Postprint of Point Cloud Registration Algorithm Based on Multi-Feature Matching

Authors: Li Qiang, Gao Baolu, Dou Mingliang

Date: 2018-12-13T00:00:00+00:00

Abstract

To address the problem of the Iterative Closest Point (ICP) algorithm's single rule for searching matching point pairs and low accuracy, a point cloud registration algorithm based on multi-feature matching is proposed. First, an improved adaptive octree algorithm is employed to segment the point cloud; following local fitting of its leaf nodes via Moving Least Squares (MLS), the multi-features of points are computed. Subsequently, a point pair similarity metric based on multi-features is introduced, where point pairs satisfying the similarity constraint are selected as matching pairs, and the rotation matrix and translation matrix are computed to achieve point cloud registration. Experimental results indicate that the algorithm can effectively improve registration accuracy while maintaining high point cloud registration speed, with the accuracy improvement exhibiting an increasing trend as the point set size increases.

Full Text

Preamble

Vol. 37 No. 2
Application Research of Computers
Accepted Paper

Point Cloud Registration Algorithm Based on Multiple-Feature Matching

Li Qiang, Gao Baolu, Dou Mingliang
(College of Information & Computer, Taiyuan University of Technology, Taiyuan 030024, China)

Abstract: To address the problems of the iterative closest point (ICP) algorithm—namely its single-feature search rule and low registration accuracy—this paper proposes a point cloud registration algorithm based on multiple-feature matching. First, an improved adaptive octree algorithm

segments the point cloud, and moving least squares (MLS) is applied to locally fit the leaf nodes for calculating multiple features of points. Then, a point-pair similarity metric based on multiple features is introduced to select matching point pairs that satisfy similarity constraints, from which the rotation and translation matrices are derived to achieve point cloud registration. Experiments demonstrate that the proposed algorithm effectively improves registration accuracy while maintaining high speed, with the accuracy improvement becoming more pronounced as the number of points increases.

Keywords: octree; moving least squares; curvature; point cloud registration; quaternion

---

0 Introduction

Three-dimensional point cloud reconstruction technology has been widely applied in model construction fields such as cultural heritage digitization and industrial manufacturing modeling [1]. Point clouds are typically acquired using laser scanners. However, due to object impenetrability and accuracy requirements, complete surface information often requires multiple scans from different angles, resulting in point cloud data that are not in the same coordinate system. Therefore, registration of point clouds obtained from different perspectives and batches is necessary. The essence of point cloud registration is to solve for the rigid transformation relationship that aligns a source point cloud to a target point cloud.

To date, numerous methods have been proposed for point cloud registration. The iterative closest point (ICP) algorithm introduced by Besl et al. [2] is the most widely used. This algorithm constructs matching point pairs by selecting points with minimum Euclidean distance between target and source point clouds, computes the rotation and translation transformation matrices for these pairs, and iteratively repeats this process after transformation until convergence constraints are met. However, this approach involves substantial computational overhead, and redundant points in regions with indistinct features significantly reduce convergence efficiency.

To address these limitations, scholars have proposed various ICP improvements. Sharp et al. [3] used the distance from a point to its neighborhood centroid as a feature for registration, describing the relative position of individual points within the point cloud. However, the single-feature approach yielded only modest accuracy improvements. Wei et al. [4] constructed matching pairs based on point-to-surface distance, offering a new perspective on feature selection, but poor surface fitting affected accuracy. Yang et al. [5] employed principal curvature constraints and point cloud normal vector angles for initial point set selection, then used inter-point distances and Gaussian curvature to obtain precise matches, though this method required numerous threshold parameters. Elbaz et al. [6] introduced deep neural networks for registration, but the offline training phase required massive amounts of data, making it more suitable for large-scale

terrain scenes. Wang et al. [7] accelerated matching through multi-resolution and introduced a matching degree concept to improve ICP point searching, but the single-feature matching produced erroneous pairs in large point sets, resulting in average accuracy.

This paper proposes a point cloud registration algorithm based on multiple-feature matching. First, an improved adaptive octree algorithm organizes point cloud data in spatial cells. Moving least squares (MLS) is applied to fit local surfaces for each cell, after which multiple features of points are calculated. Then, instead of the single curvature feature used in [7], multiple features are employed, and a multiple-feature-based point-pair similarity metric replaces the matching degree to establish matching pairs. Finally, quaternion-based methods complete the point cloud registration.

1.1 Adaptive Octree Algorithm

In point cloud segmentation, the octree division strategy directly affects storage space consumption and construction time. Adaptive octree, which controls the number of points in each node within a reasonable range through threshold settings while reducing division iterations, is one of the most commonly used segmentation algorithms. However, its threshold setting lacks flexibility: if the threshold is too large, the segmentation effect is compromised; if too small, each node contains too few points. According to the least squares principle—which finds the best function match by minimizing the sum of squared errors—fewer selected points amplify the impact of edge points, outliers, and noise on surface generation, reducing fitting accuracy. For illustration, Figure 1 [Figure 1: see original paper] demonstrates the influence of outliers on different datasets using 2D curve fitting. For cases (a) and (b) in the figure, when outlier A is added during MLS fitting, its impact on the fitted curve in (a) is significantly greater than in (b). Similarly, for complex point cloud models, the segmented object surface often contains continuous smooth regions (such as the rectangular area in Figure 1(c)). If the threshold is too small, the small number of points in each block actually reduces subsequent MLS fitting accuracy.

1.2 Improved Adaptive Octree

This paper proposes an improved adaptive octree algorithm. First, the conventional adaptive octree algorithm is applied with a relatively large threshold for coarse segmentation. Then, the standard deviation of normal vectors is introduced to judge the severity of surface variation in each octree node. If the deviation exceeds a given threshold, further subdivision continues until the minimum threshold for leaf nodes is satisfied. Finally, each leaf node undergoes local fitting. Since the purpose is point cloud registration, the resulting surface stitching problem can be avoided [8].

The segmentation process proceeds as follows: Set the maximum and minimum leaf node point counts as Dmax and Dmin, respectively. In the coarse segmentation phase, recursive division continues until the number of points in a leaf node falls below Dmax. Then, within each leaf node, the PCA algorithm computes

point normal vector information to derive the normal vector standard deviation σ. Given a threshold for normal vector standard deviation, if $\sigma >$ —indicating significant curvature variation in that node region—further subdivision continues until the point count falls below Dmin; otherwise, no further division occurs. Finally, the MLS algorithm fits a local surface z = f(x, y) for data within leaf nodes.

For a point set G, the standard deviation σ of normal vectors is calculated using Equation (1), which reflects the severity of surface curvature variation within G [9]:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left((X_i - \bar{X})^2 + (Y_i - \bar{Y})^2 + (Z_i - \bar{Z})^2\right)}$$

where n is the number of points in region Ω, (X_i, Y_i, Z_i) are the normal vector components, and ($\bar{X}$, $\bar{Y}$, $\bar{Z}$) are their means.

The improved adaptive octree reduces unnecessary segmentation and subsequent fitting operations while effectively decreasing the complexity of surface fitting within each leaf node through normal vector standard deviation evaluation, thereby facilitating rapid and accurate local surface processing. A formal algorithm description is shown in Figure 2 [Figure 2: see original paper].

On region Ω, the fitting function is established as Equation (2):

$$f(x) = \sum_{i=1}^{m} \alpha_i(x) q_i(x) = q^T(x)\alpha(x)$$

where the coefficient vector α(x) = [a_1(x), a_2(x), …, a_m(x)]^T is a function of x, and the basis function vector q(x) = [q_1(x), q_2(x), …, q_m(x)]^T is a k-th order complete polynomial (quadratic basis shown in Equation (3) with m = 6). Equation (4) represents the weighted discrete L2 norm for vector x = [x_1, x_2, …, x_n].

The quadratic basis function is:

$$q(x) = [1, x, y, x^2, xy, y^2]^T$$

The weighted discrete L2 norm is:

$$\|x\|_{2,w} = \left(\sum_{i=1}^{n} w(x - x_i) x_i^2\right)^{1/2}$$

In Equation (5), n is the number of points in region Ω, f(x) is the fitting function, w(x-x_j) is the weight function at x_j, and α(x) contains undetermined coefficients. Through derivation we obtain:

$$\frac{\partial J}{\partial \alpha} = A(x)\alpha - B(x)y = 0$$

which yields:

$$\alpha(x) = A^{-1}(x)B(x)y$$

where:

$$A(x) = \sum_{j=1}^{n} w(x - x_j)q(x_j)q^T(x_j)$$

$$B(x) = [w(x - x_1)q(x_1), w(x - x_2)q(x_2), ..., w(x - x_n)q(x_n)]$$

Substituting Equation (7) into Equation (2) yields the MLS fitting function:

$$f(x) = q^T(x)A^{-1}(x)B(x)y$$

The rotation process is shown in Equation (20). If k is the order of basis functions, then Φ_k(x) is the shape function. When k = 0, the basis function q(x) = {1}, and the shape function becomes the Shepard function shown in Equation (13):

$$\Phi_0(x) = \frac{w(x - x_i)}{\sum_{j=1}^{n} w(x - x_j)}$$

Even when q(x) is polynomial, f(x) in Equation (11) is no longer polynomial. If q C^r and w C^s, then the fitted function f C^{min(r,s)}.

2.2 Multiple Feature Calculation

When quantitatively comparing local geometric features of surfaces, the selected features must be invariant to scaling, rotation, and translation since the data to be registered are two point clouds of the same object in different coordinate systems. After evaluation, this paper employs Gaussian curvature K, mean curvature H, principal curvatures k_1 and k_2, distance to centroid L, and the average M of k-nearest-neighbor normal vector angles to describe local surface features at a point.

Gaussian curvature K measures the total bending degree at a point:

$$K = \frac{LN - M^2}{EG - F^2}$$

Mean curvature H represents the average bending degree:

$$H = \frac{EN + GL - 2FM}{2(EG - F^2)}$$

The two principal curvatures k_1 and k_2 measure bending in specific directions:

$$k_1 = H - \sqrt{H^2 - K}, \quad k_2 = H + \sqrt{H^2 - K}$$

Distance to centroid L describes the point's relative position:

$$L = \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2 + (z_i - \bar{z})^2}$$

For any point p_i in point set P, the average M_i of normal vector angles among its k nearest neighbors reflects local surface sharpness:

$$M_i = \frac{1}{k} \sum_{j=1}^{k} \arccos\left(\frac{n_i \cdot n_j}{\|n_i\| \|n_j\|}\right)$$

In summary, for any point p_i in point set P, the local fitting surface z = f(x, y) for its leaf node is obtained via Equation (10), and its local multiple features are derived through Equations (14)-(19).

3 Point Cloud Registration Based on Multiple-Feature Matching

The classic ICP algorithm employs a nearest-point matching strategy that tends to generate numerous erroneous matching pairs, particularly when overlap between source and target point clouds is small. The improved ICP algorithm in [7] still suffers from single-feature (curvature) matching and poor accuracy. This paper introduces a point-pair similarity concept based on multiple features, employing vector similarity—more suitable for multi-feature matching—as the criterion while adding constraint features, thereby significantly improving matching accuracy.

Point cloud registration essentially solves for rotation matrix R and translation matrix T that transform a source point cloud to align with a target point cloud. As shown in Figure 3 [Figure 3: see original paper], assuming the upper-left GP (gray points) is the target point cloud and the lower-right BP (black points) is the source point cloud, the algorithm computes the translation and rotation matrices from BP to GP to maximize their overlap.

According to this principle, obtaining R and T requires establishing a one-to-one correspondence between points in BP and GP. However, during point cloud acquisition, 3D laser scanners record dense 3D coordinates of measured object surfaces based on laser ranging principles, introducing randomness. Combined with noise and viewing angle effects, no absolute one-to-one correspondence exists between acquired point clouds.

Classic ICP calculates, for each point in the source cloud (BP), its distance to every point in the target cloud (GP), selecting the nearest point in GP as the match. This establishes a one-to-one correspondence from BP to GP, then computes R and T using Equation (20). Since this correspondence is hypothetical, the process iterates until the mean square error falls below a threshold. Thus, ICP essentially solves an optimization problem, using least squares to find R and T that minimize the objective function [11], typically the sum of squared point-to-point distances as shown in Equation (21):

$$f(R,T) = \sum_{i=1}^{N} \|Q_i - (RP_i + T)\|^2$$

where P_i is the source point set, Q_i is the target point set, and N is the total number of matching pairs.

This paper proposes a point-pair similarity concept based on multiple features. In point clouds BP and GP, matching pairs are established by selecting points with highest similarity, improving accuracy and reducing false matches.

First, a multi-resolution keypoint sampling method [7] samples the point cloud data. This approach rapidly registers with few initial points, then substantially increases sampling to improve accuracy, and finally slightly increases sampling for overall convergence. Points are classified into m levels based on normal vector angles from Equation (22). With maximum resolution n and current resolution t ($1 \leq t < n$), the sampling ratio R_{st} for level s ($1 \leq s < m$) is:

$$R_{st} = \text{fix}\left(\frac{\text{count}_s}{\text{count}_m} \cdot \frac{n-t+1}{n}\right)$$

where count_m is the total points at level m, count_s is the total at level s, and fix denotes rounding toward zero.

For any point p_i in P, a multiple-feature vector (p_{i1}, p_{i2}, p_{i3}, p_{i4}, p_{i5}, p_{i6}) is constructed, where p_{i1}, p_{i2} are principal curvatures k_1, k_2; p_{i3}, p_{i4} are Gaussian curvature K and mean curvature H; p_{i5} is distance to centroid L; and p_{i6} is the average normal vector angle M. For p_i, its k nearest neighbors q_j in target set Q are found, and similar feature vectors (q_{j1}, ..., q_{j6}) are built. Similarity is measured using vector cosine similarity from Equation (23):

$$W(p_i, q_j) = \frac{\sum_{k=1}^{6} p_{ik} q_{jk}}{\sqrt{\sum_{k=1}^{6} p_{ik}^2} \sqrt{\sum_{k=1}^{6} q_{jk}^2}}$$

The point q_j with maximum W(p_i, q_j) is selected as p_i's match.

Finally, the quaternion method [13] computes rotation matrix R and translation matrix T between matching pairs.

Complexity analysis shows that similarity calculation W has a dominant quadratic term, yielding O(n^2) time complexity. Since each of m points in P computes similarity with k neighbors, requiring m×k calculations, the overall algorithmic time complexity is O(n^3). Storage requirements grow linearly with point count as each point must store multiple features and its best match, giving O(n) space complexity.

The algorithm flowchart is shown in Figure 4 [Figure 4: see original paper]. Detailed steps are:

a) Let source point set be P and target set Q. Compute multiple features for all points p_i in P.

b) Classify points in P into m levels based on average normal vector angle M_i and set maximum resolution n.

c) Apply multi-resolution keypoint sampling to processed point clouds to obtain resolution t and sampling ratio R_{st}.

d) Find matching points for sampled points using Equation (23).

e) Compute rotation matrix R and translation matrix T from matching pairs using the quaternion method.

f) Transform P using R and T via Equation (20) to obtain new point cloud P'.

g) Repeat steps d)-f) until the objective function is satisfied.

h) If resolution is not n, advance to next resolution and return to step c).

4 Experimental Results and Analysis

The algorithm was implemented in Matlab 2016a on a Windows 7 system with an Intel Core i7 CPU, 8 GB RAM, NVDIA GeForce GT650M graphics, and a 256 GB SSD. Experimental data were obtained from the Stanford 3D Point Cloud Database (Dragon and Bunny models).

4.1 Comparison of Surface Fitting Rates with Different Octree Thresholds

To determine optimal thresholds for the improved adaptive octree algorithm, extensive repeated experiments were conducted using the Bunny model (33,279 points). Initial values were selected using octant division: Dmax first approximated 1/8 of total points (~4000), then halved sequentially; Dmin was set to 1/8 of Dmax; = 0.0001 [9]. Results are shown in Table 1 . Timing reflects total local MLS fitting time based on the improved adaptive octree, providing reference for subsequent processing.

Table 1 lists time costs (in seconds) for different Dmax and Dmin values. Results show minimum total time occurs at Dmax = 1000, Dmin = 125, with time increasing in all directions when varying either parameter. Analysis reveals that larger Dmax reduces segmentation iterations but increases local fitting difficulty, while smaller Dmin reduces fitting difficulty but increases iteration count and system scheduling overhead, raising total time.

After further experiments, parameters with even lower total time were selected: Dmax = 1000, Dmin = 100, yielding 10.312 seconds total time.

4.2 Comparison of Surface Fitting Algorithms

To verify speed and accuracy of the local MLS fitting algorithm based on improved adaptive octree, comparisons were made with standard MLS fitting and algorithms from [14,15]. Using identical Bunny model data and repeated experiments with Dmax = 1000, Dmin = 100, = 0.0001, Table 2 analyzes algorithm speed via block count, segmentation time, fitting time, and total time. Residual values use the mean absolute surface residual [16,17], where lower is better.

Results show direct MLS fitting of unprocessed point clouds is both most time-consuming and yields highest residuals. Reference [14] uses adaptive octree-based MLS fitting, but numerous blocks lead to long fitting times. Reference [15] employs compactly supported radial basis functions after segmentation, producing smooth surfaces with lowest residuals, but large equation system dimensions increase computational overhead. The proposed algorithm reduces octree segmentation through normal vector standard deviation analysis, achieving lower residuals than [14] (indicating more reasonable segmentation) and substantially reducing total time compared to [15] while maintaining similar fitting accuracy.

Figure 5 [Figure 5: see original paper] shows residual plots from four methods (1400 randomly selected points for visualization). Combined point distribution and vertical axis range show [15]'s residuals are closest to the x-axis with smallest range, indicating best accuracy. The proposed algorithm's results are nearly identical to [15] due to normal deviation analysis. Reference [14] shows wider residual distribution, indicating greater surface variation, while standard MLS exhibits nearly random residual distribution with large range, suggesting points are approximately uniformly distributed on both sides of the surface without forming an effective surface model.

4.3 Point Cloud Registration Algorithm Experiments

Local surface fitting serves to obtain multiple point features; the ultimate goal

is point cloud registration. To validate the proposed algorithm's effectiveness, comparisons were made with classic ICP and the improved ICP algorithm from [7].

Since no universal method exists for evaluating registration error due to the lack of one-to-one correspondence between point clouds, the evaluation method from [13] was adopted. Bunny model registration results are shown in Figure 6 [Figure 6: see original paper]. Figure 6(a) shows pre-registration Bunny models from two viewpoints (0° and 45°) with translation, containing 36,580 and 33,279 points respectively. Figure 6(b) shows classic ICP results with significant deviations in complex feature regions like ears and toes. Figure 6(c) shows [7]'s results with noticeable improvement in ear regions but still evident deviations in complex areas, particularly messy edges. Figure 6(d) presents the proposed algorithm's results (point-pair similarity = 99.9%), showing clear edges and superior registration quality.

Table 3 compares registration speed and error across three algorithms. Classic ICP is most time-consuming due to slow matching point search and iteration. Reference [7] uses multi-resolution and kd-tree acceleration with curvature-based matching, improving speed and accuracy moderately. The proposed algorithm, while spending more time computing multiple features, reduces segmentation blocks and local fitting iterations through improved adaptive octree, slightly shortening registration time. Registration error decreases significantly due to both improved segmentation enhancing feature accuracy and multiple-feature similarity reducing false matches.

Further validation used the more complex Dragon dataset (435,545 and 418,387 points). Figure 7 [Figure 7: see original paper] shows pre-registration Dragon point clouds. Due to high point density and small figure size, local enlargements are provided. Figure 8 [Figure 8: see original paper] shows magnified dragon head details. The proposed algorithm produces the cleanest surface and clearest edges. Figure 9 [Figure 9: see original paper] magnifies the sharp horn region. Classic ICP (Figure 9(b)) shows high point dispersion and blurred edges with chaotic central regions. Reference [7] (Figure 9(c)) improves edges but still exhibits high dispersion in sharp regions due to single curvature matching causing high error rates in dense, complex feature areas. The proposed algorithm (Figure 9(d), similarity = 99.99%) achieves clear edges and regular point distribution through multi-feature constraints that reduce matching errors.

Table 4 compares results across three algorithms as point cloud size increases dramatically. All methods show significant time increases, with classic ICP having the largest growth. The proposed algorithm shows slightly greater time increase than [7] due to more feature computation and similarity calculations, but exhibits the smallest error increase, further demonstrating its high accuracy in point cloud registration.

5 Conclusion

This paper addresses the problems of long registration time and low accu-

racy caused by single-feature matching point search in point cloud registration, proposing a multiple-feature matching algorithm. First, an improved adaptive octree algorithm segments point clouds. MLS surface fitting generates local fitted surfaces for computing multiple point features. Then, a point-pair similarity concept based on multiple features is introduced to obtain more accurate matching pairs for registration.

Experiments demonstrate that the method effectively reduces segmentation blocks while appropriately expanding surface fitting data volume, mitigating noise and outlier effects to improve feature accuracy. The multiple-feature-based similarity substantially reduces false matches and enhances registration accuracy.

However, limitations remain: (1) segmentation thresholds require manual tuning based on point cloud size, which is cumbersome; (2) using multiple features increases per-point storage requirements and space complexity. These issues will be addressed in future research.

References

[1] Ge Zhenhua, Wang Peng, Sun Jian, et al. An overview of registration of point cloud data [C]//Proc of the 17th Chinese Conference on System Simulation Technology & Application. Hefei: University of Science and Technology of China Press, 2016: 334-339.

[2] Besl P J, Mckay H D. A method for registration of 3D shapes [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 1992, 14(2): 239-256.

[3] Sharp G C, Lee S W, Wehe D K. ICP registration using invariant features [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2002, 24(1): 90-102.

[4] Wei Shengbin, Wang Shaoqing, Zhou Changhe, et al. An iterative closest point algorithm based on biunique correspondence of point clouds for 3D reconstruction [J]. Acta Optica Sinica, 2015, 35(5): 244-250.

[5] Yang Xiaoqing, Yang Qiuxiang, Yang Jian. Improved ICP algorithm based on normal vector [J]. Computer Engineering and Design, 2016, 37(1): 169-173.

[6] Elbaz G, Avraham T, Fischer A. 3D point cloud registration for localization using a deep neural network auto-encoder [C]//Proc of Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2017: 2472-2481.

[7] Wang Yong, Zou Hui, He Yangming, et al. ICP algorithm based on multi-resolution registration point [J]. Journal of Chinese Computer Systems, 2018, 39(3): 406-410.

[8] Queiroz R L D, Chou P A. Compression of 3D point clouds using a region-adaptive hierarchical transform [J]. IEEE Trans on Image Processing, 2018, 25(8): 3947-3956.

[9] Huang Yuan, Da Feipeng, Tang Lin. Three-dimensional point cloud compression algorithm based on improved octree [J]. Acta Optica Sinica, 2017, 37(12):

133-141.

[10] Zeng Qinghong, Lu Detang. Curve and surface fitting based on moving least square method [J]. Journal of Graphics, 2004, 25(1): 84-89.

[11] Tam G K L, Cheng Z Q, Lai Y K, et al. Registration of 3D point clouds and meshes: a survey from rigid to nonrigid [J]. IEEE Trans on Visualization and Computer Graphics, 2013, 19(7): 1199-1211.

[12] He Y, Liang B, Yang J, et al. An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features [J]. Sensors, 2017, 17(8): 1862.

[13] Wang Xin, Zhang Mingming, Yu Xiao, et al. Point cloud registration based on improved iterative closest point method [J]. Optics and Precision Engineering, 2012, 20(9): 2068-2077.

[14] Qian Guiping. Research on mesh reconstruction and repair from scattered point cloud [D]. Hangzhou: Zhejiang University, 2008.

[15] Li Jia, Duan Ping, Sheng Yehua, et al. Point cloud modeling based on compactly supported radial basis function under KD tree index strategy [J]. Journal of System Simulation, 2016, 28(9): 2154-2158.

[16] Takimoto R Y, Tsuzuki M D S G, Vogelaar R, et al. 3D reconstruction and multiple point cloud registration using a low precision RGB-D sensor [J]. Mechatronics, 2016, 35(1): 11-22.

[17] Altantsetseg E, Khorloo O, Konno K. Rigid registration of noisy point clouds based on higher-dimensional error metrics [J]. Visual Computer, 2018, 34(6): 1021-1030.

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