolerances in translation and rotation. Assume that the acceptable translation tolerance and rotation tolerance are S_T and S_R. The usage of S_T will be discussed in Sect. 3. Now we focus on how to discretize the sphere model using S_R. As shown in Fig. 2(a), two congruent surfaces are overlapping perfectly. $\triangle p_a p_b p_c$ and $\triangle q_a q_b q_c$ are two triangles with the same shape in the two surfaces respectively. There is an angle θ between $\triangle p_a p_b p_c$ and $\triangle q_a q_b q_c$. When the two surfaces are roughly registered by aligning $\triangle p_a p_b p_c$ and $\triangle q_a q_b q_c$ (see Fig. 2(b)), the angle between the two surfaces

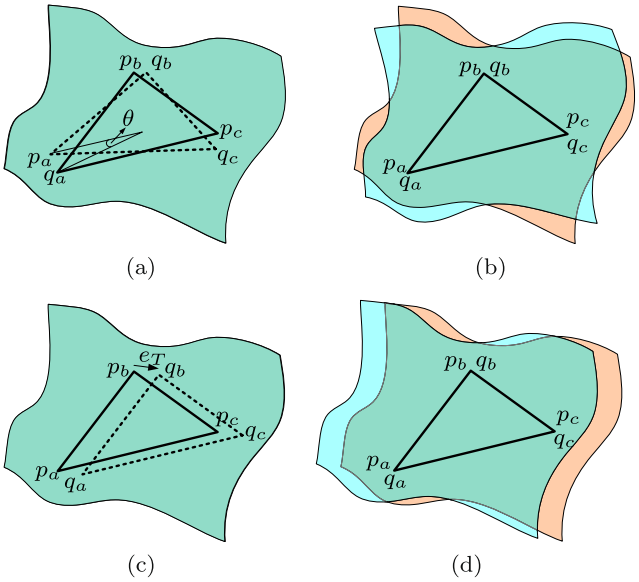

Fig. 2 Rotation error and translation error: (**a**) $\triangle p_a p_b p_c$ and $\triangle q_a q_b q_c$ are two congruent triangles respectively in the two congruent surfaces which are overlapping perfectly. The rotation error between the two triangles is θ. (**b**) The two surfaces are registered by aligning $\triangle q_a q_b q_c$ to $\triangle p_a p_b p_c$. (**c**) $\triangle p_a p_b p_c$ and $\triangle q_a q_b q_c$ are two congruent triangles respectively in the two congruent surfaces which are overlapping perfectly. There is a translation error e_T between the two triangles. (**d**) The two surfaces are registered by aligning $\triangle q_a q_b q_c$ to $\triangle p_a p_b p_c$

is also θ. If θ is not larger than the rotation tolerance S_R, this rough alignment is acceptable. That means the angle between two corresponding sides of the two triangles should be no more than S_R. Because of this, when sampling points on the sphere, we should keep the angle between the two line segments, which connect two neighboring sample points to the center of sphere, within S_R. We call this constraint *Rotation Constraint*.

Sampling points on sphere(a, l_0) For convenience, the sampling process is illustrated in spherical coordinate system (see Fig. 3). Assuming that point a is at the origin point, a point on $Sphere(a, l_0)$ can be denoted as (α, β, l_0), where $\alpha \in [-\pi, \pi)$ and $\beta \in [-\frac{\pi}{2}, \frac{\pi}{2})$. Sampling points on $Sphere(a, l_0)$ is to sample the value of α and β. When α is fixed (see Fig. 3(a)), for a sample point $b_i(\alpha, \beta, l_0)$, the next sample point is $b_{i+1}(\alpha, \beta + \Delta\beta, l_0)$. Because of *Rotation Constraint*, we set

$$\Delta\beta = S_R. \tag{1}$$

When β is fixed (see Fig. 3(b)), for a sample point b_i (α, β, l_0), the next sample point is $b_{i+1}(\alpha + \Delta\alpha, \beta, l_0)$. Because of *Rotation Constraint*, we set $\angle b_i a b_{i+1} \leq S_R$. Line segment e_1 is a chord of $Sphere(a, l_0)$, and according to cosine theorem,

$$e_1^2 \leq e_{1\max}^2 = 2l_0^2(1 - \cos S_R). \tag{2}$$

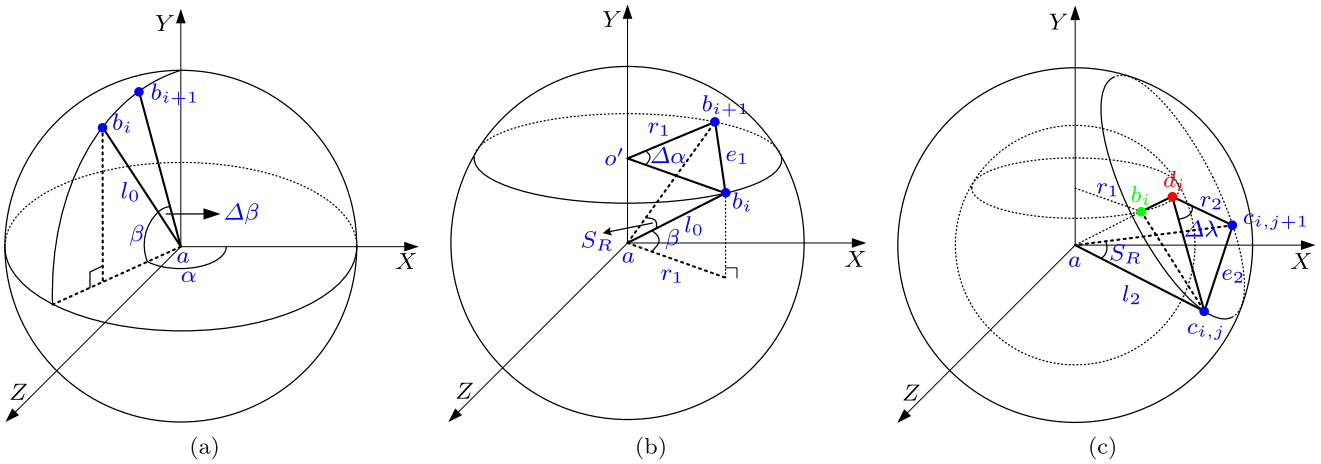

Fig. 3 Sampling points on the sphere: (**a**) how to compute $\Delta\beta$; (**b**) how to compute $\Delta\alpha$; (**c**) how to compute $\Delta\lambda$

e_1 is also a chord of $Circle(o', r_1)$, hence

$$e_1^2 = 2r_1^2(1 - \cos\Delta\alpha). \tag{3}$$

l_0 and r_1 have the relationship

$$r_1 = l_0 \cos\beta. \tag{4}$$

If $2r_1 < e_{1\max}$, the length of any chord of $Circle(o', r_1)$ is shorter than $e_{1\max}$, which means any two points on $Circle(o', r_1)$ satisfy *Rotation Constraint*. In this case, we set $\Delta\alpha = 2\pi$, since only one sample point is enough. If $2r_1 \geq e_{1\max}$, there exists at least one chord $e_1 = e_{1\max}$. Combining (2), (3) and (4), we get

$$\Delta\alpha = \arccos\left(1 - \frac{1 - \cos S_R}{\cos^2\beta}\right). \tag{5}$$

Equation (5) requires $(1 - \frac{1-\cos S_R}{\cos^2\beta}) \in [-1, 1]$, that is,

$$0 \leq 1 - \cos S_R \leq 2\cos^2\beta. \tag{6}$$

The left part of (6) is satisfied automatically. Since $e_1^2 \leq 4r_{1\max}^2$, combining with (3) and (4), we have

$$2l_0^2(1 - \cos S_R) \leq 4l_0^2\cos^2\beta. \tag{7}$$

So the right part of (6) is satisfied, too. To sum up,

$$\Delta\alpha = \begin{cases} 2\pi, & \text{if } 2r_1 < e_{1\max} \\ \arccos(1 - \frac{1-\cos S_R}{\cos^2\beta}), & \text{otherwise} \end{cases}. \tag{8}$$

All the sample points on $Sphere(a, l_0)$ constitute a point set B. Assuming that the number of points in B is n_B, according to (1) and (8), n_B is only determined by S_R. For each point $b_i \in B$, there is a corresponding $Circle(d_i, r_2)$, from which the distances to a and b_i are equal to l_2 and l_1 respectively (see Fig. 3(c)).

Sampling points on circle(d_i, r_2) In Fig. 3(c), on $Circle(d_i, r_2)$, for a sample point $c_{i,j}$, the next sample point is $c_{i,j+1}$. The angle between the two line segments, which connect $c_{i,j}$ and $c_{i,j+k}$ with the center d_i, is $\Delta\lambda$. To sample points on $Circle(d_i, r_2)$, $\Delta\lambda$ needs to be computed. Because of *Rotation Constraint*, we set that angle $\angle c_{i,j}ac_{i,j+1} \leq S_R$. Line segment e_2 is a chord of $Sphere(a, l_2)$, according to cosine theorem,

$$e_2^2 \leq e_{2\max}^2 = 2l_2^2(1 - \cos S_R). \tag{9}$$

e_2 is also a chord of $Circle(d_i, r_2)$, thus

$$e_2^2 = 2r_2^2(1 - \cos\Delta\lambda). \tag{10}$$

If $2r_2 < e_{2\max}$, we set $\Delta\lambda = 2\pi$. The reason is the same as setting $\Delta\alpha = 2\pi$ when $2r_1 < e_{1\max}$. If $2r_2 \geq e_{2\max}$, e_2 is set equal to $e_{2\max}$. Combining (9) with (10), we get

$$\Delta\lambda = \arccos\left[1 - \frac{l_2^2(1 - \cos S_R)}{r_2^2}\right]. \tag{11}$$

Equation (11) requires $[1 - \frac{l_2^2(1-\cos S_R)}{r_2^2}] \in [-1, 1]$, that is,

$$0 \leq l_2^2(1 - \cos S_R) \leq 2r_2^2. \tag{12}$$

The left part of (12) is satisfied automatically. Since $e_2^2 \leq 4r_{2\max}^2$, combining with (9), we have

$$2l_2^2(1 - \cos S_R) \leq 4r_2^2. \tag{13}$$

So the right part of (12) is also satisfied. To sum up,

$$\Delta\lambda = \begin{cases} 2\pi, & \text{if } 2r_2 \leq e_{2\max} \\ \arccos[1 - \frac{l_2^2(1-\cos S_R)}{r_2^2}], & \text{otherwise} \end{cases}. \tag{14}$$

Table 1 The number of points in a sample-sphere

$S_R(^\circ)$	n_B	The maximal m	The maximal k
9	500	40	20501
12	276	30	8557
15	174	24	4341
18	121	20	2542
24	66	15	1057

According to Fig. 1, r_2 can be computed by the equation

$$\begin{aligned} l_1^2 &= x^2 + r_2^2 \\ l_2^2 &= (l_0 + x)^2 + r_2^2 \end{aligned}, \tag{15}$$

whose solution is

$$\begin{aligned} x &= \frac{l_0^2 + l_1^2 - l_2^2}{2l_0}, \\ r_2 &= \sqrt{l_1^2 - (\frac{l_0^2 + l_1^2 - l_2^2}{2l_0})^2}. \end{aligned} \tag{16}$$

When $x < 0$, point d is on the left of point b. When $x > 0$, d is on the right. And when $x = 0$, d is overlapping with b. All the sample points on $Circle(d_i, r_2)$ constitute a point set C_i. Since $\Delta\lambda$ has no relationship with α and β, the number of points in C_i does not depend on the position of b_i or d_i. According to (14) and (16), if l_0, l_1 and l_2 are fixed, r_2 and $\Delta\lambda$ are both constant. Assuming that the number of points in C_i is m, for a certain base triangle $\triangle p_a p_b p_c$, m is a constant. When $\angle p_b p_a p_c = \frac{\pi}{2}$, $r_2 = l_2$ and m gets the maximal value. In this situation, base on (14), $\Delta\lambda = S_R$ and $m = \frac{2\pi}{S_R}$.

Defining sample-sphere The sample-sphere is defined as a model $\langle a, B, C\rangle$, where B consists of all the sample points on $Sphere(a, l_0)$, and C is an union of all the sets C_i corresponding with $b_i \in B$. Any combination of points a, $b_i \in B$, and $c_{i,j} \in C_i$ can constitute a triangle that is congruent with the base triangle. When a base triangle $\triangle p_a p_b p_c$ and a rotation tolerance S_R are given, a sample-sphere can be constructed within $O(k)$ time, where $k = (m+1)n_B + 1$ is the number of points in a sample-sphere. Table 1 enumerates several conditions about the number of points when m gets the maximal value with different S_R. In order to balance the efficiency against the precision, in our experiment, S_R is set equal to 18°. The experiments indicate that, when $S_R = 18^\circ$, the results of rough alignment can provide good initial positions to our ICP algorithm.

2.2 Find approximately congruent triplets

Given a base triangle $\triangle p_a p_b p_c$ and a rotation tolerance S_R, a sample-sphere S^*, whose center is at the origin point, can be computed. For a point $q_a \in Q$, if the center of S^* is moved to q_a, the triangle consisting of q_a, $b_i \in B$ and $c_{i,j} \in C_i$ is congruent with the base $\triangle p_a p_b p_c$. If there exits a point $q_b \in Q$, from which the distance to b_i is shorter than δl_0, and a point $q_c \in Q$, from which the distance to $c_{i,j}$ is shorter than δl_2, points q_a, q_b and q_c constitute an approximately congruent triplet. In our experiment, $\delta = 5\%$.

Algorithm 1

FindCongruentTriplets(q_a, $\triangle p_a p_b p_c$, S^*)

```
1:  T ← φ                                   ▷ Initialize T as an empty set.
2:  for all b_i ∈ B do
3:      b ← b_i + q_a→
4:      q_b ← SearchNearestPoint(b, δl_0)
5:      if such q_b exists, then
6:          for all c_{i,j} ∈ C_i do
7:              c ← c_{i,j} + q_a→
8:              q_c ← SearchNearestPoint(c, δl_2)
9:              if such q_c exists, then
10:                 add Triplet[q_a, q_b, q_c] to T.
11:             end if
12:         end for
13:     end if
14: end for
15: return T.
```

Algorithm 1 shows the main frame of searching such triplets using a sample-sphere. Given a point $q_a \in Q$, a base triangle $\triangle p_a p_b p_c$ and a sample-sphere S^* whose center is at the origin point, the algorithm returns all the approximately congruent triplets at point q_a. In order to move S^* to q_a, all the points in S^* should be translated by a vector $\overrightarrow{q_a}$, which is from the origin point to q_a (line 3). In order to find q_b, the nearest point of b is searched (line 4), and if the distance from b to its nearest point is shorter than a threshold δl_0, the nearest point is treated as q_b. Point q_c is searched in the same way.

The most time-consuming operation of Algorithm 1 is to look for the nearest points within a distance threshold. This operation can be finished in $O(\log n_Q)$ time using a kd-tree structure [3]. In the worst case, every point in B has a nearest point within the distance threshold, and the nearest search repeats $k-1$ times. Therefore, the complexity of the algorithm is $O(k \log n_Q)$ at worst. To find all approximately congruent triplets, the algorithm is executed at every point in Q, so the whole complexity is $O(kn_Q \log n_Q)$ at worst. Since k is a constant for a certain sample-sphere, the complexity can be rewritten as $O(n_Q \log n_Q)$. For example, when $S_R = 18^\circ$, k is always a number no more than 2542 (see Table 1). Additionally, due to the restriction in line 5, many operations of nearest search will be saved, and therefore the algorithm in practice runs faster.

3 Rough alignment

The RANSAC frame is employed to align point clouds P and Q. First, three points are randomly selected from P as a base triplet, and the sample-sphere is computed based on it. Then, for each point in Q, the approximately congruent triplets are extracted as shown in Algorithm 1. After that, the base triplet and each of its approximately congruent triplets are used to compute a rigid transformation M. After Q is transformed by M, the approximately overlapping points between P and Q are counted. Finally, an optimal transformation, under which P and Q have the most approximately overlapping points, is selected.

It is necessary that the three points of a base are all selected from the overlapping regions between P and Q. Otherwise, the approximately congruent triplets will not be the corresponding ones, and therefore the alignment results are wrong. Since the base triplet is selected randomly, it is highly possible that not all of the three points are in the overlapping regions. To get a correct alignment, the operation of RANSAC should repeat L times. L can be either set manually or estimated by [1]:

$$L > \frac{\log 1 - p_s}{\log 1 - p_g^N}, \tag{17}$$

where p_s is the desired probability that the alignment is correct, and p_g is the probability that a point selected randomly is in the overlapping regions. N is the number of points in a base. For a base triplet, $N = 3$.

For efficiency, approximately congruent triplets are not searched at every point in Q. Assuming that p_a and p'_a are two neighbor points in Q, the approximately congruent triplets at p_a and p'_a tend to have nearly the same shapes and positions. Therefore, if the approximately congruent triplets are searched at p_a, it is not necessary to search at p'_a. In order to decide at which point the triplets are searched, two problems need to be considered. The first one is how to control the error which is caused by ignoring other points. As shown in Fig. 2(c), if the two triplets, which are used to align point clouds, have a translation error e_T, the result of alignment also has a translation error equal to e_T (see Fig. 2(d)). If e_T is not larger than the translation tolerance S_T, the alignment is acceptable. Therefore, if at point p_a the triplets are searched, all the points, from which the distances to p_a are shorter than S_T, are ignored. In practice, it is hard to estimate an available S_T visually, but when a distance D between two points in the point cloud is given, it is easy to determine how many times of D the value of S_T should be. Because of this, the relative translation tolerance Δ_{S_T} is employed as an input parameter, and S_T can be computed by $S_T = \rho\Delta_{S_T}$, where ρ is the median of the nearest neighbor distances of all points in the point cloud. The value of ρ is

Algorithm 2

SampleSphereRegistration(P, Q, S_R, Δ_{S_T}, Δ_w, L)

1: $M \leftarrow E$ ▷ Initialize M with identity matrix E.
2: $N \leftarrow 0$
3: $Q_{\text{sub}} \leftarrow$ SelectAvailablePoints(Q, Δ_{S_T})
4: **for** $i \leftarrow 1$ to L **do** ▷ RANSAC Loop
5: $t_B \leftarrow$ SelectBaseTriplet(P, Δ_w)
6: $\triangle p_a p_b p_c \leftarrow$ ConstructBaseTriangle(t_B)
7: $S^* \leftarrow$ ComputeSampleSphere($\triangle p_a p_b p_c$, S_R)
8: **for all** $q_j \in Q_{sub}$ **do**
9: $T \leftarrow$ FindCongruentTriplets(q_j, $\triangle p_a p_b p_c$, S^*)
10: **for all** $t_k \in T$ **do**
11: $M' \leftarrow$ ComputeTransformMatrix(t_B, t_k)
12: $N' \leftarrow$ CountOverlappingPoint(P, Q, M')
13: **if** $N' > N$ **then**
14: $N \leftarrow N'$
15: $M \leftarrow M'$
16: **end if**
17: **end for**
18: **end for**
19: **end for**
20: **return** M

calculated by

$$\rho = \text{Median of } \{D | D = \|q_i - q'_i\|, i \in [1, n_Q]\}, \tag{18}$$

where q'_i is the nearest point of q_i. The second problem is to avoid outliers, since it is meaningless to search approximately congruent triplets at an outlier. Generally, the distance between a random outlier and its nearest point is longer than the distance between a valid point and its nearest point. Thus, if a point's nearest neighbor is further than 2ρ, it is treated as an outlier and ignored.

Algorithm 2 shows the entire procedure of our rough alignment method. The algorithm returns an optimal transformation matrix M, which is used to align Q to P. For efficiency, all the points, at which the approximately congruent triplets need to be searched, are selected to construct a subset Q_{sub} (line 3). When a base triplet is selected (line 5), it should be a *wide base*, with which it is robust to align point clouds [1, 11]. Thus, the distance between any pair of points in a base triplet should be longer than a threshold $\rho\Delta_W$. If a random base is not a wide one, the base is selected again. In line 11, the transformation matrix is computed using the quaternion-based algorithm [13]. Lines 12–16 try to find an optimal transformation matrix, and this process is called LCP (Largest Common Point Set) measure [1]. When the number of overlapping points is counted (line 12), for a point $q \in Q$, if the distance from point $q' = qM'$ to its nearest point in P is shorter than a threshold ε, it is considered as an overlapping point. In our experiment, $\varepsilon = \rho \max(1, \Delta_{S_T})$.

This operation can be accelerated using ANN search structure [3].

4 ICP refinement with adaptive distance restriction

In ICP algorithm, the main problem that affects the precision of registration is how to select corresponding points. The classical ICP always assumes the closest points are corresponding ones. However, this assumption is not always rational. When the point clouds are only partially overlapping, for points in non-overlapping regions, their closest points are not the reasonable corresponding ones needed in ICP. In this paper, an adaptive distance restriction is employed to select reasonable corresponding points.

4.1 Adaptive distance restriction

In our observation, assuming that $p_i \in P$ is the closest point of $q_i \in Q$, if q_i is in the non-overlapping region, the distance between p_i and q_i tends to be longer than that if q_i is in the overlapping region. Thus, when the closest points are treated as corresponding ones, the distance between a pair of corresponding points can be used to distinguish reasonable pairs from unreasonable ones. If the distance is longer than a threshold r, the corresponding pair is judged to be an unreasonable one. In order to decide the value of r, three problems should be considered.

1. If r is small, the convergence rate of iteration is slow. Since all the distances of reasonable corresponding pairs are no more than r, the length of translation vector computed according to these pairs is at most r. Consequently, Q is moved toward P at a speed of each step less than r.
2. A small r may bring the algorithm converging to a local optimal solution. For example, if the current position of Q is the best within the range of d, it is an optimal local solution. However, there may be a better position out of range d. If $r < d$, the translation vector is not able to make Q jump out of d to the better position.
3. Though the two above problems can mostly be solved by assigning r a large value, a larger r involves more unreasonable corresponding point pairs, and finally leads to a poor precision of the registration result.

In different stages of ICP algorithm, different problems should be mainly considered. At the early stage, without the requirement of high precision, the algorithm should have the ability of jumping out of local optimal solutions and moving the point cloud toward a correct direction quickly. At the terminal stage, precision becomes more important, so the algorithm is expected to exclude all the unreasonable corresponding pairs of points.

Based on the above analysis, we design a dynamic threshold r, which changes with the algorithm converging. At the beginning, r is assigned a large value r_B. When the algorithm tends to converge, r is updated by being multiplied a factor $r_s \in (0, 1)$. The change of r breaks the previous convergence, and makes the point cloud move toward a new more precise direction. This updating repeats until r is less than an end value r_e. Here we need a measure to detect the convergence. Two traditional measures of convergence are sum-square distance metric and mean-square distance metric. However, when a threshold r is used to exclude the unreasonable corresponding pairs of points, the number of pairs involved in each iteration is different, and this difference may cause the sum and mean values of square distances to fluctuate a lot during the early stage (see Fig. 4). As a result, it is unreliable to detect convergence by these two measures. According to our observation, the number of reasonable pairs almost always increases monotonously before the convergence entering a relatively stable stage. So we use this number to detect convergence. Assuming that the numbers of reasonable pairs in two successive iterations are c_k and c_{k+1}, in our algorithm, if $|c_{k+1} - c_k| < c_{k+1}c_s$, r is updated, where $c_s \in (0, 1)$ is a given parameter.

Algorithm 3

IcpAlgorithm(P, Q, r_b, r_e, r_s, c_s, τ)

```
M ← E                                   ▷ Initialize M with identity matrix E.
r ← r_b
while r_e < r do                        ▷ The First Stage
    c_0 ← 1
    c_1 ← (c_0 + 1)/(1 − c_s)
    while |c_1 − c_0| > c_0 c_s do
        S ← FindCorrespondingPairs(P, Q, M, r)
        M ← ComputeTransformMatrix(S)
        c_0 ← c_1
        c_1 ← Size of S
    end while
    r ← r r_s
end while
r ← r_e
δ_0 ← 0
δ_1 ← 2τ
while |δ_1 − δ_0| > τ do                ▷ The Second Stage
    S ← FindCorrespondingPairs(P, Q, M, r)
    M ← ComputeTransformMatrix(S)
    δ_1 ← δ_0
    δ_0 ← ComputeErrorSummation(S, M)
end while
return M
```

4.2 Improved ICP algorithm

We divide our ICP algorithm into two stages. In the first stage, the distance restriction r is updated when a convergence is detected. In the second stage, a constant $r = r_e$

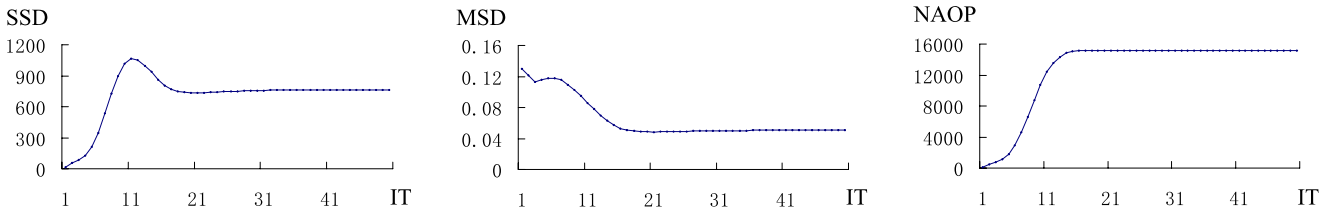

Fig. 4 Measure of convergence. This figure shows the variations of sum-square distance (SSD), mean-square distance (MSD) and the number of approximately overlapping points (NAOP) with iteration times (IT). When a constant distance restriction is employed to erase the unreasonable corresponding pairs, SSD and MSD fluctuate before the algorithm really converging, but NAOP always keeps increasing until the algorithm goes into a relatively stable stage

is used, and the iteration repeats until the change of sum-square error falls below a threshold $\tau > 0$. Algorithm 3 shows the frame of our improved ICP. To make the parameter setting easily, r_b and r_e are relative values of ρ (see (18)). While searching corresponding pairs (line 7 and 19), two alternative strategies, point-to-point and point-to-plane [26], are available. No matter which strategy is used, the corresponding pairs are checked by the distance restriction. In detail, for a point $q \in Q$, if its closest $p \in P$ satisfying $\|p - qM\| \leq \rho r$, point p and q are extracted as a reasonable corresponding pair. In line 8 and 20, a quaternion-based algorithm [5, 13] is employed to compute the transformation matrix. Since the value of sum-square distances (line 23) is more sensitive than the number of reasonable pairs, it can reflect the trend of convergence more precisely during the final stage. Therefore, the variation of sum-square error is used as the terminal criterion. In our experiments, the parameters are set as follows: $r_b = 60$, $r_e = \frac{\sqrt{2}}{2}$, $r_s = 0.5$, $c_s = 0.001$, $\tau = 0.001$.

5 Results

In this section, some experiments and comparisons are shown to illustrate the performance of our algorithms in the respects of robustness, efficiency and registration quality. We implemented the algorithms using C++, and all experiments and comparisons were performed on a PC with 2.53 GHz Intel(R) Core(TM)2 Duo CPU and 2 GB RAM. Figure 5 shows some results of sample-sphere rough alignment and our improved ICP algorithm.

We compared the efficiency of sample-sphere method with 4PCS [1], which is a fast method also employing RANSAC scheme. Table 2 reports the results. When performing sample-sphere methods, we set $S_R = 18°$ and $\Delta_w = 3$. Because RANSAC scheme is a random one, the time of each execution for the same model is different. The time shown in Table 2 is the average time of several (about 10) tests. The results indicate that for large point clouds, the sample-sphere method is faster than the 4PCS method.

Table 2 Comparing sample-sphere with 4PCS. T_{4PCS} and T_{SS} stand for the time (second) cost by 4CPS and sample-sphere method, respectively

Model	n_P	n_Q	Δ_{S_T}	T_{4PCS}	T_{SS}
Coati	28241	28107	14	1.5	2.23
Lady	78722	67571	15	11	5.9
Girl	74672	72282	15	5.5	4.3
Fist	100428	97672	17.5	21	7.1
Teeth	28150	22516	8	2.5	2.8
Face	54681	59943	9.25	199	21.2
Pig	104753	94906	17.25	78	14.6

Figure 6 illustrates the robustness of sample-sphere method and 4PCS method under outliers, and the corresponding time cost is displayed in Fig. 7. The results indicate that when the input data have outliers, sample-sphere method is much more robust than 4PCS method.

Figure 8 illustrates the robustness of sample-sphere method and 4PCS method under noise, and the corresponding time cost is displayed in Fig. 7. It can be seen that time fluctuation of the sample-sphere method is less than that of 4PCS, which indicates that the sample-sphere method is at least as robust as 4PCS when the input data have noise.

We implemented two versions of our improved ICP algorithm: point-to-point version and point-to-plane version. We compared them with the similar functions in two software, PolyWorks 9.0 and Geomagic Studio 10. In our comparison, SIM (Surface Interpenetration Measure) [25] was employed to estimate the quality of refinement. Visually, SIM reflects how large the "splotchy" section a result has, and a better registration should present a larger "splotchy" surface [8] (see Fig. 9). Table 3 reports the SIM value of each refinement. It shows that the point-to-plane version of our improved ICP algorithm is better than PolyWorks and Geomagic. Though the point-to-point version may sometimes perform better than the point-to-plane version, the latter is more stable than the former.

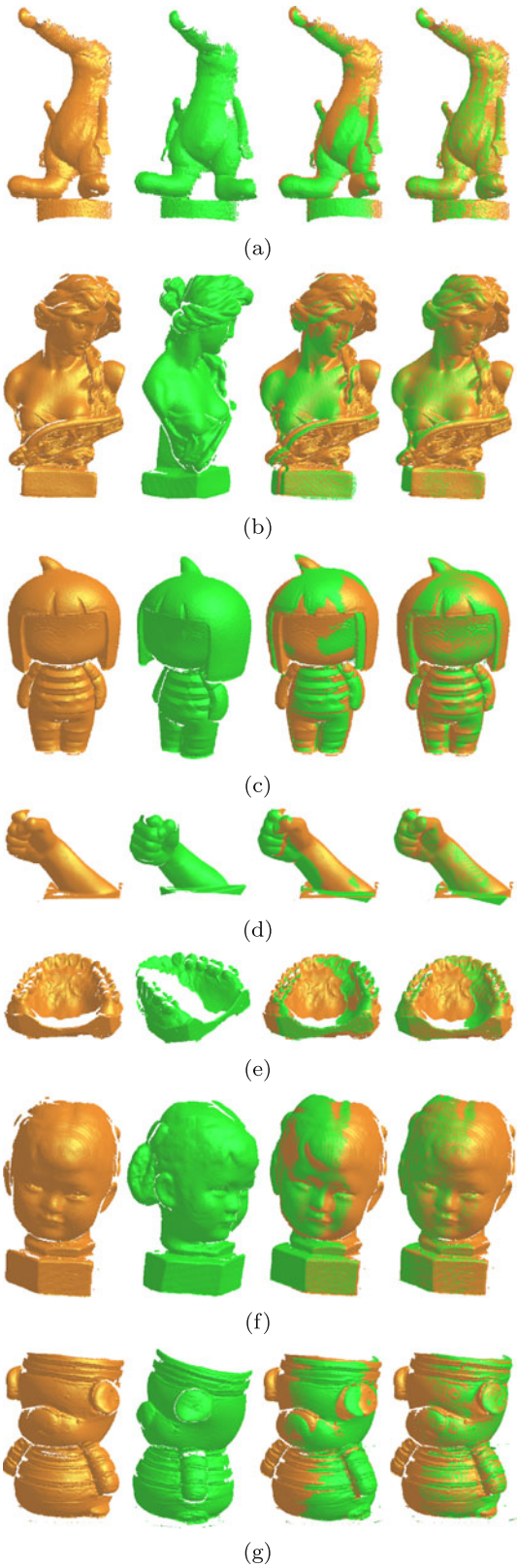

Fig. 5 Results of registration. From top to bottom, the models are named "Coati", "Lady", "Girl", "Fist", "Teeth", "Face" and "Piggy". The rough alignment and refinement are shown in the third and fourth column, respectively

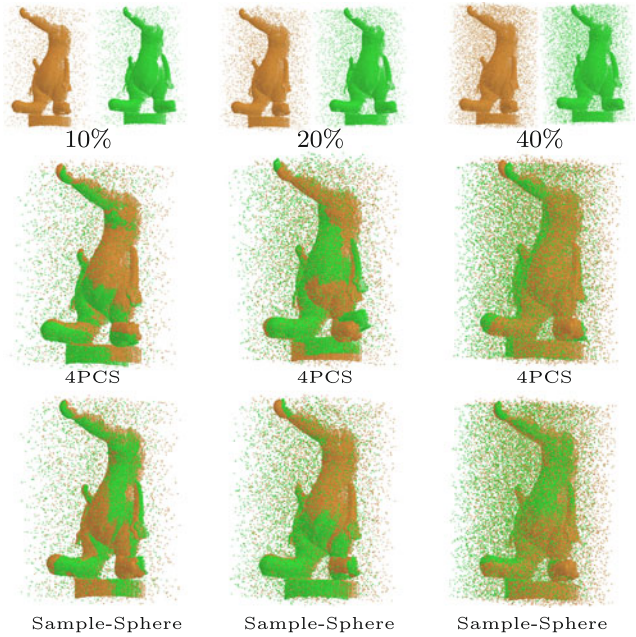

Fig. 6 Rough alignment with outliers. We add random outliers in the bounding boxes of the input data. From left to right, the number of outliers are 10%, 20%, 40% of the original numbers of points. The input point clouds are shown in the *first row*, and the results of the 4PCS and sample-sphere methods are in the *second* and *third row*, respectively

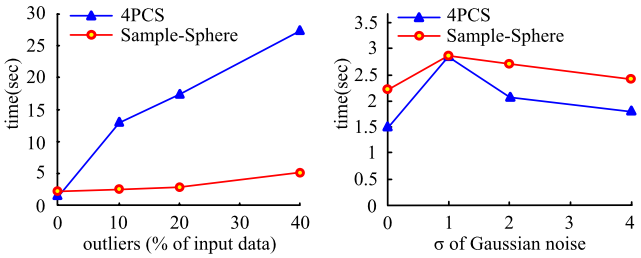

Fig. 7 Comparing the resilience to noise and outliers. The corresponding results of alignments are shown in Fig. 6 and Fig. 8 respectively

Table 3 The SIM values of refinements

Method	Girl	Lady	Fist
PolyWorks	0.504450	0.487478	0.608041
Geomagic	0.512935	0.490059	0.605184
OurICP(point)	0.524704	0.488528	0.610284
OurICP(plane)	0.535121	0.490264	0.609734

6 Conclusion

In this paper, we present a sample-sphere method that roughly aligns two point clouds. This method reduces the time complexity of finding all the approximately congruent triplets of a base to $O(n \log n)$, and it is resilient to noise and outliers. We also propose the adaptive distance restriction to improve ICP algorithm. With this restriction,

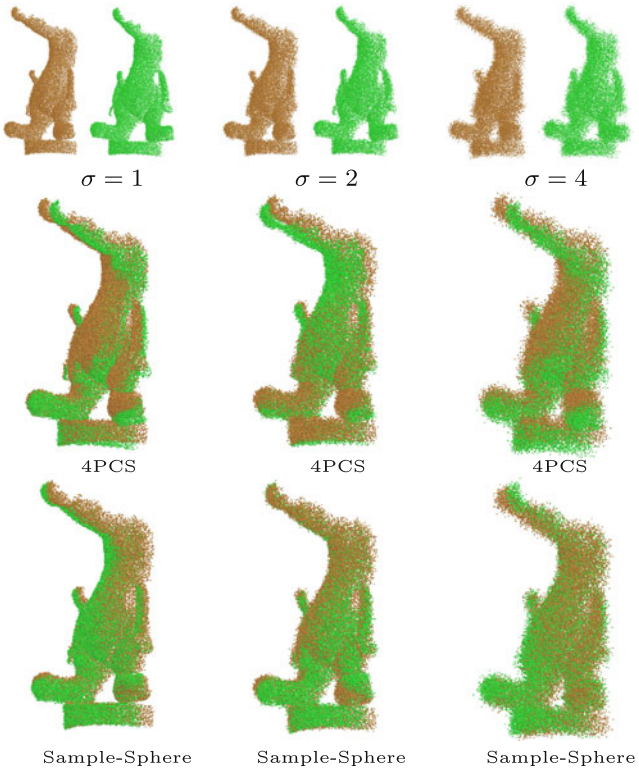

Fig. 8 Rough alignment with Gaussian noise. We disturb the input data with a Gaussian noise, which has a zero-mean. From left to right, the standard deviations σ are 1, 2 and 4. The input point clouds are shown in the first row, and the results of 4PCS and sample-sphere methods are in the *second* and *third row*, respectively

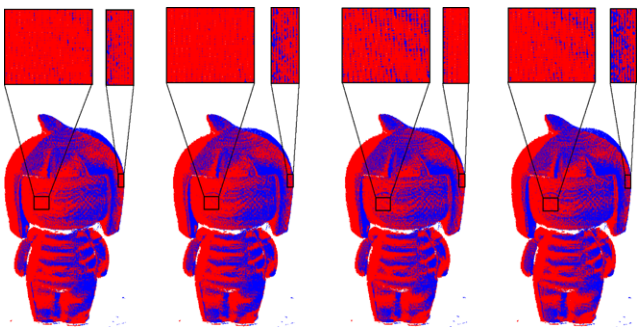

Fig. 9 SIM measure. From left to right, they are the results of PolyWorks, Geomagic, the point-to-point version and point-to-plane version of our ICP Algorithm. The result with higher SIM value (see Table 3) should have larger "splotchy" section

the algorithm gradually erases almost all the unreasonable corresponding pairs of points, thus it can precisely register point clouds only partially overlapping. Combining the sample-sphere method and our improved ICP algorithm, the whole pipeline is able to automatically align two coarse point clouds in arbitrary initial positions accurately.

Acknowledgement We thank the anonymous reviewers for many insightful suggestions that greatly helped improving the paper. This work was supported by the 973 Program of China (Grant No. 2010CB328001) and the National Nature Science Foundation of China (Grant No. 60903106 and No. 61035002).

References

1. Aiger, D., Mitra, N.J., Cohen-Or, D.: 4-points congruent sets for robust surface registration. ACM Trans. Graph. **27**(3), 1–10 (2008)
2. Almhdie, A., Léger, C., Deriche, M., Lédée, R.: 3d registration using a new implementation of the icp algorithm based on a comprehensive lookup matrix: Application to medical imaging. Pattern Recognit. Lett. **28**(12), 1523–1533 (2007)
3. Arya, S., David, M.M., Nathan, S.N., Silverman, R., Angela, Y.W.: An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. J. ACM **45**(6), 891–923 (1998)
4. Basdogan Cagatay, O., Cengiz, A.: A new feature-based method for robust and efficient rigid-body registration of overlapping point clouds. Vis. Comput. **24**(7–9), 679–688 (2008)
5. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992)
6. Bogdan, R.R., Nico, B., Michael, B.: Fast point feature histograms (fpfh) for 3d registration. In: ICRA'09: Proceedings of the 2009 IEEE International Conference on Robotics and Automation, pp. 1848–1853. IEEE Press, Piscataway (2009)
7. Brown, B.J., Szymon, R.: Global non-rigid alignment of 3-d scans. ACM Trans. Graph. **26**(3), 21 (2007)
8. Dalley, G., Flynn, P.: Range image registration: A software platform and empirical evaluation. In: Proceedings of the 3th International Conference on 3-D Digital Imaging and Modeling, pp. 246–253. IEEE Comput. Soc., Los Alamitos (2001)
9. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
10. Gelfand, N., Niloy, J.M., Leonidas, J.G., Pottmann, H.: Robust global registration. In: SGP'05: Proceedings of the third Eurographics Symposium on Geometry Processing, pp. 197–206 (2005)
11. Goodrich, M.T., Mitchell, J.S.B., Orletsky, M.W.: Practical methods for approximate geometric pattern matching under rigid motions: (preliminary version). In: Proceedings of the Tenth Annual Symposium on Computational Geometry, SCG '94, pp. 103–112. ACM, New York (1994)
12. Greenspan, M., Yurick, M.: Approximate k-d tree search for efficient icp. In: International Conference on 3D Digital Imaging and Modeling, p. 442 (2003)
13. Horn, B.K.P.: Closed-form solution of absolute orientation using unit quaternions. J. Opt. Soc. Am. A **4**(4), 629–642 (1987)
14. Irani, S., Raghavan, P.: Combinatorial and experimental results for randomized point matching algorithms. In: Proceedings of the Twelfth Annual Symposium on Computational Geometry, SCG '96, pp. 68–77. ACM, New York (1996)
15. Li, N., Cheng, P., Sutton, M.A., McNeill, S.R.: Three-dimensional point cloud registration by matching surface features with relaxation labeling method. Exp. Mech. **45**(1), 71–82 (2005)
16. Li, X., Igor, G., Jacob, B.: Robust alignment of multi-view range data to cad model. In: SMI '06: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006, p. 17. IEEE Comput. Soc., Washington (2006)

17. Liu, Y.: Improving icp with easy implementation for free-form surface matching. Pattern Recognit. **37**(2), 211–226 (2004)
18. Liu, Y.: Automatic registration of overlapping 3d point clouds using closest points. Image Vis. Comput. **24**(7), 762–781 (2006)
19. Liu, Y.: A mean field annealing approach to accurate free form shape matching. Pattern Recognit. **40**(9), 2418–2436 (2007)
20. Liu, Y.: Replicator dynamics in the iterative process for accurate range image matching. Int. J. Comput. Vis. **83**(1), 30–56 (2009)
21. Liu, Y., Wei, B., Li, L., Zhou, H.: Projecting registration error for accurate registration of overlapping range images. Robot. Auton. Syst. **54**(6), 428–441 (2006)
22. Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M.: Aligning point cloud views using persistent feature histograms. In: IROS, pp. 3384–3391 (2008)
23. Rusu, R.B., Marton, Z.C., Blodow, N., Beetz, M.: Persistent point feature histograms for 3d point clouds. In: Proceedings of the 10th International Conference on Intelligent Autonomous Systems (2008)
24. Sharp, G.C., Lee, S.W., Wehe, D.K.: Icp registration using invariant features. IEEE Trans. Pattern Anal. Mach. Intell. **24**(1), 90–102 (2002)
25. Silva, L., Bellon, O., Boyer, K.: Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5), 762–776 (2005)
26. Szymon, R., Marc, L.: Efficient variants of the icp algorithm. In: Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling, pp. 145–152 (2001)

Yu Meng is a master candidate in School of Software, Tsinghua University, China. He received his B.E. in Electrical Engineering and Automation from Tongji University, in 2008. His research interests include computer-aided design and computer graphics.

Hui Zhang is an Associate Professor in School of Software, Tsinghua University, China. She received her B.Sc. and Ph.D. in Computer Science from Tsinghua University, in 1997 and 2003, respectively. Her research interests include computer-aided design and computer graphics.