Postprint of Adaptive Evolutionary Point Cloud Registration Algorithm Based on Color Information

Authors: Zou Li, Baozhen Ge, Chen Lei

Date: 2018-05-20T00:00:00+00:00

Abstract

Addressing the limitation that existing evolutionary point cloud registration algorithms do not utilize point cloud color information, this paper proposes an adaptive evolutionary point cloud registration algorithm based on color information. The algorithm samples the input point cloud using a combination of random sampling and color feature points, establishes an objective function by minimizing the median of point-pair distances that incorporate color constraints, and employs an adaptive evolutionary algorithm to solve for the optimal spatial transformation between two point clouds, thereby achieving effective registration. Registration experiments on four colored point clouds demonstrate that, compared with the adaptive evolutionary point cloud registration algorithm using only spatial information and two other relatively recent evolutionary registration algorithms, the proposed algorithm can effectively reduce registration time while maintaining equivalent registration accuracy.

Full Text

Preamble

Point Cloud Registration Based on Self-Adaptive Evolutionary Optimization and Color Information

Zou Li[1] , , Ge Baozhen[1] , , Chen Lei[1] , ,[2]†
(1. Tianjin University a. School of Precision Instruments & Opto-Electronics Engineering; b. Key Laboratory of Opto-Electronics Information & Technical Science of Ministry of Education, Tianjin 300072, China; 2. School of Information Engineering, Tianjin University of Commerce, Tianjin 300134, China)

Abstract: Traditional evolutionary point cloud registration methods often neglect the color information in models. To overcome this limitation, this paper introduces a point cloud registration method based on self-adaptive evolutionary optimization algorithm and color information. The method subsamples input point clouds by extracting color feature points and randomly chosen points, utilizes the median of all pairs of color-constrained points as the objective function, and employs a self-adaptive evolutionary optimization algorithm to obtain the optimal solution. Registration experiments on four colorized point clouds show that, compared with evolutionary point cloud registration methods using only spatial information and two state-of-the-art registration methods, the proposed method significantly shortens processing time while achieving similar registration precision.

Keywords: color point cloud registration; self-adaptive evolution optimization algorithm; feature points extraction

0 Introduction

With the development of 3D digitization technology, 3D modeling has become an important component of computer vision. Due to the limited field of view of 3D scanning devices and the complex geometric shapes of objects, complete depth information of an actual object cannot be obtained from a single viewpoint. To acquire a complete data model of the measured object, multiple scans from different perspectives are required, making precise multi-view point cloud registration a critical step in 3D modeling [**?**].

Currently, the most widely used registration algorithm is the Iterative Closest Point (ICP) algorithm proposed by Besl et al. [**?**]. ICP achieves point cloud registration by iteratively minimizing the mean square error of corresponding point distances between two point clouds. While ICP offers fast speed and high precision, its performance is highly sensitive to initial position, noise, and overlap rate. In recent years, some scholars have applied evolutionary algorithms to 3D point cloud registration [**?**]. Evolutionary point cloud registration algorithms directly search for the optimal transformation in the solution space. The global search capability of evolutionary algorithms ensures that the solution is less likely to fall into local optima, overcoming the ICP algorithm' s dependence on initial position and enabling successful registration even for point clouds with significant noise and low overlap rates.

However, existing evolutionary point cloud registration algorithms primarily focus on point clouds without color information, as traditional depth cameras and 3D scanners only provide spatial position data without surface texture information, limiting the realism of 3D point clouds. Recently, researchers have combined 3D object color information with depth data to generate 3D color point clouds [**?**]. Color point clouds provide better visual effects and enhanced realism for scene objects. Moreover, the additional color information creates opportunities for further optimization of evolutionary point cloud registration algorithms.

To address this, we propose a color information-based adaptive evolutionary point cloud registration algorithm. The algorithm first samples the original point cloud using a combination of color feature points and random sampling. It establishes an objective function by minimizing the median of point-to-point distances with color constraints, which limits the range of corresponding points involved in the calculation and effectively reduces computational complexity. Finally, it employs an adaptive evolutionary algorithm to solve for the optimal spatial transformation between two point clouds. Experiments on multiple color point clouds demonstrate that compared with adaptive evolutionary point cloud registration algorithms using only spatial information and two other recent evolutionary registration algorithms, our proposed algorithm effectively reduces registration time while maintaining equivalent registration precision.

1 Point Cloud Registration Background

The goal of point cloud registration is to find the optimal rigid transformation in 3D space that enables one point cloud to align with another adjacent point cloud through transformation. A registration algorithm typically includes the following components:

a) Input Model. Assume two partially overlapping point clouds: a source point cloud P and a target point cloud Q, where $\mathbf{p}_i$ and $\mathbf{q}_j$ are three-dimensional vectors representing spatial coordinates (X, Y, Z). In color point clouds, each point also includes RGB values. Since input models are typically large-scale, direct computation using original point clouds is time-consuming. Therefore, sampling of the source point cloud is necessary to reduce computational time, while the target point cloud remains unchanged to ensure precision [**?**]. Sampling points should uniformly cover the original model as much as possible, with common methods including random sampling, uniform sampling, and feature point sampling.

b) Spatial Transformation $T(\mathbf{F})$. This determines the parameter space of the registration algorithm, primarily including rigid transformation, affine transformation, and nonlinear transformation.

c) Objective Function $F(T)$. Point cloud registration can be formulated as a nonlinear optimization problem where the objective function, a function of spatial transformation T, quantifies the quality of a transformation.

d) Optimization Method. This is used to find approximate optimal solutions in the solution space. ICP is the most commonly used optimization method in registration, but its convergence precision is severely affected by initial position. Due to their strong global convergence and high precision, evolutionary algorithms are increasingly chosen for optimizing objective functions. Point cloud registration algorithms that use evolutionary algorithms to optimize objective functions are called evolutionary point cloud registration algorithms.

The principle of point cloud registration is illustrated in Figure 1 [Figure 1: see original paper].

2 Adaptive Evolutionary Point Cloud Registration Algorithm

The adaptive evolutionary point cloud registration algorithm uses the Self-Adaptive Evolutionary Optimization (SaEvO) [?] method and represents one of the most advanced evolutionary point cloud registration algorithms available.

SaEvO is an evolutionary algorithm specifically designed for point cloud registration, proposed by Santamaria et al. in 2013. SaEvO employs a meta-optimization framework that uses one evolutionary algorithm to optimize the control parameters of another optimization algorithm. SaEvO consists of two steps: (a) using an evolutionary algorithm based on Differential Evolution [?] and Variable Neighborhood Search [?] to optimize the objective function; and (b) using an Artificial Immune Algorithm [?] to adjust the control parameters in step (a). During optimization, steps (a) and (b) execute alternately, enabling optimization of the objective function under optimal control parameters to obtain the best transformation between two point clouds. The SaEvO algorithm framework is shown in Figure 2 [Figure 2: see original paper].

In addition to the optimization method, the adaptive evolutionary point cloud registration algorithm employs rigid transformation, which includes three rotation angles (α, β, γ) and three translation parameters (T_x, T_y, T_z). Sampling points are obtained through random sampling, and the objective function $F(T)$ is established using the median of point-to-point distances. The calculation method is as follows: for each point $\mathbf{p}_i$ in source point cloud P, let $\mathbf{q}_{c(i)}$ be the nearest point in target point cloud Q after transformation T, typically obtained using Kd-tree [?] or GCP [?]. The corresponding point distance d_i is:

$$d_i = \|T(\mathbf{p}_i) - \mathbf{q}_{c(i)}\|^2$$

where $T(\mathbf{p}_i)$ represents the point $\mathbf{p}_i$ after transformation T. The objective function $F(T)$ represents the median of squared distances for all corresponding points:

$$F(T) = \mathrm{MedSE} = \mathrm{median}\{d_i \mid i = 1, \dots, N_p\}$$

The adaptive evolutionary point cloud registration algorithm finally uses SaEvO to optimize and solve for the optimal spatial transformation.

In reference [?], SaEvO was compared with other improved evolutionary algorithms including Santamaria' s Scatter Search algorithm [?], Silva' s improved Genetic Algorithm [?], and Wachowiak' s improved Particle Swarm Optimization algorithm [?]. Results demonstrated that SaEvO exhibits excellent optimization capability for registration, and its self-adaptive characteristics provide high robustness for different point clouds, yielding the most accurate registration results. However, the adaptive evolutionary point cloud registration algorithm does not utilize color information and only addresses traditional point cloud registration, leaving room for further optimization in color point cloud registration.

3 Color-Based Adaptive Evolutionary Registration Algorithm

Unlike traditional 3D point clouds, 3D color point clouds contain not only each point' s positional information (3D spatial coordinates X, Y, Z) but also color information (RGB values). Existing evolutionary point cloud registration algorithms do not use model color information, presenting opportunities for further optimization. Therefore, this paper introduces color information into the point cloud registration algorithm and proposes a color information-based adaptive evolutionary point cloud registration algorithm. The algorithm uses a sampling method combining random sampling and color feature points to ensure uniform distribution of sampling points in both geometric and color spaces. Additionally, it incorporates a color similarity constraint for matching points in the objective function, further reducing algorithmic complexity. Finally, it solves for the optimal spatial transformation through SaEvO.

3.1 Color Point Cloud Model and Feature Point Sampling

The input model in the algorithm is a color point cloud model where each point contains both spatial position information and RGB values. However, for the same object viewed from different perspectives, RGB values may vary significantly due to lighting effects. To effectively utilize color information, lighting effects must be eliminated. An object' s Hue is less affected by lighting [**?**, **?**]. Therefore, in this algorithm, each point' s normalized hue value represents color information (hereinafter referred to as hue value). The formula for converting RGB values to hue value H is:

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 2\pi - \theta & \text{if } B > G \end{cases}$$

where

$$\theta = \frac{1}{2} \arccos \left[\frac{R - G + R - B}{2\sqrt{(R-G)^2 + (R-B)(G-B)}} \right]$$

Thus, source point cloud P and target point cloud Q can be further represented as $\{(\mathbf{p}_i, h_i^P)\}_{i=1}^{N_p}$ and $\{(\mathbf{q}_j, h_j^Q)\}_{j=1}^{N_q}$, where $\mathbf{p}_i$ and $\mathbf{q}_j$ are three-dimensional vectors representing spatial coordinates of source and target point clouds, and h_i^P and h_j^Q represent hue values of the i-th point in P and the j-th point in Q, respectively.

In color point cloud registration, due to non-uniform color distribution in models, random sampling may result in most sampled points having similar or identical hues, reducing algorithm efficiency. Therefore, a sampling method is needed to ensure both spatial uniformity (covering the entire model) and color uniformity. This paper proposes a sampling method combining color feature points and random sampling. First, introduce the method for selecting color feature points. Given a parameter n, divide the hue range from 0 to 1 into n equal portions, then partition P into n sub-point-clouds $P_{d_1}, P_{d_2}, \dots, P_{d_n}$, where each sub-point-cloud P_{d_k} includes only points with hue h_i^P satisfying:

$$\frac{k-1}{n} \le h_i^P < \frac{k}{n}$$

Similarly, divide Q into n sub-point-clouds $Q_{w_1}, Q_{w_2}, \dots, Q_{w_n}$, where each sub-point-cloud Q_{w_k} includes only points with hue h_j^Q satisfying:

$$\frac{k-1.5}{n} < h_j^Q \le \frac{k-0.5}{n}$$

Figure 3 [Figure 3: see original paper] illustrates this point cloud partitioning for the simple case when $n = 3$.

To select color feature points, randomly choose a sub-point-cloud P_{d_k} from all sub-point-clouds of P. If P_{d_k} is not empty and the corresponding sub-point-cloud Q_{w_k} in Q is also not empty, randomly select one point from P_{d_k} as a color feature point. Repeating this process ensures that the hue distribution of sampled points is essentially uniform. Let the required number of color feature points be N_{sample}. The color feature point selection process is shown in Figure 4 [Figure 4: see original paper].

Color feature points alone cannot guarantee spatial uniformity and may cause local fitting issues. To ensure both spatial uniformity and rich color information, the algorithm combines color feature point selection with random sampling: first select a portion of color feature points, then randomly sample additional points. In practice, we set $n = 15$, with a ratio of 2:1 between color feature points and random sampling points being appropriate.

3.2 Color-Constrained Objective Function

This paper proposes a color-constrained objective function $\Psi(T)$. Traditional median distance objective functions evaluate a rotation-translation transformation by finding, for each point $\mathbf{p}_i$ in P, its nearest point $\mathbf{q}_{c(i)}$ in Q as a matching point. Finding matching points consumes the vast majority of computation time. To reduce this computational cost, we propose a color-constrained objective function that assumes matching points should have similar hues. Therefore, the search for nearest points is limited to points in Q with similar hues, eliminating the need to search all points in Q.

The color constraint increases matching point accuracy, as points with large hue differences cannot be matched, resulting in fewer local optima in the function curve and easier optimization. Simultaneously, $\Psi(T)$ reduces the search range for matching points, significantly shortening the time required to find them.

For a point in P belonging to sub-point-cloud P_{d_k}, the nearest point is only searched within the same-index sub-point-cloud Q_{w_k} in Q. The hue range covered by Q_{w_k} is larger than P_{d_k}, ensuring that points in P with hues at the boundaries can still find matches without mismatches. The sub-point-cloud index k for a point with hue h can be obtained using:

$$k = \left\lfloor \frac{h \cdot n}{1} \right\rfloor + 1$$

where $\lfloor\cdot\rfloor$ denotes the floor function. A Kd-tree data structure can be pre-built for each sub-point-cloud to accelerate nearest-point searches. After obtaining matching points, the median of matching point distances is used as the objective function value.

The following lists the operational procedure for $\Psi(T)$ when evaluating spatial transformation T:

```
for i = 1, ..., N_p:
    Step 1) Calculate point p_i after transformation T
    Step 2) Calculate sub-point-cloud index k using equation (7)
    Step 3) Find nearest point q_c(i) in Q_wk
    Step 4) Calculate point pair distance d_i = ||T(p_i) - q_c(i)||^2
    Step 5) Output result Ψ(T) = median{d_i | i = 1, ..., N_p}
```

The time complexity of objective function $\Psi(T)$ is analyzed as follows. Assuming nearest points are obtained through Kd-tree with complexity $O(\log N_q)$, the complexity of traditional median objective function is $O(N_p \log N_q)$. Assuming uniform hue distribution in point clouds, the complexity of our proposed objective function $\Psi(T)$ is $O(N_p \log(N_q/n))$. In practice, point cloud hue distribution is non-uniform; besides the dominant hue, other hues have low proportions, so most color feature points have few corresponding points. Therefore, actual algorithmic complexity will be even lower.

3.3 Population Encoding and Optimization Solution

This paper employs SaEvO to optimize and solve for the optimal transformation between point clouds. The variables to be solved are a set of spatial transformations including three rotation angles and three translation vectors—six independent variables total. Therefore, 6-dimensional real-valued encoding is adopted, representing rotation variables (α, β, γ) and translation variables (T_x, T_y, T_z). The spatial transformation matrix $\mathbf{T}$ can be expressed as:

$$\mathbf{T} = \mathbf{R}_\alpha \mathbf{R}_\beta \mathbf{R}_\gamma \mathbf{S}$$

where

$$\mathbf{R}_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{R}_\beta = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{R}_\gamma = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Using $\Psi(T)$ as the objective function and the sampled source point cloud with the original target point cloud as input, the approximate optimal transformation between point clouds can be obtained after iterating for a certain time.

4 Experimental Results and Analysis

To verify the effectiveness of the proposed algorithm, two groups of registration experiments were conducted. In Experiment 1, our algorithm was compared with the evolutionary point cloud registration algorithm using only spatial information (hereinafter referred to as the spatial algorithm). The specific structures of the two algorithms are shown in Table 1 . In Experiment 2, our algorithm was compared with two other recent evolutionary point cloud registration algorithms from the literature.

4.1 Point Cloud Models

Four pairs of scene point clouds from the RGB-D Object Dataset [?] were selected as experimental models: meeting_small_1_171 and meeting_small_1_176 (abbreviated as meeting1 and meeting2); kitchen_small_1_33 and kitchen_small_1_38 (kitchen1 and kitchen2); desk_3_27 and desk_3_32 (desk1 and desk2); and table_1_67 and table_small_1_72 (table1 and table2). The number of points in each viewpoint point cloud is shown in Table 2 , with the point clouds visualized in Figure 5 [Figure 5: see original paper].

4.2 Experimental Environment

The experimental platform was configured with an Intel Core i7-4790 processor at 3.6 GHz, 8 GB RAM, and Windows 7 Ultimate 64-bit SP1 operating system. All registration algorithms were implemented in C++ and compiled with Microsoft Visual Studio 2010. Nearest neighbor searches were implemented using Kd-tree from the FLANN library.

The stopping condition was set to a runtime of 200 seconds. Since objective functions differ, direct comparison of convergence results is not possible. Therefore, the current optimal transformation was recorded every 10 seconds during each run. For fair evaluation, all results were finally assessed using the traditional median point-to-point distance objective function. To eliminate the influence of different sampling points on final evaluation, unsampled source point clouds and target point clouds were used as input for the final unified evaluation. Convergence curves were obtained from 20 objective function values. To avoid randomness, each algorithm was run independently 30 times, with convergence curves representing the average of 30 runs.

Since optimization algorithms have a small probability of falling into local optima, resulting in objective function values far exceeding those of correct registration and significantly impacting averages, a threshold was set: if the objective function value exceeded 5×10^{-5}, the run was marked as failed. Statistics were collected on successful runs out of 30 attempts, including average convergence curves, average objective function values, and average iteration counts for successful runs.

Each algorithm sampled 200 points from the source point cloud using the methods shown in Table 1 . The initialization range for rotation angles was $[0, 2\pi]$, and for each translation vector dimension was $[-1, 1]$. Parameters for the self-adaptive evolutionary algorithm were: population size 50, $\rho = 0.0625$, $\gamma = 20$, $\beta = 0.8$, $\nu = 0.25$.

The convergence curves for the four point cloud models are shown in Figure 6 [Figure 6: see original paper]. To avoid excessive span in objective function values, only convergence curves between 50-200 seconds are plotted. Table 3 shows the average objective function values, success counts, and average iteration counts for 30 runs of each model.

The convergence curves in Figure 6 [Figure 6: see original paper] demonstrate that our algorithm converges significantly faster than the spatial algorithm, typically achieving good registration precision within 100 seconds, while the spatial algorithm requires approximately 180 seconds to basically converge. This is primarily because the color constraint in our algorithm reduces objective function complexity. Data in Table 3 better illustrate our algorithm' s advantages: while convergence precision is similar to the spatial algorithm, our algorithm only failed once during meeting model registration, whereas the spatial algorithm failed more frequently (5 times for the desk model, with 9 total failures), showing a much higher probability of falling into local optima. This further demonstrates that our objective function has fewer local optima and is easier to optimize. Additionally, our algorithm can achieve approximately 2,500 generations within 200 seconds, while the spatial algorithm only reaches 1,500 generations—just 60% of our algorithm' s iteration count—indicating that our proposed objective function significantly reduces computational load compared to traditional objective functions. Figure 7 [Figure 7: see original paper] shows successful registration results for our algorithm, demonstrating excellent alignment between the two point clouds.

4.4 Registration Experiment 2 Results Analysis

In Experiment 2, our algorithm was compared with two other evolutionary point cloud registration algorithms:

a) Artificial Bee Colony Registration Algorithm (ABC2016) [?]: Proposed in 2016, this method uses ISS feature points for sampling and artificial bee colony algorithm for optimization.

b) Invasive Differential Evolution Registration Algorithm (DE2014) [?]: Proposed in 2014, this adds an invasive model to differential evolution. The original literature [?] includes two strategies; we selected the better-performing AIM-dDE-ChAvg for comparison.

Control parameters for our algorithm were identical to Experiment 1, while parameters for the other two algorithms remained consistent with their original literature, as shown in Table 4 .

Convergence curves for Experiment 2 are shown in Figure 8 [Figure 8: see original paper]. Average objective function values and success counts for successful runs are presented in Table 5 . Since population sizes differ among algorithms, comparing iteration counts is not meaningful, so total iteration counts were not recorded.

Figure 8 [Figure 8: see original paper] shows that our algorithm consistently achieves the fastest convergence speed, averaging 80-90% faster than other algorithms. Table 5 indicates our algorithm also has the highest success rate and better robustness across different models. For example, both ABC2016 and DE2014 have low success rates for the meeting model, while our algorithm registers accurately. In terms of precision, our algorithm achieves the highest accuracy for kitchen and table models, while for other models its precision is very close to the best values. These experiments demonstrate that compared with the latest evolutionary registration algorithms, our algorithm offers faster convergence while remaining highly competitive in success rate and precision.

5 Conclusion

This paper proposes a color information-based adaptive evolutionary point cloud registration algorithm for color point cloud registration. The algorithm first samples the source point cloud using a combination of color feature points and random sampling. When computing the objective function, the original point cloud is divided into multiple sub-point-clouds, limiting the search for corresponding points to a single sub-point-cloud to reduce computation. Finally, the optimal transformation is solved through the self-adaptive evolutionary algorithm.

Compared with the adaptive evolutionary point cloud registration algorithm using only spatial information, our algorithm achieves 1.7 times the iteration count within the same time, offering faster convergence and higher registration success rates. Compared with two other evolutionary registration algorithms, our algorithm is highly competitive in precision, convergence speed, and success rate. These results demonstrate the effectiveness of our proposed algorithm.

References

[1] Besl P J, Mckay N D. A method for registration of 3-D shapes [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 1992, 14 (2): 239-256.
[2] Li T, Gao L, Pan Q, et al. Differential evolution algorithm-based range image registration with scaling parameters [C]// Proc of IEEE International Conference on Image Processing. 2016: 4508-4512.
[3] Bermejo E, Cordón O, Damas S, et al. A comparative study on the application of advanced bacterial foraging models to image registration [J]. Information Sciences, 2015, 295: 160-181.
[4] García-Torres J M, Damas S, Cordón O, et al. A case study of innovative population-based algorithms in 3D modeling: artificial bee colony, biogeography-based optimization, harmony search [J]. Expert Systems with Applications, 2014, 41 (4): 1750-1762.
[5] Domeniconi S, Solerio L. Genetic algorithms for voxel-based medical image registration [C]// Proc of IEEE Fourth International Workshop on Computational Intelligence in Medical Imaging. 2013: 22-29.
[6] Falco I D, Cioppa A D, Maisto D, et al. Differential evolution as a viable tool for satellite image registration [J]. Applied Soft Computing, 2008, 8 (4): 1455-1462.
[7] 葛宝臻, 周天宇, 陈雷, 等. 基于改进 ISS 特征点与人工蜂群算法的点云拼接方法 [J]. 天津大学学报: 自然科学与工程技术版, 2016, 49 (12): 1273-1280.
[8] 高鹏东, 彭翔, 李阿蒙, 等. 基于混合遗传算法和点面距离测度的深度像配准 [J]. 计算机应用研究, 2007, 24 (12): 354-356, 360.
[9] Khoshelham K, Elberink S O. Accuracy and resolution of kinect depth data for indoor mapping applications [J]. Sensors, 2012, 12 (2): 1437-54.
[10] Nguyen C V, Izadi S, Lovell D. Modeling kinect sensor noise for improved 3D reconstruction and tracking [C]// Proc of the 2nd IEEE International Conference on 3d Imaging, Modeling, Processing, Visualization & Transmission. 2012: 524-530.
[11] 陈晓明, 蒋乐天, 应忍冬. 基于 Kinect 深度信息的实时三维重建和滤波算法研究 [J]. 计算机应用研究, 2013, 30 (4): 1216-1218.
[12] Chow K C, Tsui H T, Tong L. Surface registration using a dynamic genetic algorithm [J]. Pattern Recognition, 2004, 37 (1): 105-117.
[13] Santamaría J, Damas S, Cordón O, et al. Self-adaptive evolution toward new parameter free image registration methods [J]. IEEE Trans on Evolutionary Computation, 2013, 17 (4): 545-557.
[14] Storn R, Price K. Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces [J]. Journal of Global Optimization, 1997, 11 (4): 341-359.
[15] Hansen P, Mladenovic N. Variable neighborhood search [J]. Computers in Operations Research, 1997, 24: 1097-1100.
[16] Dasgupta D, Dasgupta D. Artificial immune systems and their applications

[J]. Lecture Notes in Computer Science, 1999, 1 (4): 121-124.
[17] Zhang Z. Iterative point matching for registration of free-form curves and surfaces [J]. International Journal of Computer Vision, 1994, 13 (2): 119-152.
[18] Yamany S M, Ahmed M N, Hemayed E E, et al. Novel surface registration using the grid closest point (GCP) transform [C]// Proc of International Conference on Image Processing. 1998: 809-813.
[19] Santamaría J, Cordón O, Damas S, et al. Performance evaluation of memetic approaches in 3D reconstruction of forensic objects [J]. Soft Computing, 2009, 13 (8): 883-904.
[20] Silva L, Bellon O R, Boyer K L. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2005, 27 (5): 762-776.
[21] Wachowiak M P, Smolikova R, Zheng Y, et al. An approach to multimodal biomedical image registration utilizing particle swarm optimization [J]. IEEE Trans on Evolutionary Computation, 2004, 8 (3): 289-301.
[22] Men H, Gebre B, Pochiraju K. Color point cloud registration with 4D ICP algorithm [C]// Proc of IEEE International Conference on Robotics and Automation. 2011: 1511-1516.
[23] Men H, Pochiraju K. Hue-assisted automatic registration of color point clouds [J]. Journal of Computational Design & Engineering, 2014, 1 (4): 229-243.
[24] Lai K, Bo L, Ren X, et al. A large-scale hierarchical multi-view RGB-D object dataset [C]// Proc of IEEE International Conference on Robotics and Automation. 2011: 1817-1824.
[25] Falco I D, Cioppa A D, Maisto D, et al. Using an adaptive invasion-based model for fast range image registration [C]// Proc of GECCO' 14 Conference on Genetic and Evolutionary Computation. 2014: 1095-1102.

Note: Figure translations are in progress. See original paper for figures.