

A hybrid point cloud alignment method combining particle swarm optimization and iterative closest point method

Quan Yu · Kesheng Wang

Received: 14 January 2014 / Accepted: 27 January 2014 / Published online: 2 March 2014
© Shanghai University and Springer-Verlag Berlin Heidelberg 2014

Abstract 3D quality inspection is widely applied in many industrial fields including mould design, automotive and blade manufacturing, etc. A commonly used method is to obtain the point cloud of the inspected object and make a comparison between the point cloud and the corresponding CAD model or template. Thus, it is important to align the point cloud with the template first and foremost. Moreover, for the purpose of automatization of quality inspection, this alignment process is expected to be completed without manual interference. In this paper, we propose to combine the particle swarm optimization (PSO) with iterative closest point (ICP) algorithm to achieve the automated point cloud alignment. The combination of the two algorithms can achieve a balance between the alignment speed and accuracy, and avoid the local optimal caused by bad initial position of the point cloud.

Keywords Quality inspection · Point cloud alignment · Particle swarm optimization (PSO) · Iterative closest point (ICP)

1 Introduction

Product quality inspection is of importance for both final and semi products in industrial production. 3D measurement techniques [1] are increasingly applied in various quality control fields, especially for free-form products. A general approach for 3D product quality inspection is employed by obtaining the point cloud of the inspected product and

comparing it with the ideal product template, i.e., a CAD model is used to check the difference between the product and the design. The point cloud can be obtained by using coordinate measurement machine or other non-contact optical techniques, such as structured light system [2, 3].

Before the comparison based quality inspection can be performed, the point cloud must be aligned with the template at first. Otherwise the difference will not be found accurately. Researchers have developed many alignment algorithms in recent years, which can be divided into different classes: iterative closest point (ICP) [4–6] and robust point matching (RPM) [7, 8]. The ICP algorithm is proposed by Besl and McKay [4] on the basis of quaternions and singular value decomposition (SVD), which effectively solves the point alignment problem. Many improvements have been developed by researchers on the basis of the original algorithm to achieve the acceleration or improve the accuracy. Researchers studied the effects of point cloud resampling [9–11], point matching mechanism [12, 13] and the point pairs weighting [14], etc.

In this paper, we propose an algorithm by combining the particle swarm optimization (PSO) with ICP algorithm to achieve the automated alignment. PSO is a type of heuristic algorithm with good adaptivity. It has been widely applied in sorts of research fields. The main objective of using PSO is to explore the global optimized solution in a complex solution space. Unlike other optimisation algorithms, PSO almost has not limitation about the features of the solution space, which means that it can be applied in a black-box problem.

Experiment has been done to test the applicability of PSO-ICP algorithm. Results show that the combination of the two algorithms can achieve more accurate alignment, and the local optimal trap caused by bad initial position of the point cloud can be avoided.

Q. Yu (✉) · K. Wang
Department of Production and Quality Engineering, Norwegian
University of Science and Technology, Trondheim, Norway
e-mail: quan.yu@ntnu.no

The paper is organized as follows. Section 1 introduces the importance of point cloud alignment in quality inspection problem, the related ICP algorithm and its improvements, and the algorithm proposed in this paper. Section 2 introduces the mechanism of ICP algorithm. Section 3 introduces the principle of PSO algorithm. Section 4 describes the combination of PSO with ICP algorithm in detail. Section 5 shows the evaluated experiment and makes a comparison between PSO-ICP and ICP algorithm. Section 6 comes to the conclusion of the proposed PSO-ICP algorithm.

2 ICP algorithm

The ICP algorithm is proposed for the accurate and computationally efficient alignment of 3D shapes [4]. The algorithm converges monotonically to the nearest local minimum of a mean square distance metric. The rate of convergence is quite fast during the first a few iterations. Given the point set P and the model template X , the ICP algorithm accomplishes the alignment as follows:

- (i) Compute the closest points Y on X for each point of P .
- (ii) Compute the alignment vector \mathbf{q} from P to Y .
- (iii) Apply the transformation to P according to the alignment vector \mathbf{q} .
- (iv) Terminate the iteration if the termination rules are met.

ICP algorithm has a rather rapid convergence during the alignment. However, it depends much on the initial relative position of the point set P and the model template X . A bad initial position always leads to a local optimal alignment.

2.1 Spatial rotation on the basis of quaternions

Spatial rotation can be represented in different mathematical ways such as the quaternions and combination of the Euler angles. Quaternion based representation is preferred in ICP algorithm over the SVD approach [4].

The quaternion is a four vector $\mathbf{q}_R = [q_0 \ q_1 \ q_2 \ q_3]^T$, where $q_0 \geq 0$, and $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. The corresponding 3×3 rotation matrix can be computed to calculate the transformation of the point cloud, which is used to pre-multiply column vectors, as shown follows:

$$\mathbf{R} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \tag{1}$$

The translation vector in ICP is defined by $\mathbf{q}_T = [q_4 \ q_5 \ q_6]^T$. Thus, the complete alignment vector \mathbf{q} is written as $\mathbf{q} = [\mathbf{q}_R \ \mathbf{q}_T]^T$. Suppose $P = \{P_i\}$ is a point cloud to be aligned with a model $X = \{x_i\}$, where N_X and N_P are respectively numbers of the point in X and P , and $N_X = N_P$ which means that the index of each point p_i and x_i is corresponding. The object is to minimize the mean square objective function shown as follows:

$$f(\mathbf{q}) = \frac{1}{N_P} \sum_{i=1}^{N_P} \|x_i - \mathbf{R}(\mathbf{q}_R)p_i - \mathbf{q}_T\|^2. \tag{2}$$

The cross-covariance matrix Σ_{PX} of P and X is computed by

$$\Sigma_{PX} = \frac{1}{N_P} \sum_{i=1}^{N_P} [(p_i - \mu_P)(x_i - \mu_X)^T], \tag{3}$$

where μ_P and μ_X are the mass centers of P and X given by

$$\mu_P = \frac{1}{N_P} \sum_{i=1}^{N_P} p_i, \tag{4}$$

$$\mu_X = \frac{1}{N_X} \sum_{i=1}^{N_X} x_i. \tag{5}$$

The cyclic components of the anti-symmetric matrix $A_{ij} = (\Sigma_{PX} - \Sigma_{PX}^T)_{ij}$ are used to get the column vector $\mathbf{A} = [A_{23} \ A_{31} \ A_{12}]$, which is used to form the symmetric 4×4 matrix

$$\mathbf{Q}(\Sigma_{PX}) = \begin{bmatrix} \text{tr}(\Sigma_{PX}) & & & \\ & \mathbf{A} & & \\ & & \Sigma_{PX} + \Sigma_{PX}^T - \text{tr}(\Sigma_{PX})\mathbf{I}_3 & \\ & & & \end{bmatrix}, \tag{6}$$

where \mathbf{I}_3 is the 3×3 identity matrix. The quaternion vector $\mathbf{q}_R = [q_0 \ q_1 \ q_2 \ q_3]^T$ is corresponding to the maximum eigenvalue of the matrix $\mathbf{Q}(\Sigma_{PX})$. The translation vector can be computed by

$$\mathbf{q}_T = \mu_X - \mathbf{R}(\mathbf{q}_R)\mu_P. \tag{7}$$

The mean square error (MSE) d denotes the matching error between P and X , which is computed by Eq. (2). The termination criterion comprises mainly two considerations. One is the MSE threshold and the other one is the difference between the current MSE and last MSE, which is defined as

$$d = |f(\mathbf{q}_t) - f(\mathbf{q}_{t-1})|, \tag{8}$$

where q_t and q_{t-1} are respectively the alignment vectors in iteration t and $t-1$.

3 PSO

PSO is proposed by Kennedy and Eberhart [15] for the purpose of the best solution searching. PSO is inspired by the movement of organisms in a bird flock, which simulates birds' social behaviour. Gradient method is not used in PSO, which means that PSO is feasible for the indifferentiable optimization problems that are partially irregular, noisy even change over time.

3.1 Mechanism of PSO

PSO algorithm has a population of particles, and each particle denotes a candidate solution. The particles move around in the solution space of the optimisation problem, and they are evaluated by the fitness values computed with the fitness function of the problem model. The best known position of each particle and the entire swarm are stored for guiding the movements of the particles, which are defined as the personal best value and the global best value. During the iteration, the movements of the particles are decided by their current positions, the best known individual position and the best known global position. The iteration is repeated until a satisfactory solution is finally discovered.

A particle comprises 3 vectors as shown below:

- (i) The x -vector records the current position of the particle in the search space;
- (ii) The p -vector records the best known location so far by the particle, which is named as personal best;
- (iii) The v -vector contains a velocity for particle, which denotes the moving direction of each particle in the next iteration.

At iteration t , the updating of the velocity and position for the particles follow the equations shown as below:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 (p_i - x_i(t)) + c_2 r_2 (p_g - x_i(t)), \quad (9)$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (10)$$

where v the velocity of the particle, x the current position of the particle, which means the candidate solution, ω the inertia weighting, i the i th particle, c_1 and c_2 the acceleration coefficients of positive constraint, r_1 and r_2 random numbers distribution on $[0,1]$, t the t th iteration.

The pseudo code of PSO can be written, as shown in Fig. 1.

The flowchart of PSO algorithm is shown in Fig. 2.

The given value ω of inertial weight and the acceleration coefficient c affect the solution searching performance

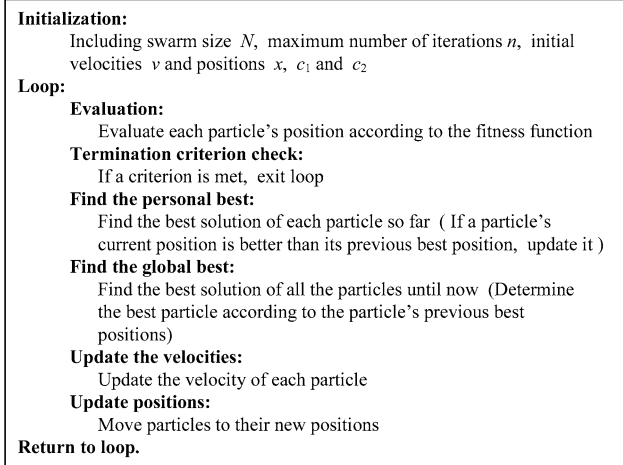


Fig. 1 Pseudo code of PSO

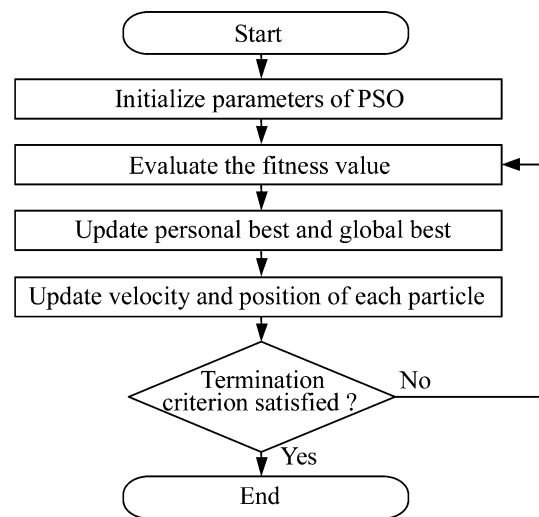


Fig. 2 Flowchart of PSO

of PSO. A larger inertia weight ω determines a global search while a small inertia weight determines a local search. Moreover, the algorithm will have greater local search ability when $c_1 > c_2$, but greater global search ability when $c_1 < c_2$.

4 Combination of PSO with ICP

In point cloud alignment problem, ICP algorithm is able to obtain a good alignment result with small time consumption. Manual parameter adjustment is not required, which makes ICP quite applicable in the automated point cloud alignment. However, the alignment result of ICP algorithm depends much on the initial relative position of the point cloud and model template. And it is difficult to find a better

solution as soon as the convergence starts towards the local optimum. To overcome this, we combine the PSO with ICP to enhance the global optimal searching ability of the original ICP algorithm. PSO is applied to provide various positions of the point cloud as candidates to ICP algorithm. The MSE value of each particle is used as the fitness value for PSO to control the update of the particles. The com-

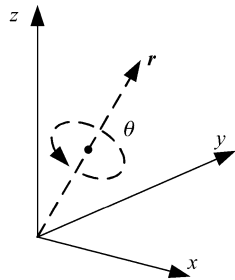


Fig. 3 Axis-angle representation

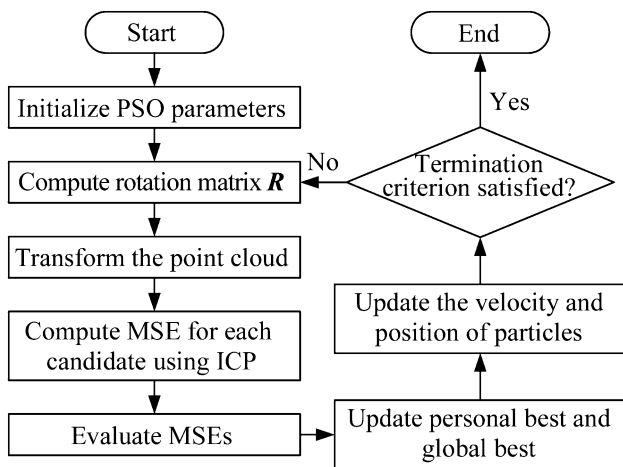


Fig. 4 Flowchart of PSO-ICP algorithm

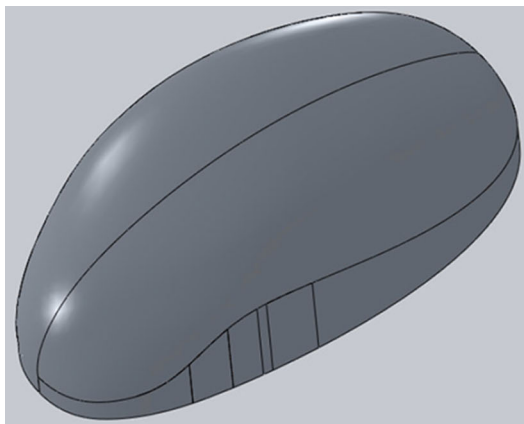


Fig. 5 CAD model and CNC model

bination makes the ICP more robust for the automated point cloud alignment.

4.1 Conversion between axis-angle representation and quaternions

In the proposed PSO-ICP algorithm, axis-angle representation is used for the point cloud rotation transformation. This method uses axis-angle representation to define the rotation of a rigid body in a three-dimensional space. The axis and angle are defined respectively as follows: a unit vector $r = [x, y, z]$ indicates the direction of an axis of rotation, and an angle θ describes the magnitude of the rotation about the axis, as shown in Fig. 3.

Because the ICP algorithm uses quaternions to compute the rotation transformation, the axis-angle representation should be converted to quaternions firstly according to

$$q_R = [\cos(\theta/2) \quad x\sin(\theta/2) \quad y\sin(\theta/2) \quad z\sin(\theta/2)]^T, \tag{11}$$

and then converted to 3×3 rotation matrix according to Eq. (1).

4.2 Proposed PSO-ICP algorithm

On the basis of the axis-angle representation, the solution space is defined as a 4-dimensional searching space including 3 Cartesian coordinates and 1 rotation angle. The PSO-ICP algorithm comprises 4 main steps, shown as follows:

- (i) Randomly generate multiple axis-angle variables and transform the point cloud with corresponding rotation matrix;
- (ii) Perform the ICP algorithm for each point cloud candidate and get the corresponding MSE value as the fitness value;

- (iii) Perform the PSO algorithm to obtain the personal best value and global best value on the basis of MSE values. Update the axis-angle variables following the equations;
- (iv) Finish the algorithm if the termination criterion is met, otherwise continue with the algorithm.

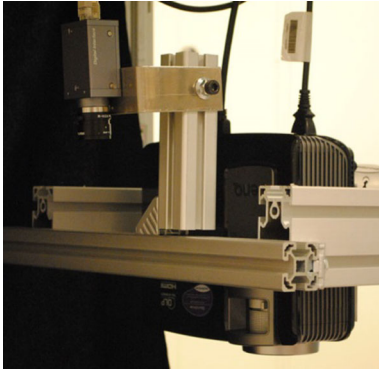


Fig. 6 Structured light system

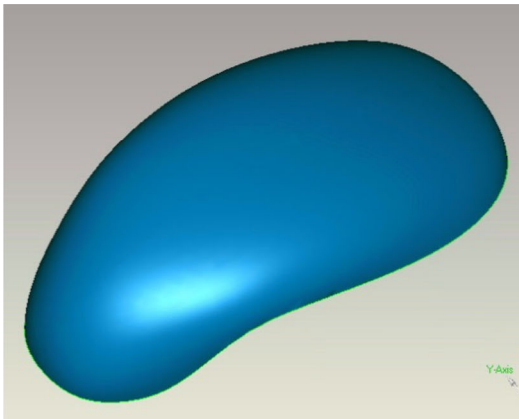


Fig. 7 CAD free-form surface and obtained point cloud

The detailed flowchart of PSO-ICP algorithm is shown in Fig. 4.

5 Experimental results

To test the feasibility of the proposed PSO-ICP algorithm, we produce a CNC model according to the CAD model of a free-form mouse, as shown in Fig. 5.

Then the point cloud of the CNC model is obtained by using a structure light system (SLS), as shown in Fig. 6. The test program is written and run in MATLAB. Regarding that only the top free-form surface is obtained in the point cloud, the vertical and bottom surfaces of the CAD model are removed for the observation of the alignment process, as shown in Fig. 7.

In the experiment, the CAD model is converted to STL format to be imported into MATLAB, which stores points, as shown in Fig. 8.

The original obtained point cloud contains 10^5 points. For the purpose of the test, the original point cloud is

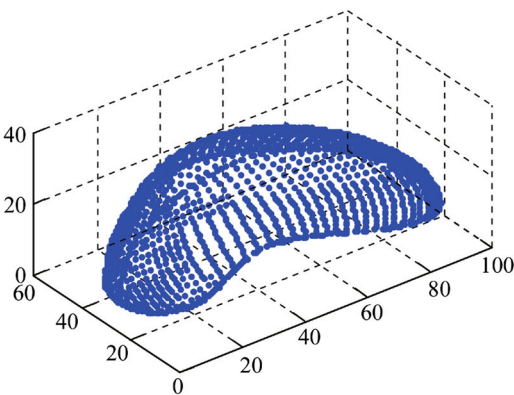
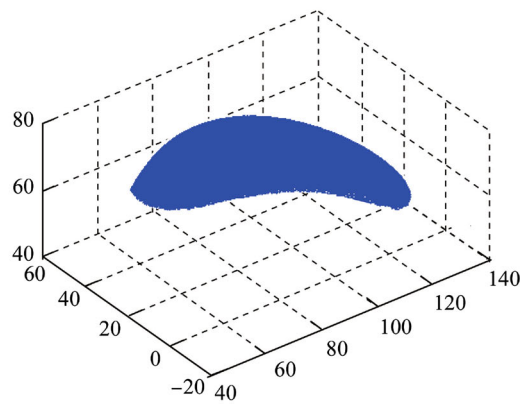
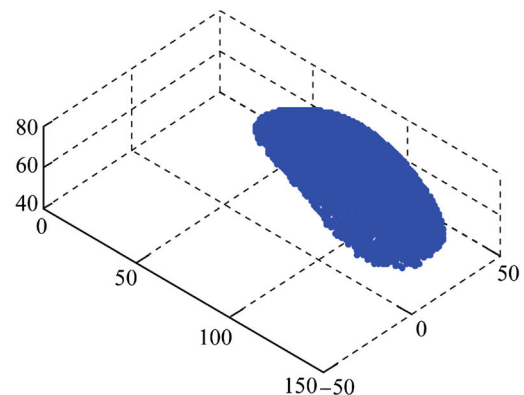


Fig. 8 STL formatted CAD model and obtained point cloud



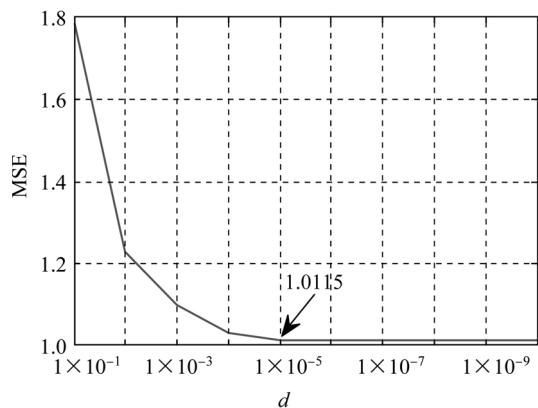


Fig. 9 Obtained MSE with respect to different d

resampled and only 0.02 % points have been kept. The CAD model contains 2,237 points and the point cloud contains 2,170 points. At first, the alignment is tested by using ICP only. The CAD model is marked in blue and the point cloud is marked in green. During the test, the point cloud is randomly given to different initial positions to test

the alignment result. To decide the termination criterions, different d is set to compute the corresponding MSE of the final alignment. The result shows the MSE will not decrease when $d < 1 \times 10^{-5}$, as shown in Fig. 9.

The termination parameters are set to be $MSE < 1$ or $d < 1 \times 10^{-5}$. As shown in Figs. 10 and 11, different initial positions lead to different convergence results.

The experiment shows that as soon as the ICP goes to the local optimal solution, it is impossible to find the best solution. The same initial position is used to test the PSO-ICP algorithm. In the proposed algorithm, we use 3 particles to search the solutions and the initial position is the same with the example shown in Fig. 11. The result shows that PSO-ICP algorithm finds a global optimum better than using ICP, as shown in Fig. 12, which means a better alignment position.

Without loss of generality, we randomly assign the initial position of the point cloud, and test the alignment result of PSO-ICP algorithm. The test result shows that even if the initial position of the point cloud is random, the algorithm can still obtain better optimums. Among 200

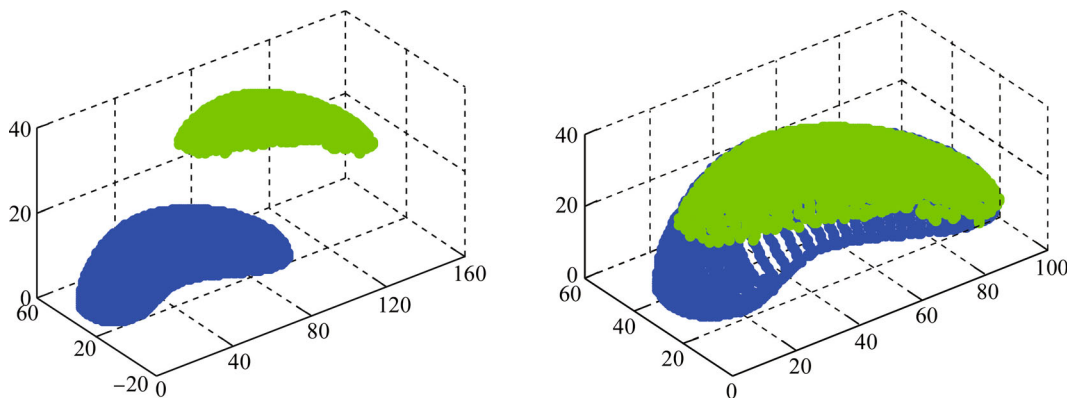


Fig. 10 Registrations using ICP algorithm at a good initial position

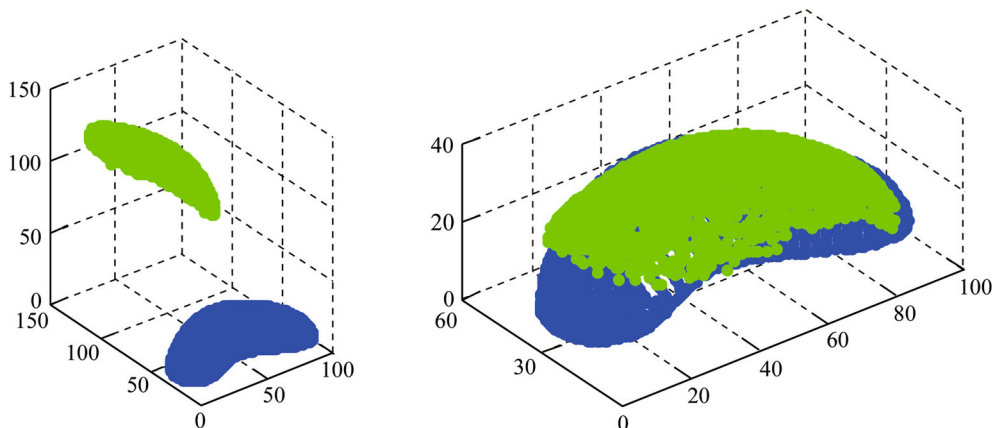


Fig. 11 Alignment using ICP algorithm at a bad initial position

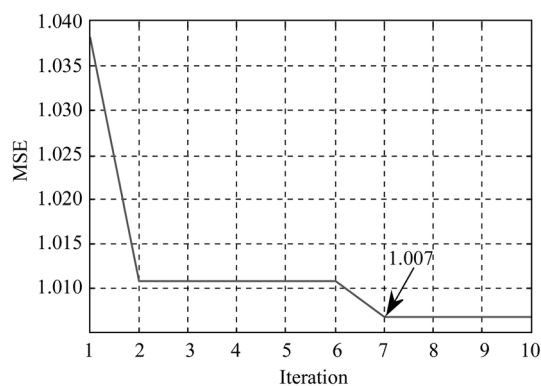


Fig. 12 MSE obtained with PSO-ICP algorithm

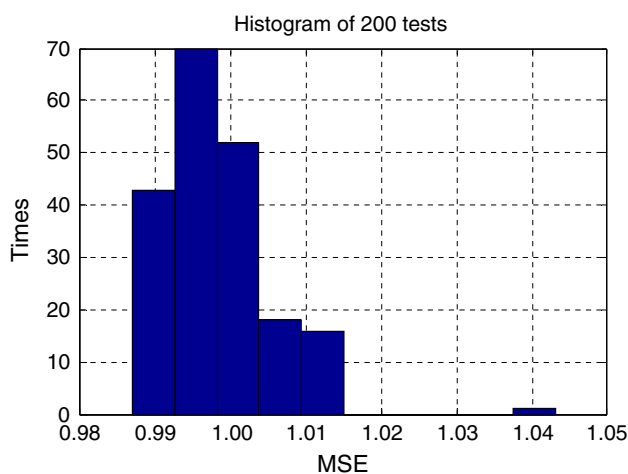


Fig. 13 Result of 200 tests with random initial position

tests, the ideal alignment position is found in 199 tests, as shown in Fig. 13.

6 Conclusions

The point cloud alignment is important in 3D quality inspection. The proposed PSO-ICP algorithm combines the advantages of fast convergence of ICP and global optimum exploring ability of PSO. This combination overcomes the drawback of ICP algorithm that strongly relies on the initial relative position of the point cloud and CAD model. The

experiment result shows that PSO-ICP is more robust and self-adaptive. Thus, the PSO-ICP is applicable to achieve an automated point cloud alignment without artificial interference, and obtain more accurate alignment for further quality inspection. The algorithm is also applicable for point cloud based 3D object recognition.

References

1. Granero L, Sánchez J, Micó V, Esteve JJ, Hervás J, Simón S, Pérez E (2007) 3D digitising using structured illumination. Application to mould redesign, vol 6616, Part 2 edn
2. Barbero BR, Ureta ES (2011) Comparative study of different digitization techniques and their accuracy. *Comput Aided Des* 43(2):188–206
3. Rocchini C, Cignoni P, Montani C, Pingi P, Scopigno R (2001) A low cost 3D scanner based on structured light. *Eurographics* 20:299–308
4. Besl PJ, McKay HD (1992) A method for alignment of 3-D shapes. *IEEE Trans Pattern Anal Mach Intell* 14(2):239–256
5. Chen Y, Medioni G (1991) Object modeling by alignment of multiple range images. In: *Proceedings of IEEE international conference on robotics and automation*, vol 2723, pp 2724–2729
6. Zhang Z (1994) Iterative point matching for alignment of free-form curves and surfaces. *Int J Comput Vis* 13(2):119–152
7. Delibasis K, Asvestas PA, Matsopoulos GK (2010) Multimodal genetic algorithms-based algorithm for automatic point correspondence. *Pattern Recognit* 43(12):4011–4027
8. Gold S, Rangarajan A, Lu C-P, Pappu S, Mjolsness E (1998) New algorithms for 2D and 3D point matching: pose estimation and correspondence. *Pattern Recognit* 31(8):1019–1031
9. Gruen A, Akca D (2005) Least squares 3D surface and curve matching. *ISPRS J Photogramm Remote Sens* 59(3):151–174
10. Masuda T, Sakaue K, Yokoya N (1996) Alignment and integration of multiple range images for 3-D model construction. In: *Proceedings of the 13th international conference on pattern recognition*, vol 1, 871 pp 879–883
11. Turk G, Levoy M (1994) Zippered polygon meshes from range images. In: *Proceedings of the 21st annual conference on computer graphics and interactive techniques (ACM)*, pp 311–318
12. Pulli K (1999) Multiview alignment for large data sets. In: *Proceedings of second international conference on 3-D digital imaging and modeling*, pp 160–168
13. Simon DA (1996) Fast and accurate shape-based alignment. Carnegie Mellon University, Pittsburgh
14. Godin G, Rioux M, Baribeau R (1994) Three-dimensional alignment using range and intensity information, vol 2350 pp 279–290
15. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of IEEE international conference on neural networks*, vol 4, pp 1942–1948