IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 18, 2019, accepted September 12, 2019, date of publication September 23, 2019, date of current version October 2, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2942127*

# Registration Algorithm for Point Cloud Based on Normalized Cross-Correlation

**YUAN HUANG**[1,2,3,4] **AND FEIPENG DA**[1,2,3]

[1]School of Automation, Southeast University, Nanjing 210096, China
[2]Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing 210096, China
[3]Shenzhen Research Institute, Southeast University, Shenzhen 518000, China
[4]School of Mathematics and Information Science, Nanjing Normal University of Special Education, Nanjing 210096, China

Corresponding author: Yuan Huang (whhbb@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51475092 and Grant 61405034, in part by the Doctoral Fund of Ministry of Education of China under Grant 20130092110027, and in part by the Shenzhen Science and Technology Innovation Committee (STIC) under Grant JCYJ20180306174455080.

**ABSTRACT** We propose a registration algorithm based on neighborhood similarity for 3D point clouds collected by optical measurement and without prior information. The algorithm first applies the improved minimum spanning tree (Prim algorithm) to classify the point cloud in order to obtain the topology information of the data. Specifically, vectors among root nodes and child nodes are processed, and the points on nodes are classified into different levels according to their scanning angle to simplify data and preserve the most representative points. Then, through the perspective conversion between 2D and 3D and according to the corresponding point set obtained by previous classification, the fast normalized cross-correlation (a 2D matching criterion) is applied to determine the relationship between initial characteristic points. Finally, distance constraints remove the errors between point pairs and allow calculating the registration parameters. Experimental results show that the algorithm has high registration accuracy and is suitable for point cloud data obtained by laser and structured light acquisition.

**INDEX TERMS** Minimum spanning tree, normalized cross-correlation, optical measurement, point cloud classification, point cloud registration.

## I. INTRODUCTION

Accurate 3D reconstruction usually relies on optical 3D measurements [1], [2] to acquire surface data from an object by scanning over different views. Hence, point cloud registration from different scanning angles is essential for 3D reconstruction, as it is a factor that determines the achievable accuracy. In objects of regular size, coarse registration is usually supported by placing markers on their surface [3] or using geometric characteristics of the acquired point cloud data. Methods based on these approaches include curvature [4], normal vector [5], and descriptions of characteristics in high dimensions, such as shape [6], contour curves [7], 3D local feature descriptors [8], and others [9]. In addition, the extended Gaussian image and Fourier transform have been used to perform coarse registration [10]. Likewise, accelerated point cloud data can perform coarse registration [11], and fast Gauss transform improved by the dual-tree approach can be used to calculate the correspondence probability matrix. Moreover, Tsallis and Shannon entropies have been employed for transformation estimation considering the weighted least squares to improve parameter estimation and regularization [12]. In [13], selection of key points, calculation of feature descriptors, and the determination and optimization of corresponding points have been combined to perform point cloud registration.

Regarding fine registration, the mainstream is the iterative closest point (ICP) algorithm [14] and its improvements [15], [16], [31]. Several studies have been devoted to further improve fine registration [17], and multiple-view registration has been also investigated [18]. However, the abovementioned algorithms start from the 3D perspective. Although 3D data has high reliability in terms of information accuracy and security, it seems that it is not necessarily the optimal solution to consider the problem only from a pure 3D perspective [39].

On the other hand, coarse matching for point clouds based on hierarchical normal space sampling has been

The associate editor coordinating the review of this manuscript and approving it for publication was Tallha Akram.

investigated [19]. This algorithm hierarchically groups point cloud data according to the distribution of normal vectors, thereby expanding the normal space sampling. The hierarchy allows to adaptively search for corresponding point pairs using geometric information, and the algorithm terminates when an accurate initial pose for registration is obtained. An algorithm for registration between a large-scale and a close-proximity scanned point cloud has been proposed [20]. The algorithm proposes an artificially designed key point descriptor, super-point, which uses geometric information for data transformation and an autoencoder based on a deep neural network to encode the local 3D geometry. In addition, a multidirectional affine registration algorithm based on statistical and shape features of point clouds introduces the similarity of global vector features and retrieves the scaling factor that maximizes similarity [21]. Then, the algorithm applies rigid registration using the estimated affine factors. In [22], a different approach for registration is developed by assuming that point clouds are rigid bodies composed of particles. The algorithm is based on principles of mechanics and thermodynamics, and the resulting registration framework is compact and robust, supports physics-based registration and uses driving forces according to the desired behavior. Considering possible false correspondences or outliers during 3D point cloud registration, guaranteed outlier removal reduces the data input into a smaller set [23], ensuring that any false correspondence is disregarded in the global optimal solution. As set reduction is performed via purely geometric operations, it is highly reliable. The iterative closest optimal plane is based on the ICP algorithm [32], and point cloud registration is accomplished by searching the optimal plane over a curved surface. In [33], a method to search for corresponding points based on stereo lithography of triangular meshes in a CAD model is proposed, where a dynamic adjustment factor aims to reduce the number of iterations and accelerate registration. In [34], a normal vector angle of the 3D model is used, and heuristic search is integrated into the ICP algorithm for improved efficiency, especially when registering complex surfaces. Zhu *et al.* [35] proposes an effective approach for multi-view registration of range scans using a spanning tree. The algorithm uses the structural features of spanning tree and can achieve global registration, reducing the accumulated errors at the same time. Xu *et al.* [36] design a novel objective function, which can avoid the phenomenon that the scale factor converges to zero. In [37], an algorithm based on K-mean clustering with high robustness is proposed to solve multi-view registration problems. Zhou *et al.* [40] proposed a fast registration algorithm that does not involve iterative sampling, model fitting or local refinement, and further extends to global registration of multiple partially overlapping surfaces. In [41], a hierarchical Gaussian Mixture Model (GMM) based on tree structure is used in registration algorithm.

Although current technology enables to acquire more information with higher accuracy, expensive calculations and low algorithm robustness can result from redundant information. Unlike 3D registration, research on 2D image registration is more mature. In fact, extraction algorithms of 2D image characteristics are very accurate, and many methods for characteristic detection and extraction with high robustness have been proposed. For example, the scale-invariant feature transform [24] proposed by Lowe has been widely applied in different settings to 2D images. Hence, many studies have been aimed at extending 2D registration and matching to 3D point cloud registration. In such studies, 3D registration is often supported by the scale-invariant feature transform [25]. In this paper, we propose an algorithm that applies the fast normalized cross-correlation (NCC) [26], which is a matching criterion for 2D grayscale images, to calculate the degree of similarity between neighborhoods of two pixels from two images. As a result, valid corresponding pairs of pixel areas can be obtained. We apply this method in three dimensions to the problem of point cloud registration, thus establishing an algorithm that leverages neighborhood similarity. We hope that 2D algorithm can be organically combined with 3D algorithm in the actual measurement, acquisition, as well as the manufacture and construction of identification equipment.

## II. PROPOSED POINT CLOUD REGISTRATION

In order to obtain topological information of point cloud data, the proposed point cloud registration algorithm first classifies points in the point cloud using the minimum spanning tree. This process simplifies data and preserves representative points. Consequently, possible redundancy caused by massive data is also mitigated. Then, we use the point cloud classification to acquire approximate corresponding point sets. Next, we apply the NCC 2D matching criterion to determine corresponding point pairs and remove error pairs via distance constraints. Finally, we obtain the rotation and translation parameters through a quaternion-based algorithm. The algorithm for the proposed point cloud registration is summarized in Fig. 1.

### A. POINT CLOUD SIMPLIFICATION

Let the currently processed point clouds be denoted as $P$ and $Q$, where the latter is the reference point cloud. Based on the improved minimum spanning tree algorithm, divide the point cloud into different point cloud units by using the approach in [27] and build the tree corresponding to the point cloud. The proposed algorithm classifies each point cloud throughout the tree structure.

The minimum spanning tree has two common algorithms, the prim algorithm and the kruskal algorithm. For point cloud, Kruskal algorithm is not suitable. The time complexity of the prim algorithm is $O(n^2)$ and time efficiency is low. The improved algorithm in this paper combines each unit point cloud after dividing them so that computation of the entire spanning tree is reduced. This can greatly improve time efficiency. The algorithm is summarized as follows:

(a) An octree cut of depth n is performed on each unit point cloud, and each point cloud belongs to a small cube.

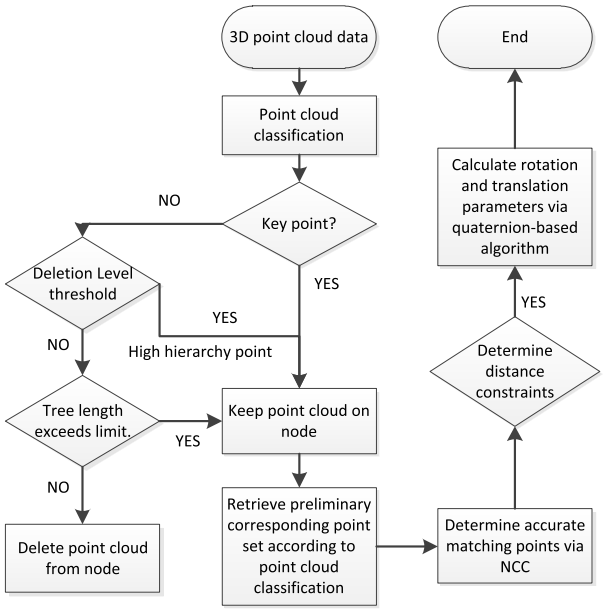



**FIGURE 1.** Flowchart of proposed point cloud registration algorithm.

(b) Find adjacent non-empty cubes centered on each cube and combine them into one calculation unit. Reducing a point to find the query range of its nearest point can effectively reduce the time of the spanning tree algorithm.

The flow of the algorithm is as follows:

Set $G = (V, E)$. The set of vertices of the spanning tree is U.

1) Look for a point o in the unit point cloud and put o in U.
2) Find the nearest point $S_1'$ of o in its cube and its adjacent cube, namely, its computational unit. $S_1'$ and o points form the edge. Join the spanning tree, and $S_1$ joins U. At the same time, the predicted value $S_1'$ of $S_1$ is obtained by $S_1$ according to the linear prediction rule. $S_1'$ joins its cube, and $S_1'$ joins U.
3) Continue to find the nearest point $S_2$ of V-U and U. $S_2$ joins U. If the point closest to $S_2$ is the predicted value in U, the edge of $S_2$ and its true value is added to the minimum spanning tree, and the predicted value $S_2'$ of its true value is also added to U. Otherwise $S_2$ and the real value directly constitute the minimum spanning tree.
4) Check whether the V-U is empty. If it is empty, stop building minimum spanning tree. Otherwise, jump out of the computational unit and find the nearest point of V-U and U in the entire point cloud unit, and then repeat step 3.
5) V-U is empty, complete the minimum spanning tree.

We extracted a unit point cloud from the bunny model (Fig. 2) from the Stanford 3D Scanning Repository and a 3D face model (Fig. 3) collected at our laboratory. Then prim algorithm and the improved algorithm in this paper are used to process the data. The results are shown in Figure 4 and Figure 5.

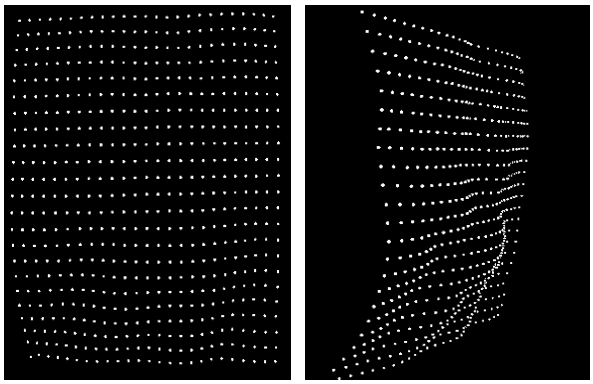



**FIGURE 2.** Point cloud from a bunny in the (a) front and (b) side views.

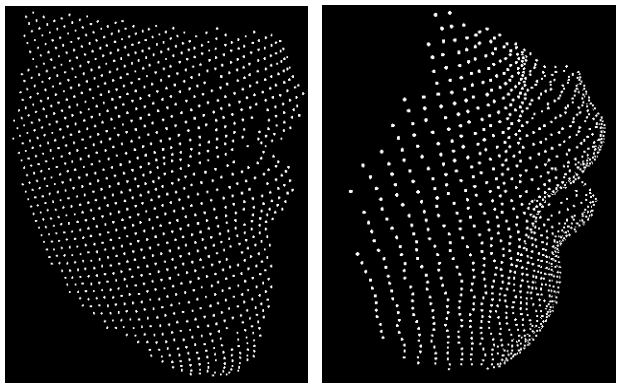



**FIGURE 3.** Point cloud from a face in the (a) front and (b) side views.

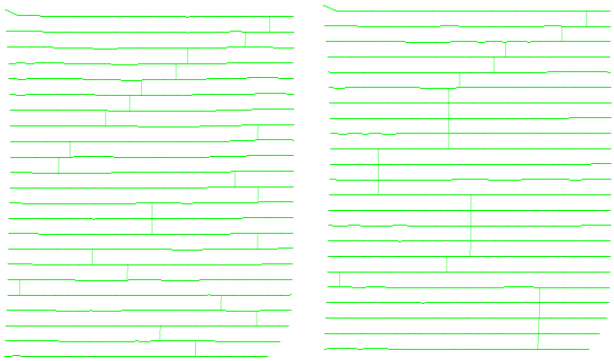


**FIGURE 4.** The visual map of minimum spanning tree of unit bunny point cloud model. Prim algorithm (left) improved prim algorithm (right).

We can see that the rabbit point cloud provided by Stanford University is extremely high in quality, well distributed, and has a small curvature variation. The tree structure of improved prim algorithm minimum spanning tree and original prim algorithm minimum spanning is very similar. The curvature variation of face point cloud data collected in the lab is relatively big. The tree structures of improved prim algorithm minimum spanning tree and prim algorithm minimum spanning tree change greatly. Therefore, compared with the

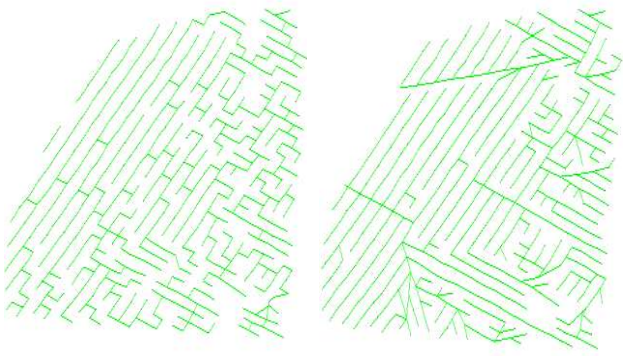


**FIGURE 5.** The visual map of minimum spanning tree of unit face point cloud model. Prim algorithm (left) improved prim algorithm (right).

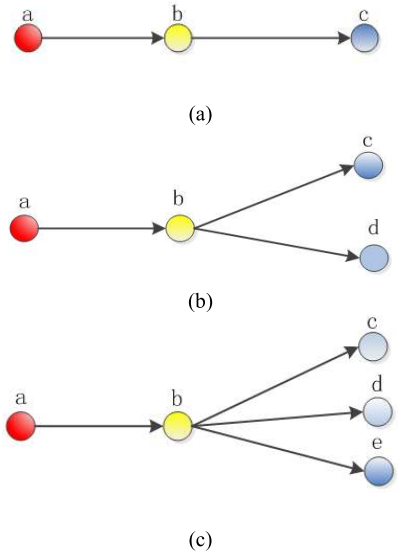



**FIGURE 6.** Cases in minimum spanning tree to simplify point cloud: (a) one, (b) two, and (c) three (or more) child nodes.

original prim algorithm, the tree structure of the improved prim algorithm is more frame-oriented, the lines are more straight and longer, and it is easier to describe the curvature of the point cloud, which is more conducive to simplification of point cloud.

For each node in the minimum spanning tree, excluding root and leaf nodes, there are three possible cases, namely, one, two, or more child nodes. Different methods are adopted to label the node level according to the case:

1) Node b in point cloud $P$ has only one child node, as shown in Fig. 6(a). Determine angle $\theta$ between vectors $\overrightarrow{ab}$ and $\overrightarrow{bc}$. Then, calculate $\theta$ using (1) and determine the level of node b accordingly.

$$\theta = \arccos \frac{\left(\overrightarrow{ab} \cdot \overrightarrow{bc}\right)}{\left(\left|\overrightarrow{ab}\right| \left|\overrightarrow{bc}\right|\right)} \tag{1}$$

2) Node b in point cloud $P$ has two child nodes, as shown in Fig. 6(b). Determine the normal vectors to planes abc and abd. Then, calculate the angle between the two normal vectors according to (2) and determine the level of node b based on angle $\theta$.

$$\theta = \arccos \frac{(\vec{n}_{abc} \cdot \vec{n}_{abd})}{(|\vec{n}_{abc}|\,|\vec{n}_{abd}|)} \tag{2}$$

3) Node b has more than two child nodes, as shown in Fig. 6(c). Determine the normal vectors to planes abc, abd, and abf. Then, calculate the minimum angle between any two of the normal vectors in (3) and determine the level of node b based on minimum angle $\theta$.

$$\theta = \arccos \left\{ \begin{array}{l} \dfrac{(\vec{n}_{abc} \cdot \vec{n}_{abd})}{(|\vec{n}_{abc}|\,|\vec{n}_{abd}|)} \\ \dfrac{(\vec{n}_{abc} \cdot \vec{n}_{abe})}{(|\vec{n}_{abc}|\,|\vec{n}_{abe}|)} \\ \dfrac{(\vec{n}_{abd} \cdot \vec{n}_{abe})}{(|\vec{n}_{abd}|\,|\vec{n}_{abe}|)} \end{array} \right. \quad \left( \begin{array}{l} \theta \text{ takes the} \\ \text{minimum value} \end{array} \right) \tag{3}$$

**TABLE 1.** Number of points in point clouds according to their level.

| Level | 1 | 2 | 3 | 4 | 5 | X |
|---|---|---|---|---|---|---|
| **Bunny (unit)** | 410 | 109 | 22 | 4 | 1 | 29 |
| **Face (unit)** | 316 | 178 | 74 | 23 | 23 | 177 |

After processing all the point cloud data, we classify the data into five levels from 0 to 50° at increments of 10° according to the angle between vectors or normal vectors. Above 50°, the processed points can be considered as representative. To prevent information loss from the applied point cloud simplification, we set the maximum length of the spanning tree to be $n$. Specifically, if $n$ consecutive point clouds (spanning tree nodes) throughout the tree structure are candidates for removal, the $n$-th point is maintained.

The point cloud hierarchy of these point clouds retrieves the distribution listed in Table 1. Node, starting nodes, and leaf nodes above 50° are set as key points that cannot be removed from the data. The remaining points are divided into the five hierarchies according to their angles. As the level increases, the importance of the points also increases. Therefore, we can simplify the point cloud according to the level by maintaining the most important feature points and reduce the computation effort in the subsequent registration. Figs. 7(a) and (b) show the point cloud simplification after removing level 1 and both levels 1 and 2, respectively, from the rabbit model. Fig. 8 shows the same type of point cloud simplification applied to the face model. In relatively flat regions of a surface, the point cloud level is relatively low, and hence holes can easily appear after simplification.

To remove the holes, we use the maximum limit tree length, $n$, as defined above. Fig. 9 shows the effect diagram of on the point clouds after removing level 1 and setting the maximum limit tree length to 3. The large holes in Figs. 7 and 8 disappear, and there are relatively many points in the original

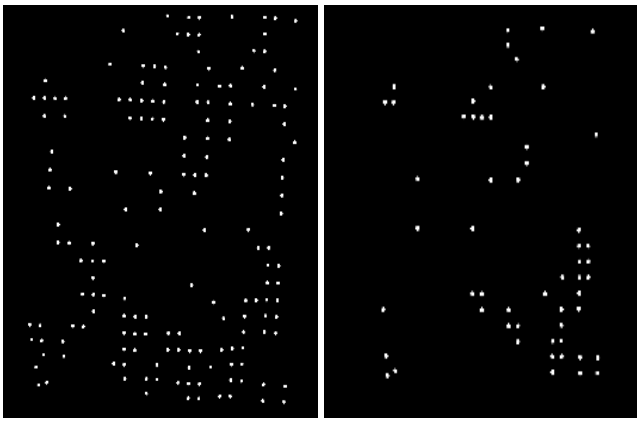



**FIGURE 7.** Point cloud from bunny after removing nodes in (a) level 1 and (b) levels 1 and 2.

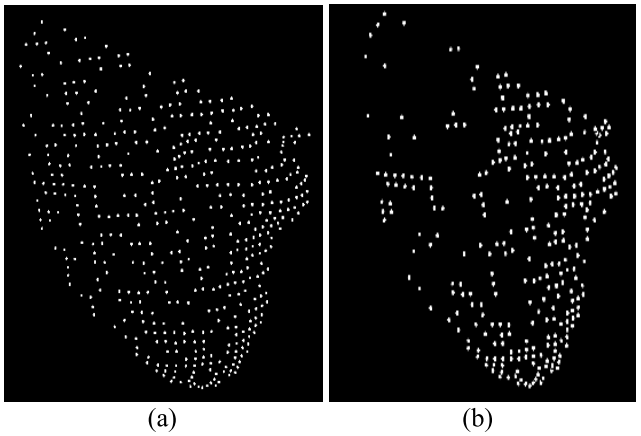



**FIGURE 8.** Point cloud from face after removing nodes in (a) level 1 and (b) levels 1 and 2.

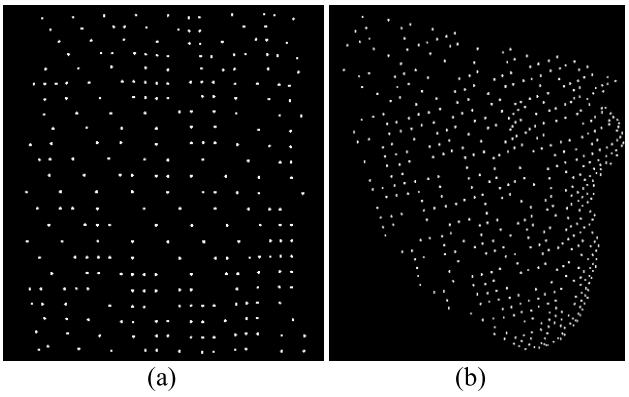



**FIGURE 9.** Simplified diagram after removing nodes in level 1 with maximum limit tree length of 3 for the point clouds from (a) bunny and (b) face.

curvature, thus preserving detailed features of the original point cloud.

Note that through the algorithm above, some points in the point cloud might be processed several times, but their minimum level prevails. As the point cloud on the minimum spanning tree is divided into different levels, initially deleting

**TABLE 2.** Number of points after different point cloud simplifications.

| | Original | Deletion of level 1, tree length of 3 | Deletion of levels 1 and 2, tree length of 5 |
|---|---|---|---|
| **Bunny** | 31,607 | 16,998 | 11452 |
| **Face** | 15,912 | 11,761 | 6494 |

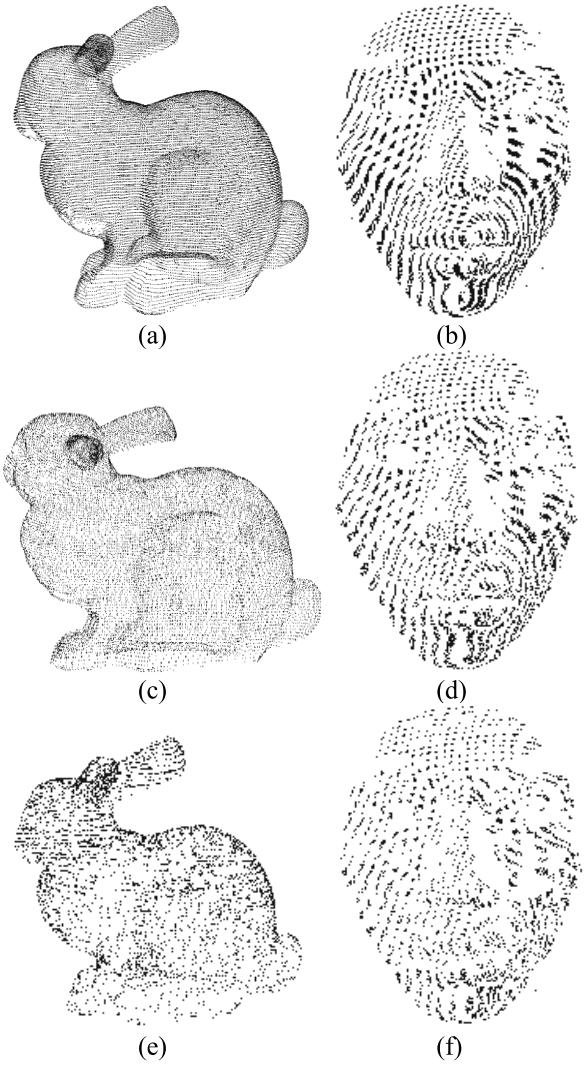



**FIGURE 10.** Data simplification effect under different removal of levels and tree length limits. Original point clouds of (a) bunny and (b) face. Deletion of level 1 and limit tree length of 3 for point clouds of (c) bunny and (d) face. Deletion of levels 1 and 2 and limit tree length of 5 for point clouds of (e) bunny and (f) face.

the lower levels retains the details of the point cloud. Fig. 10 shows different simplification effects by adjusting the level deletion and setting the maximum limit tree length. Table 2 lists the number of the original number of points in the two models and that after the simplifications. Increasing the level of deletion and the maximum limit tree length, the simplification rate is increased at a tradeoff with the point cloud details. After heuristically evaluating several combination of level deletion and tree length integrated with the proposed registration algorithm, we found that removing levels 1 and 2

and setting the limit tree length to 5 is the most suitable combination for simplification.

By analyzing relations among neighbors in the tree structure and adjusting the density of point cloud simplification, the algorithm can prevent holes and thus information loss. As the simplified point cloud has approximately the same information as the original model, it does not undermine registration.

### B. ACQUISITION OF CORRESPONDING POINT SETS

For to-be-registered point cloud data $P$ and $Q$, we can get data sets $P_S$ and $Q_S$ after simplification by the above method. Consider the key points of $P_S$ and $Q_S$ as $Pt = \{pt_1, pt_2, \cdots pt_m\}$ and $Qt = \{qt_1, qt_2, \cdots qt_n\}$. In the sets, m and n are the number of key data points of $P_S$ and $Q_S$. Generally, high-dimensional features have high computational complexity, while low-dimensional features have less information and thus feature recognition is low. Considering previous simplified grading processing and the subsequent introduction of NCC, only angle is used as a basic geometric feature to obtain corresponding point set. Taking point set $Pt$ as an example, the geometric features are as follows:

For each $pt_i$ in set $Pt$, we apply (1)–(3) to the point to obtain corresponding angle. Take it as the local characteristics at this point and set it as:

$$f(pt_i) = f(\theta) \tag{4}$$

For each point $pt_i$ in the point set $Pt$ searches for the corresponding point in the point set $Qt$ If $qt_j$ is the corresponding point or approximate corresponding point of the $pt_i$, the two points should be the same or approximately the same for the geometric features proposed above. Therefore, we propose the following conditions. For a point $pt_i$ in the point set $Pt$ and a point $qt_j$ in the point set $Qt$, as long as the condition is met, we consider them to have a corresponding relationship. The conditions are as follows:

$$\left|f(pt_i) - f(qt_j)\right| \Big/ \left|f(pt_i) + f(qt_j)\right| \leq \sigma_1, \tag{5}$$

where $\sigma_1 = 0.01$, the above relationship is used for filtering and the point correspondence is initially established. Here, one-to-one corresponding relationship of the points is not considered. Only consider whether the conditions are met to establish the corresponding point pair relationship. Since there are many regions with similar features in the point cloud, there is often a one-to-many situation in corresponding relationship of points here. This way, we build matching point set $M_p$ with uncertain number of elements for each $pt_i$ in $Pt$. The number of points in set is indefinite.

### C. POINT CLOUD REGISTRATION

NCC is a matching criterion for 2D grayscale images. The NCC coefficient at pixel location $(x, y)$ in two images is given by (6), as shown at the bottom of the next page, where $I$ is the image to be matched image, $T$ is the sub-image area of size $M \times N$ in a template image, $u(I_c(x, y))$ is the mean grayscale value of the $M \times N$ image area located at $(x, y)$ in $I$, and $u(T)$ is the mean grayscale value of $T$.

Higher NCC implies closer neighborhood similarity between two pixels. As this criterion is widely used given its high robustness, we adopt it to calculate the neighborhood similarity of 3D points, which are regarded as pixels in 2D images. Specifically, the angle of each point in the point cloud obtained using (1)–(3) is regarded as the grayscale value. Therefore, NCC matching is transformed from a 2D neighborhood into 3D space to calculate the similarity of neighborhoods in three dimensions.

We test approximately fitting points according to the NCC and calculate the characteristic similarity of neighborhoods between point pairs using (7), as shown at the bottom of the next page, where $NCC(pt_i, qt_j)$ is the similarity between point pair $(pt_i, qt_j)$. Like in (4), the values of $f(pt_i)$ and $f(qt_j)$ represent the local characteristics of points from the angles of vectors or normal vectors. Variables $u1$ and $u2$ represent the mean value of the characteristic inside the corresponding neighborhood, and $k$ is the number of points in the neighborhood. We then calculate the average distance between two areas of the point cloud and set the smallest value as $r$. A radius of $6r$ is considered from each point to form a sphere, whose points are set to be a neighborhood. Hence, $k$ is the smallest number of points in the compared neighborhoods.

We set threshold $\sigma_2 = 0.9$ and search in set $M$ composed of points $qt_j$ at the same level of $Pt$ for points with the highest neighborhood similarity to each $pt_i$ in $Pt$. Relation $NCC(pt_i, qt_j) > \sigma_2$ should also be satisfied. Then, $qt_j$ is set to be the only matching point of $pt_i$. Points in point set $Pt$ that do not satisfy the conditions above are considered to have no matching points.

Note that a problem arises when establishing the correspondence of neighborhood points using the above method that considers similarity. Pixels in a 2D image are located in a plane, and the correspondence of neighboring points can be determined according to coordinates. However, the distribution of 3D points is not planar. Therefore, determining correspondent points in a neighborhood becomes challenging. To determine the (approximate) correspondence in point pair $(pt_i, qt_j)$, with both sets containing $k$ points, we propose the following method.

Assume point pair $(pt_i, qt_j)$ contains matching points, that is, they are correspondent in two pieces of the point cloud. Then, their normal vectors $n_i^1$ and $n_i^2$ should be aligned, which allows to translate the origin of $n_i^1$ toward that of $n_i^2$ and hence obtain translation transformation parameter $T$. Then, we project the two normal vectors to the $YOZ$ plane, which contains $qt_j$. Similarly, the projected area of $n_i^1$ is rotated to match that of $n_i^2$ to obtain angle $\alpha$, which indicates the rotation of $n_i^1$ around the $x$ axis in the counterclockwise direction. Using (8), as shown at the bottom of the next page, we obtain transformation matrices $T_{Rx}$, $T_{Ry}$, and $T_{Rz}$. By multiplying these three matrices, we can obtain complete rotation $R$. Rotation $R$ and translation $T$ allow to transform neighborhood points of $pt_i$ to match with the coordinates of $qt_j$.

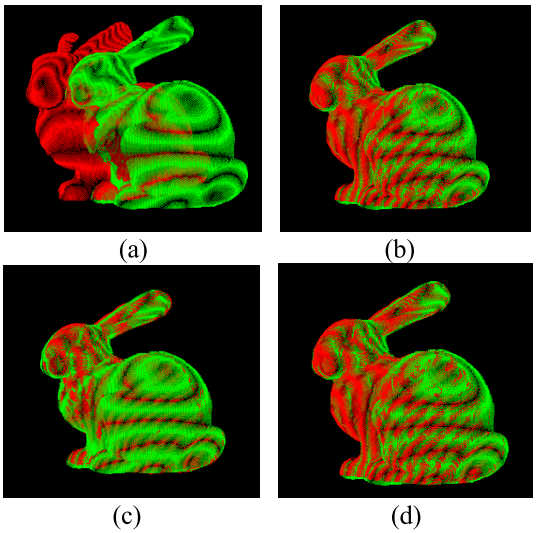



**FIGURE 11.** Point cloud registration of bunny model from two similar scanning angles. (a) Initial poses of point clouds. (b) Registration obtained from the algorithm in [14]. (c) Registration obtained from the algorithm in [12]. (d) Registration obtained from the proposed algorithm.

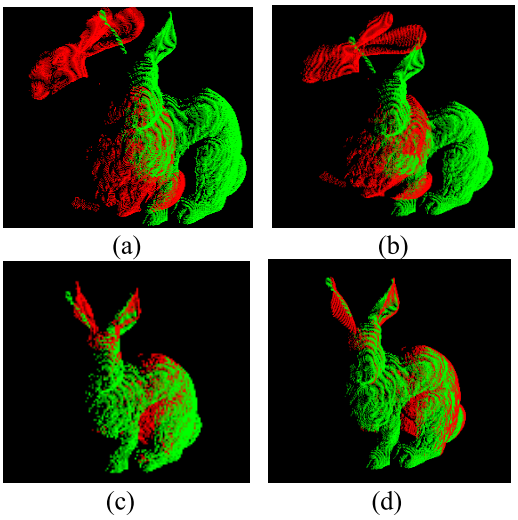



**FIGURE 12.** Point cloud registration of bunny model from two notably different scanning angles. (a) Initial poses of point clouds. (b) Registration obtained from the algorithm in [14]. (c) Registration obtained from the algorithm in [12]. (d) Registration obtained from the proposed algorithm.

Then, we search for the closest point in the neighborhood of $qt_j$ for each point in the neighborhood of $pt_i$. If the distance is below threshold $\lambda$, we regard them as the correspondent point pairs in (9), as shown at the bottom of this page, and the points are fit to calculate $NCC(pt_i, qt_j)$.

When searching for the closest point to each point in the neighborhood, many points in $pt_i$ may correspond to one point in $qt_j$. Therefore, after one point in $qt_j$ has a correspondence, it is not considered during the subsequent search. To mitigate the divergence in the order of magnitude of coordinates in the point cloud data, we set $\lambda = r$.

After establishing the correspondence between neighborhood points, we calculate $NCC(pt_i, qt_j)$. Note that holes in any of the compared point clouds are not considered to calculate $NCC(pt_i, qt_j)$, and hence neighborhood similarity is robust in the hole regions. Regarding two regions of point clouds with different point density, the proposed method is also robust. In fact, we determine approximate matching point pairs meeting some conditions for each point and remove points with no matching point pairs. Finally, the set of approximately matching point pairs is given by (9), where $num(Matchingdots)$ denotes the number of elements in the set.

### D. ERROR REMOVAL FROM MATCHING POINTS

We use a distance constraint to verify the approximately matching point pairs obtained using the method described above. If the matching point pairs are correct, any two matching point pairs $(h_i^1, h_i^2)$ and $(h_j^1, h_j^2)$ should satisfy $\mathrm{dist}(h_i^1, h_j^1) \approx \mathrm{dist}(h_i^2, h_j^2)$ with threshold $\sigma_3 = 0.01$. For each point pair $(h_i^1, h_i^2) \in Matchingdots$, we process the remaining number of point pairs excluding $(h_j^1, h_j^2)$, which must satisfy the distance constraints of $(h_j^1, h_j^2)$. The remaining number of pairs is denoted as $num_i$.

$$NCC(x,y) = \frac{\sum_{j=1}^{N}\sum_{i=1}^{M}[I(x+i,y+j) - u(I_c(x,y))]\cdot[T(i,j)-u(T)]}{\sqrt{\sum_{j=1}^{N}\sum_{i=1}^{M}[I(x+i,y+j)-u(I_c(x,y))]^2}\cdot\sqrt{\sum_{j=1}^{N}\sum_{i=1}^{M}[T(i,j)-u(T)]^2}} \tag{6}$$

$$NCC(pt_i, qt_j) = \frac{\sum_k [f(pt_i)-u1]\cdot\left[f(qt_j)-u2\right]}{\sqrt{\sum_k [f(pt_i)-u1]^2}\cdot\sqrt{\sum_k \left[f(qt_j)-u2\right]^2}} \tag{7}$$

$$T_{Rx} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ T_{Ry} = \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ T_{Rz} = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 & 0 \\ -\sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

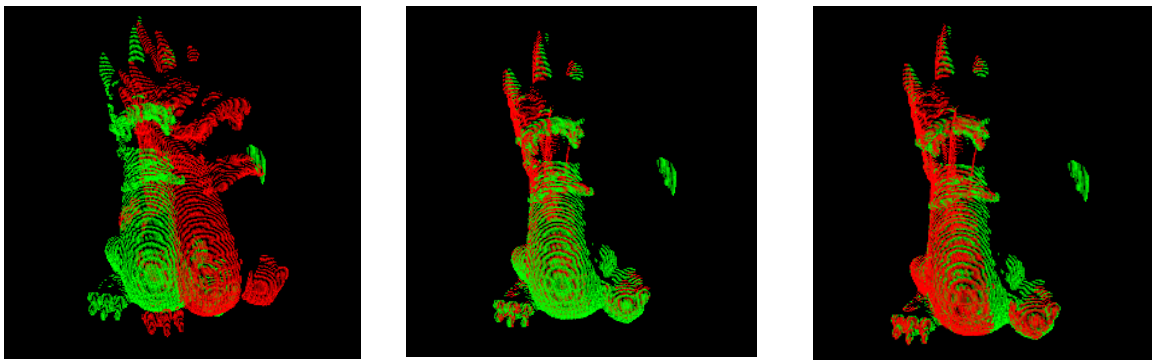
$$Matchingdots = \{(h_i^1, h_i^2) | h_i^1 \in Pt, h_i^2 \in Qt, i = 1, 2, 3 \cdots num(Matchingdots)\} \tag{9}$$

**TABLE 3.** Performance comparison of evaluated registration algorithms.

| | Algorithm in [14] | | Algorithm in [12] | | Proposed algorithm | |
|---|---|---|---|---|---|---|
| | Fig. 9 | Fig. 10 | Fig. 9 | Fig. 10 | Fig. 9 | Fig. 10 |
| **Iterations** | 92 | 103 | 12 | 15 | 10 | 14 |
| **Mean squared error (mm)** | 0.0942 | 25.1926 | 0.0745 | 0.2393 | 0.0717 | 0.1496 |
| **Time (s)** | 62.18 | 71.18 | 28.07 | 33.17 | 28.50 | 31.94 |

**TABLE 4.** Registration performance of proposed and ICP algorithms.

| | Coarse registration using proposed algorithm | | Fine registration[14] | |
|---|---|---|---|---|
| | Dragon | Buddha | Dragon | Buddha |
| **Iterations** | 18 | 29 | 11 | 28 |
| **Mean squared error (mm)** | 0.1012 | 0.2796 | 0.0837 | 0.1424 |
| **Time (s)** | 38.87 | 53.15 | 8.29 | 19.34 |



(a) (b) (c)

**FIGURE 13.** Point cloud registration of dragon model from two scanning angles. (a) Initial poses of point clouds. (b) Coarse registration obtained from the proposed algorithm. (c) Fine registration from the original ICP algorithm [14].

If a point pair $(h_j^1, h_j^2)$ in *Matchingdots* satisfies the following relation:

$$\frac{|\text{dist}(h_i^1, h_j^1) - \text{dist}(h_i^2, h_j^2)|}{\text{dist}(h_i^1, h_j^1) + \text{dist}(h_i^2, h_j^2)} < \sigma_3, \tag{10}$$

then $(h_j^1, h_j^2)$ is a point pair correspondent to $(h_i^1, h_i^2)$ and obeying the distance constraints.

Then, if $(h_i^1, h_i^2)$ is a correct matching point pair, most point pairs in approximate matching point set *Matchingdots* should satisfy (10). Let threshold $\sigma_4$ be 0.9. If $num_i$, obtained by calculation of point pair $(h_i^1, h_i^2)$, satisfies

$$num_i \geq \sigma_4 \times num(Matchingdots), \tag{11}$$

$(h_i^1, h_i^2)$ is a correct matching point pair. Otherwise, it is removed to mitigate the error in matching point pairs. The updated number of approximate matching point pairs after removal results in the following set:

$$Matchingdots = \{(h_i^1, h_i^2) | h_i^1 \in Pt, h_i^2 \in Qt, i = 1, 2 \cdots N\}, \tag{12}$$

where $N$ is the number of correct matching point pairs after removing point pairs not satisfying the distance constraints.

### E. CALCULATION OF MATCHING PARAMETERS

After determining the matching point pairs between point clouds $P$ and $Q$, we calculate the rigid transformation matrix between the two regions of corresponding point clouds. Typical methods to obtain this matrix include the least squares [28], quaternion-based algorithm [29], and singular value decomposition [30]. We use the quaternion-based algorithm as it retrieves an accurate transformation matrix comprising rotation $R$ and translation $T$.

## III. EXPERIMENTAL RESULTS

To demonstrate the effectiveness of the proposed registration algorithm, we use the scanned point cloud data from the bunny model over several angles. We also compare the proposed algorithm with those in references [12] and [14]. To verify its general applicability, we also apply the proposed algorithm to the registration of point cloud data from models of a dragon and a Buddha, retrieved from the Stanford 3D Scanning Repository. The algorithm was executed on a Windows XP PC with Core E7500 and 2 GB RAM.

Fig. 11 shows the registration of the three evaluated algorithms on the bunny model from two similar scanning angles. When the two point clouds for registration are relatively in the same position and orientation, the three algorithms retrieve

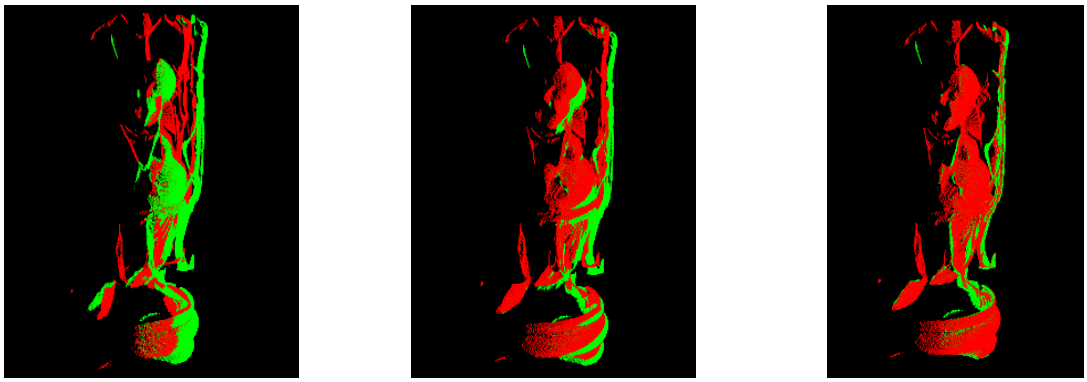



**FIGURE 14.** Point cloud registration of Buddha model from two scanning angles. (a) Initial poses of point clouds. (b) Coarse registration obtained from the proposed algorithm. (c) Fine registration from the original ICP algorithm [14].

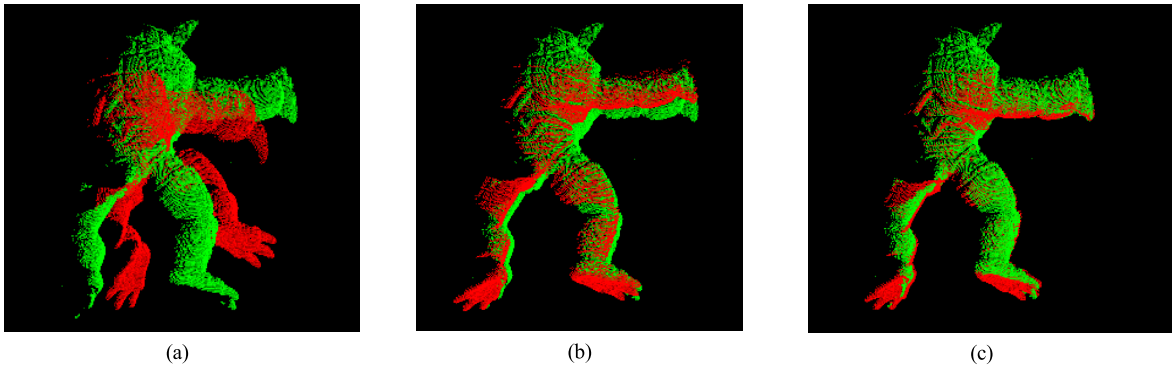



**FIGURE 15.** Point cloud registration of armadillo model from two scanning angles. (a) Initial poses of point clouds. (b) Coarse registration obtained from the proposed algorithm. (c) Fine registration from the original ICP algorithm [14].

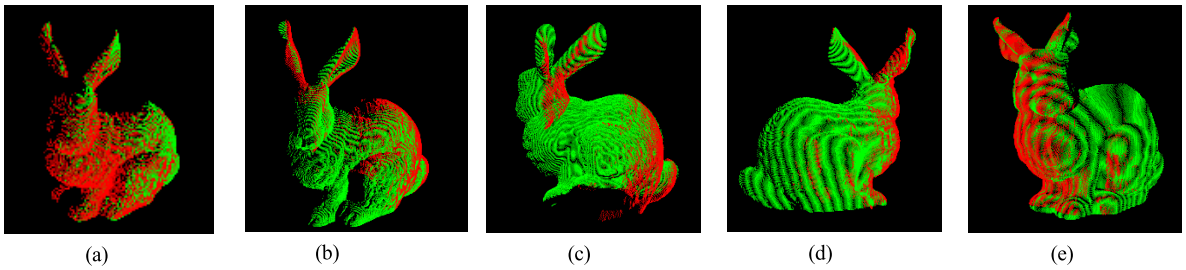



**FIGURE 16.** Registration of Bunny model considering pairwise scanning angles.

suitable registration results. Fig. 12 shows the registration of the three algorithms on the bunny model from two notably different scanning angles. When the relative position and orientation between the two point clouds for registration considerably diverge, the error from the algorithm in [14] is large and registration is distorted. However, the proposed algorithm and that in [12] still retrieve suitable registration results, but the proposed algorithm clearly outperforms the algorithm in [12].

Table 3 lists the number of iterations of the three evaluated algorithms, the registration mean squared error, and the computation time. The computation time of the proposed algorithm includes simplification, which accounts for a large part of the calculations, and thus the time is not considerably reduced compared to the other algorithms.

We also present the combination of the proposed algorithm with the original ICP algorithm for registration, whose results are shown in Figs. 13 and 14 for the dragon and Buddha models, respectively. The proposed algorithm improves the performance of the original ICP algorithm and suitably handles divergent position and orientation between the registered point clouds. Table 4 lists the number of iterations from the algorithm combination, mean squared error, and computation time.

The algorithm in this paper is also compared with the algorithm proposed in [38] and results are shown in Fig. 15.

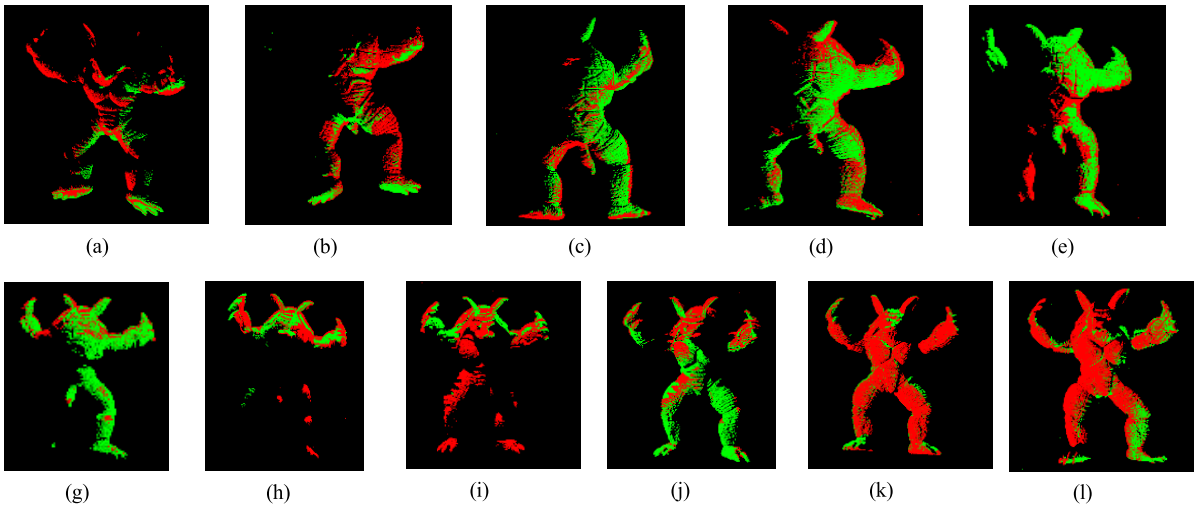



**FIGURE 17.** Registration of armadillo model considering pairwise scanning angles.

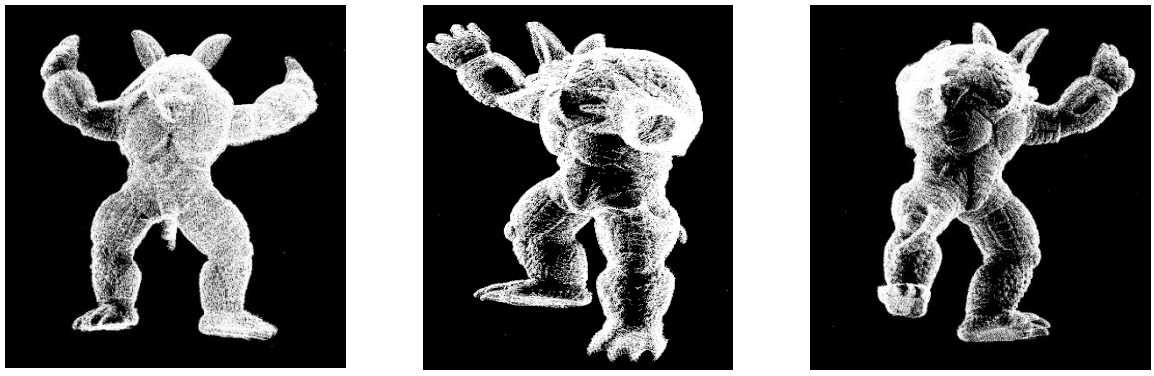



**FIGURE 18.** Final point cloud registration of armadillo model from different views.

Finally, Fig. 16 shows the registration of whole Bunny model from 6 angles pairwise and Fig. 17 shows the registration of an armadillo model (retrieved from the Stanford 3D Scanning Repository) from 12 angles pairwise, from which one point cloud is considered as the reference coordinate system. Fig. 18 shows the final registration results of the armadillo model being transformed into the reference coordinate system. Overall, the registration of the point cloud model is accurate, but some errors remain in areas from the feet and fingers. The error results from accumulation over the several iterations of registration. Hence, optimization is required for improved registration accuracy.

## IV. CONCLUSION

Modern point cloud acquisition systems allow to obtain increasing amount of point cloud data with increasing accuracy. Hence, expensive computations result when directly processing the data. Also, considering the problem from a pure 3D perspective is not the only solution. Based on current technology and under certain conditions, 2D perspective and 3D perspective can be converted to each other, and can complement each other. When dealing with 3D point cloud date obtained from laser and structured light acquisition, We use a minimum spanning tree to simplify the point cloud data into different levels. The information from the levels can be used for subsequent registration. In addition, NCC is applied in the 3D data to improve registration.

Although point cloud classification and simplification impedes a substantial reduction of the computation time, registration accuracy is notably improved using the proposed algorithm. When the initial position and orientation of registered point clouds are similar, other algorithms (e.g., original ICP) retrieve suitable registration results. However, large divergence in position and orientation undermines other algorithms if used directly. Therefore, a robust coarse registration, such as that provided by the proposed algorithm, is necessary.

Overall, the proposed and other evaluated registration algorithms of pairwise data can achieve high accuracy. However, after converting the coordinates of multiple point clouds from different scanning angles, measurement and registration errors accumulate and transfer, affecting the final registration accuracy. Hence, errors may be large, and overall optimization should be further investigated and implemented to mitigate them.

## REFERENCES

[1] M. Schaffer, M. Grosse, B. Harendt, and R. Kowarschik, "High-speed three-dimensional shape measurements of objects with laser speckles and acousto-optical deflection," *Opt. Lett.*, vol. 36, no. 16, pp. 3097–3099, 2011.
[2] K. Liu, C. Zhou, S. Wei, S. Wang, X. Fan, and J. Ma, "Optimized stereo matching in binocular three-dimensional measurement system using structured light," *Appl. Opt.*, vol. 53, no. 26, pp. 6083–6090, 2014.
[3] L. X. P. X. Y. Yongkai and L. A. Z. Xiaobo, "A method for global registration of range data combined with markers," *Acta Optica Sinica*, vol. 29, no. 4, pp. 1010–1014, 2009.
[4] R. Rantoson, H. Nouira, N. Anwer, and C. Mehdi-Souzani, "Novel automated methods for coarse and fine registrations of point clouds in high precision metrology," *Int. J. Adv. Manuf. Technol.*, vol. 81, nos. 5–8, pp. 795–810, 2015.
[5] Y. Huang, F. Da, and H. Tao, "An automatic registration algorithm for point cloud based on feature extraction," *Chin. J. Lasers*, vol. 42, no. 3, pp. 242–248, 2015.
[6] X. Luo, Y. Zhong, and L. I. Renju, "Data registration in 3-D scanning systems," *J. Tsinghua Univ.*, vol. 44, no. 8, pp. 1104–1106, 2004.
[7] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Computer Vision—ECCV 2004* (Lecture Notes in Computer Science), vol. 3023, T. Pajdla and J. Matas, Eds. Berlin, Germany: Springer, 2004, pp. 224–237.
[8] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A comprehensive performance evaluation of 3D local feature descriptors," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 66–89, 2016.
[9] G. K. L. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin, "Registration of 3D point clouds and meshes: A survey from rigid to nonrigid," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 7, pp. 1199–1217, Jul. 2013.
[10] A. Makadia, A. Patterson, and K. Daniilidis, "Fully automatic registration of 3D point clouds," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 1297–1304.
[11] Y. Liu, H. Liu, R. R. Martin, L. De Dominicis, R. Song, and Y. Zhao, "Accurately estimating rigid transformations in registration using a boosting-inspired mechanism," *Pattern Recognit.*, vol. 60, pp. 849–862, Dec. 2016.
[12] M. Lu, J. Zhao, Y. Guo, and Y. Ma, "Accelerated coherent point drift for automatic three-dimensional point cloud registration," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 2, pp. 162–166, Feb. 2016.
[13] J. Lu, Z. Peng, H. Su, and G. Xia, "Registration algorithm of point clouds based on multiscale normal features," *J. Electron. Imag.*, vol. 24, no. 1, 2015, Art. no. 013037.
[14] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
[15] T. Zinsser, J. Schmidt, and H. Niemann, "A refined ICP algorithm for robust 3-D correspondence estimation," in *Proc. Int. Conf. Image Process.*, vol. 3, Sep. 2003, pp. II–695.
[16] S. Granger and X. Pennec, "Multi-scale EM-ICP: A fast and robust approach for surface registration," in *Proc. Eur. Conf. Comput. Vis.*, Copenhagen, Denmark, 2002, pp. 418–432.
[17] K. Pathak, A. Birk, N. Vaskevicius, and J. Poppinga, "Fast registration based on noisy planes with unknown correspondences for 3-D mapping," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 424–441, Jun. 2010.
[18] Y. Guo, F. Sohel, M. Bennamoun, J. Wan, and M. Lu, "An accurate and robust range image registration algorithm for 3D object modeling," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1377–1390, Aug. 2014.
[19] Y. Diez, J. Martí, and J. Salvi, "Hierarchical Normal Space Sampling to speed up point cloud coarse matching," *Pattern Recognit. Lett.*, vol. 33, no. 16, pp. 2127–2133, 2012.
[20] G. Elbaz, T. Avraham, and A. Fischer, "3D point cloud registration for localization using a deep neural network auto-encoder," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4631–4640.
[21] C. Wang, Q. Shu, Y. Yang, and F. Yuan, "Point cloud registration in multidirectional affine transformation," *IEEE Photon. J.*, vol. 10, no. 6, Dec. 2018, Art. no. 6901215.
[22] P. Jauer, I. Kuhlemann, R. Bruder, A. Schweikard, and F. Ernst, "Efficient registration of high-resolution feature enhanced point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1102–1115, May 2019.
[23] Á. P. Bustos and T.-J. Chin, "Guaranteed outlier removal for point cloud registration with correspondences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2868–2882, Dec. 2018.
[24] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[25] S. Allaire, J. J. Kim, S. L. Breen, D. A. Jaffray, and V. Pekar, "Full orientation invariance and improved feature selectivity of 3D SIFT with application to medical image analysis," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2008, pp. 1–8.
[26] J. Luo and E. Konofagou, "A fast normalized cross-correlation calculation method for motion estimation," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 57, no. 6, pp. 1347–1357, Jun. 2010.
[27] S. Lv, F. Da, and Y. Huang, "A fast and lossy compression algorithm for point-cloud models based on data type conversion," *J. Graph.*, vol. 37, no. 2, pp. 199–205, Apr. 2016.
[28] H. Shen, F. Da, and J. Lei, "Research of point-clouds registration based on least-square method," *J. Image Graph.*, vol. 10, no. 9, pp. 1112–1116, Sep. 2005.
[29] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 698–700, Sep. 2009.
[30] O. D. Faugeras and M. Hebert, *The Representation, Recognition, and Locating of 3-D Objects*. Newbury Park, CA, USA: Sage, 1986.
[31] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke, "Registration with the point cloud library: A modular framework for aligning in 3-D," *IEEE Robot. Autom. Mag.*, vol. 22, no. 4, pp. 110–124, Dec. 2015.
[32] W. Tao, X. Hua, K. Yu, X. He, and X. Chen, "An improved point-to-plane registration method for terrestrial laser scanning data," *IEEE Access*, vol. 6, pp. 48062–48073, 2018.
[33] W. Li and P. Song, "A modified ICP algorithm based on dynamic adjustment factor for registration of point cloud and CAD model," *Pattern Recognit. Lett.*, vol. 65, no. 11, pp. 88–94, Nov. 2015.
[34] L. Li, X. Cao, and J. Sun, "Three-dimensional point cloud registration based on normal vector angle," *J. Indian Soc. Remote Sens.*, vol. 47, no. 4, pp. 585–593, 2019.
[35] J. Zhu, L. Zhu, Z. Jiang, X. Bai, Z. Li, and L. Wang, "Local to global registration of multi-view range scans using spanning tree," *Comput. Elect. Eng.*, vol. 58, pp. 477–488, Feb. 2017.
[36] S. Xu, J. Zhu, Y. Li, J. Wang, and H. Lu, "Effective scaling registration approach by imposing emphasis on scale factor," *Electron. Lett.*, vol. 54, no. 7, pp. 422–424, 2018.
[37] J. Zhu, Z. Jiang, G. D. Evangelidis, C. Zhang, S. Pang, and Z. Li, "Efficient registration of multi-view point sets by K-means clustering," *Inf. Sci.*, vol. 488, pp. 205–218, Jul. 2019.
[38] J.-H. Back, S. Kim, and Y.-S. Ho, "High-precision 3D coarse registration using RANSAC and randomly-picked rejections," in *Proc. Int. Conf. Multimedia Modeling*. Springer, Cham, Switzerland, 2018, pp. 254–266.
[39] R. A. Cohen, M. Krivokuća, C. Feng, Y. Taguchi, H. Ochimizu, D. Tian, and A. Vetro, "Compression of 3-D point clouds using hierarchical patch fitting," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 4033–4037.
[40] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Proc. Eur. Conf. Comput. Vis.* Springer, Cham, Switzerland, 2016, pp. 766–782.
[41] B. Eckart, K. Kim, and J. Kautz, "HGMR: Hierarchical Gaussian mixtures for adaptive 3D registration," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 705–721.

**YUAN HUANG** received the B.Sc. degree from Nanchang University, in 2010, and the master's degree from the Galway-Mayo Institute of Technology, in 2012. He is currently pursuing the Ph.D. degree with the School of Automation, Southeast University. His main research interests include 3D measurement and image processing.

**FEIPENG DA** received the Ph.D. degree, in 1998. He is currently a Professor and a Ph.D. Supervisor with the School of Automation, Southeast University. His research interests include surface reconstruction, intelligent control, and computer visualization.