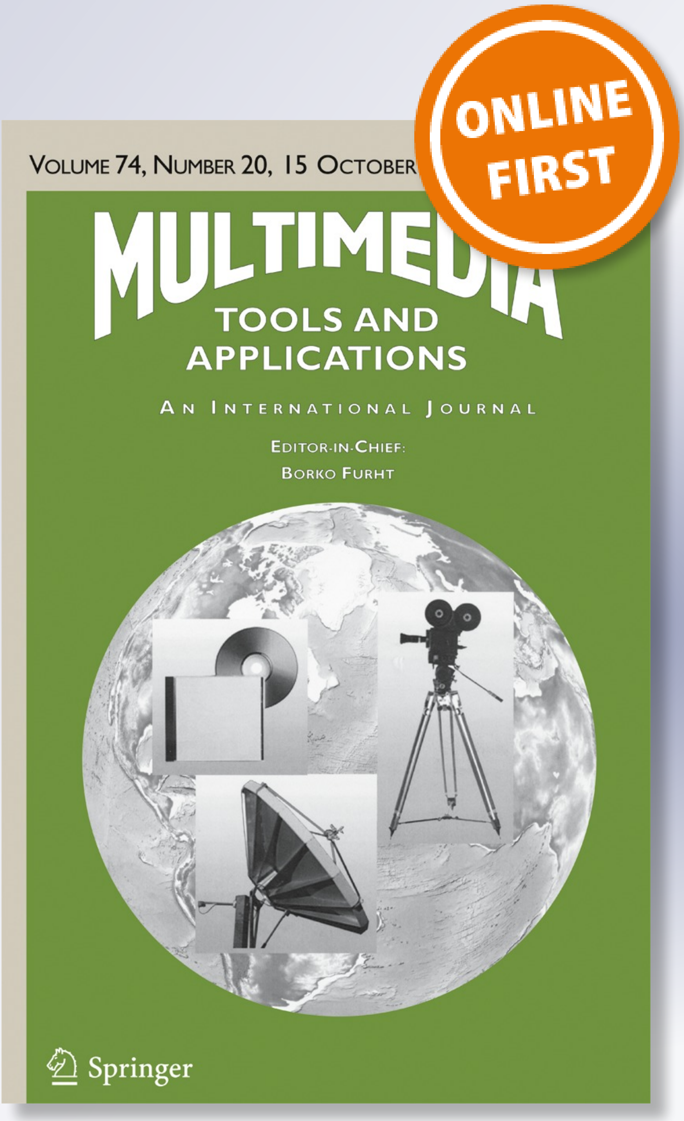
Neighboring constraint-based pairwise point cloud registration algorithm

Nan Geng, Fufeng Ma, Huijun Yang, Boyang Li & Zhiyi Zhang

Multimedia Tools and Applications
An International Journal

ISSN 1380-7501

Multimed Tools Appl
DOI 10.1007/s11042-015-2941-6



Springer

Multimed Tools Appl
DOI 10.1007/s11042-015-2941-6

Neighboring constraint-based pairwise point cloud registration algorithm

Nan Geng[1] · Fufeng Ma[1] · Huijun Yang[1] · Boyang Li[1] ·
Zhiyi Zhang[1]

Received: 31 January 2015 / Revised: 23 July 2015 / Accepted: 8 September 2015

Abstract Three-dimensional point cloud registration is important in reverse engineering. In this paper, we propose a registration method for large-scale 3D point clouds, which is based on neighborhood constraints of geometrical features. The method consists of initial and exact registration steps.In the process of initial registration, we define a new functon that measures feature similarity by calculating the distance function, and in the process of exact registration, we introduce the angle information that improve the accuracy of iterative closest point algorithm. Compared with the traditional feature-based and iterative closest point algorithms, our method significantly reduced the registration time by 11.9 % and has only 1 % of the registration error of the traditional feature-based algorithm. The proposed algorithm can be used to create efficient 3D models for virtual plant reconstruction and computer-aided design, and the registration results can provide a reference for virtual plant reconstruction and growth.

Keywords Point cloud registrations · Weighted sampling · Geometrical feature · Features similar degree · Iterative closest point (ICP)

1 Introduction

Developments in computer-aided design and reverse engineering mean that point cloud data registration has become an important research topic. Point cloud registration generally consists of an initial registration and an exact registration. The initial registration process is

✉ Zhiyi Zhang
zhangzhiyi@nwafu.edu.cn

[1] College of Information Engineering, Northwest A&F University YangLing, YangLing, ShaanXi, 712100, China

Published online: 21 September 2015

affected by many factors. The classic iterative closest point algorithm [2, 3] and its variants are generally used for the exact registration process. Feature-based registration methods have recently become popular, because they do not require an exhaustive search of all points to find the correspondence. The main idea behind feature-based registration is to search for effective matching feature points. Two approaches have been applied to reduce the number of feature points: sampling the data point set, and using feature extraction techniques [1, 27]. In terms of feature implementations, existing feature registration methods can be mainly split into two categories: (1) features derived via artificial targets; and (2) features derived by automatically extracting geometrical primitives.

Luo [18], Wu [25], and Chen [4] proposed the tag method, which manually labels feature information during the measurement process. The registration operation uses this information, because the labels may be the most reliable information. However, properly placing and identifying artificial targets can be costly and laborious. Zhang [30] proposed the center superposition method, and Dai [16] used the main direction of the joint for the initial registration. Diez [8] used layered normal space sampling to capture the corresponding points and implement the initial registration. Jiang [15] used the angle between any point and its neighbor as geometric feature information, and proposed an angle-based feature registration algorithm. However, the estimate of the normal vector may be affected by noise. Oztireli [7] implemented a point-to-point distance constraint between matching point pairs of two point clouds, but the effect of this constraint for excluding the error matching point depends on a threshold value. Other registration methods are required to identity and extract conjugate features such as key points [24], liner segments and curves [9], and planar patches [14]. However, these methods depend on the distinctive features of a point cloud and are sensitive to noise and outliers. Dai [6], Liao [17], and Mian [19] proposed the local feature-based method. In conclusion, existing initial registration methods mainly use geometric feature-based registration. That is, they (1) calculate the geometric feature of a two test point set, (2) compare the feature point pair, and (3) select the valid feature points and calculate the rigid body transformation based on the geometric features of corresponding point pairs. This process completes the initial registration.

Iterative closest point (ICP) uses these feature primitives to further register a point cloud. It is time-consuming, has restrictive requirements for the initial positions of the point cloud data, and can easily fall into local optimums [23, 26]. Many researchers have attempted to improve the ICP algorithm. Rusinkiewicz [20] summarized variants of the ICP algorithm and divided it into six stages. They compared the various improved algorithms for each stage, and proposed a new method with improved convergence. Chen and Mediom [3] used a tangent plane fit to the point clouds, but their algorithm was slow. The method will not converge if there are significant changes to the surface curvature of the model. Zhu [32] proposed a two-way distance field to implement the pre-registration of the model, and further used the ICP to achieve an accurate alignment of the model. Du [11] proposed an affine ICP algorithm, and Yang [28] proposed Go-ICP. ICP and its variants require good approximations of the point clouds, because they are iterative descent algorithms. Finding the closest point is time-consuming, so we must obtain high quality features if we wish to improve the method.

Microsoft Kinects have been used as 3D scanners for obtaining 3D point clouds. A registration method for a Kinect used in an indoor environment has been proposed [21]. However, Kinects does not work normally in an outdoor environment with natural light. It is very difficult to extract feature points from point clouds captured by a Kinect. To satisfy the needs of our project, we researched and developed an independent 3D scanner [31] that functions correctly in an outdoor environment. Our proposed registration method efficiently supports this 3D Scanner.

This paper represents two contributions to point cloud registration. First, we define a new function that measures feature similarity and improves the initial registration method. The initial point sets are sampled multiple times. Then, the initial corresponding point-pairs are determined using the feature similarities calculated using the distance function, which is defined in quintuple space. We show that our feature similarity function works well with the proposed method. Second, we introduce the angle of the vector between the matching point and the center of the global coordinate system and the normal vector of the matched point-pairs, which is used in exact registration process to improve the accuracy of the algorithm.

2 ICP algorithm registration theory

The classical ICP algorithm solves the following least squares mean square error problem. Given two sets of 3D point cloud models, $P = \left\{p_i, i = 1, 2, 3, \cdots, N_p\right\}$ and $Q = \left\{q_j, j = 1, 2, 3, \cdots, N_q\right\}$, let p_i and q_j denote the position vectors of points in 3D space, and N_p andN_q denote the number of reference and target models, respectively. The goal is to estimate a rigid motion with a rotation transformation in 3D space, and translation transformation in 3D space. Let $E\,(R, t)$ denote a rigid transformation. Then, the mean square objective function that we wish to minimize is

$$f(E\,(R,t)) = \frac{1}{N_P}\sum_{i=0}^{N_p} ||Rp_i - q_j^T + t||^2. \tag{1}$$

2.1 Calculating the rigid transformation

1. Calculate the centers of mass of the reference and target models, and estimate their cross-covariance

$$\mu_P = \frac{1}{N_P}\sum_{i=1}^{N_P} p_i, \mu_Q = \frac{1}{N_Q}\sum_{j=1}^{N_Q} q_j, \tag{2}$$

$$\sum\nolimits_{P,Q} = \frac{1}{N_P}\sum_{i=1}^{N_P}[\left(p_i - \mu_p\right) \times \left(q_i - \mu_Q\right)^T] = \frac{1}{N_P}\sum_{i=1}^{N_p}[p_i q_j^T] - \mu_p \mu_Q^T. \tag{3}$$

2. The anti-symmetric matrix $Mat_{ij} = (\sum_{P,Q} - \sum_{P,Q}^T)$ is used to form the column vector $\triangle = [Mat_{23} \quad Mat_{31} \quad Mat_{12}]^T$, and is constructed using (3). Then, we can construct the symmetric matrix $Sym(\sum_{P,Q})$,

$$Sym(\sum\nolimits_{P,Q}) = \begin{bmatrix} tr\sum_{P,Q} & \triangle^T \\ \triangle & \sum_{P,Q} + \sum_{P,Q}^T - tr\sum_{P,Q} I_3 \end{bmatrix}. \tag{4}$$

Here, I_3 is the 3×3 identity matrix. The unit eigenvector $VecE = [r_0 \quad r_1 \quad r_2 \quad r_3]^T$ that corresponds to the maximum eigenvalue of matrix $Sym(\sum_{P,Q})$ is selected as the unit quaternion for calculating the optimal rotation, R. The optimal translation vector is

$$t = \mu_p - R(VecE)\mu_Q. \tag{5}$$

3. Finally, the registration vector $\overrightarrow{T}$ and the mean square point matching error e_{ms} can be obtained using

$$(\overrightarrow{T}, e_{ms}) = \Phi(P, Q), \tag{6}$$

where Φ represents the least squares quaternion operation.

The ICP algorithm can be summarized as: (1) compute the closest point from point sets P and Q; (2) compute the transformation vector $\overrightarrow{T}$ and matching error e_{ms}; (3) apply the registration vector ($\overrightarrow{T}$) to the target point set obtained from the new target model, $P_{k+1} = \overrightarrow{T}(P_k)$; and (4) repeatedly compute the closest points between the reference and new target model, and the transformation vector, until obtaining the complete model shape.

3 Feature extraction

3.1 Estimating neighborhood features

We use two steps to extract the geometric features of the point cloud data in 3D space before registration. First, we select a point set that is stable and distinctive. Second, we determine the neighborhood of a point, and extract its geometric features based on the neighborhood. The curvature and normal vector represent the geometric features of the local area of the point cloud data. The curvature with translation, rotation, and scale invariance [1] represents changes to the shape of the local area of the measurement point. Meanwhile, for point cloud data captured under a different coordinate system, the relative position and topological relationships between feature points stay the same. Therefore, the topological invariance of the feature space can be used to find the matching point pair.

The adjacency features of a point cloud consist of the number of neighboring points, the centers of neighboring points, the point-to-center distances, the normal vectors, and the curvatures. The first three features are easy to determine, so we must mainly develop a way to effectively calculate the last two features. For scattered point cloud data, the least squares fitting method is used to derive the normal vector and curvature, which is a fast and robust method. In this paper, we used the least squares fitting plane to calculate the normal vector [7, 12]. To adapt to the local shapes of different surfaces, we used the least squares fitting surface $S(x, y)$ to compute the curvature using the approach described in [5, 29]. We then calculated the first and second fundamental magnitudes using $L_s = S_{xx}n$, $N_s = S_{yy}n$, $M_s = S_{xy}n$, $E_s = S_x S_y$, $F_s = S_x S_y$, and $G_s = S_y S_y$, substituted into Equations (810). $S_x, S_y, S_{xx}, S_{yy}, and S_{xy}$ are the partial derivatives of surface S(x, y). The quadric surface represents the local area, and (7)-(10) are used to calculate the principal curvatures (k_1, k_2), the mean curvature (H), and the Gaussian curvature (K). That is,

$$k_1, k_2 = H \pm \sqrt{H^2 - K}, \tag{7}$$

$$H = \frac{E_s N_s - 2F_s M_s + G_s L_s}{2(E_s G_s - F_s^2)}, \tag{8}$$

$$K = \frac{L_s N_s - M_s^2}{E_s G_s - F_s^2}, \tag{9}$$

and

$$n = \frac{S_x \times S_y}{|S_x \times S_y|}. \tag{10}$$

The direction of the surface normal vector calculated using this method is not generally consistent. The inconsistency of the normal vector directly affects the accuracy of the curvature estimate, and the normal importantly helps to exclude errors in the matching point pairs. Therefore, we must improve the inconsistency of the normal. In this paper, we use the normal vector spread idea proposed by Hoppe [13] to solve this inconsistency.

3.2 Quintuple feature space

Let $p^f = \left\{ p_i^f \mid p_i^f \in R^3, i = 1, 2, \cdots, s : s < N \right\}$ be the target feature point set, and let $Q^f = \left\{ q_j^f \mid q_j^f \in R^3, j = 1, 2, \cdots, s : s < N \right\}$ be the reference feature point set. For each point in the set $p^f : p_i^f \in p^f$, we search all points that have similar curvature to P^f in the point set Q^f, that is, the points with principal curvatures that satisfy

$$\begin{cases} \sqrt{[k_1(p_i^f) - k_1(q_j^f)]^2 + [k_2(p_i^f) - k_2(q_j^f)]^2} \leq \varepsilon_1, \\ |[k_1(p_i^f) - k_1(q_j^f)] \times [k_2(p_i^f) - k_2(q_j^f)]| \leq \varepsilon_2, \end{cases} \tag{11}$$

where ε_1 and ε_2 represent two adjustable parameters. As calculated above, the corresponding features for a point in the reference and target point clouds can be expressed as quintuple feature vectors

$$[p_i \quad q_j] \tag{12}$$

and

$$[K \quad H \quad \Theta \quad k_1 \quad k_2]. \tag{13}$$

In (12), p_i and q_j represent the corresponding point pair from P and Q. In (13), $K \quad H \quad k_1 \quad k_2$ can be obtained using (7)-(9), where Θ is the normal vector angle. The rotation and translation transformations in 3D space may change the position of the item in (11), but the item in (12) remains the same.

3.3 Definition of the measurement function

In this paper, we use a distance function to measure the similarity of a matching point pair. Let $X = (x_1, x_2, \cdots, x_5)$ denote the feature vector of a matching point. Here, x_1 is the Gaussian curvature mean, x_2 is the average curvature, $x_3 and x_4$ represent two principle curvatures, and x_5 is the mean of the normal vector angles. We use $X_i and X_z$ to represent the features of the registration point p_i^f and its corresponding point q_z^f.

The similarity between p_i^f and q_z^f is defined as

$$S_{iz}(p_i^f, q_z^f) = 1 + \frac{1}{D(p_i^f, q_z^f) + 10^{-6}}, \tag{14}$$

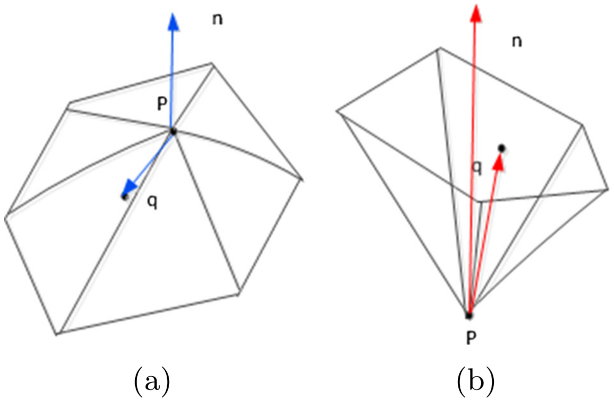


Fig. 1 Here, p is a point in the neighborhood, q is the center of the neighborhood, and n is the normal of p.**a** $(p-q)\cdot n<0$ indicates convex points; and (**b**)$(p-q)\cdot n>0$ indicates concave points

where $D(p_i^f, q_j^f) = ||X_i - X_z||$.

We calculate the similarity of the registration points and any corresponding point. If the maximal similarity satisfies $max S_{ij}(p_i^f, q_z^f) < \tau(z = 1, 2, \cdots, t)$, then we consider that they are not valid matching point pairs. Otherwise, we accept them as matching point pairs. The remaining points are investigated in the same manner, that is, we calculate the similarity of every matching point in the corresponding point set. There may be multiple valid registration points, so we choose the most similar point.

4 Registration method

The registration method based on constraints of neighboring geometric features consists of initial and exact registration steps. The initial registration step effectively adjusts to the orientation of the point cloud, and captures more accurate matching point pairs. This provides a better foundation for the exact registration. In the exact registration process, we add two new geometric feature constraints. In this way, we can simplify the matching point pair and remove incorrect matching point pairs. This accelerates the convergence and improves the registration accuracy.

4.1 Initial registration

The conventional way of extracting a matching point pair is to search all points in the target and reference sets. However, this method has some problems: 1) not all points are suitable for matching; and 2) the method is time-consuming with a time complexity of $O(N_p N_Q)$, whereN_p is the size of point cloud P and N_Q is the size of point cloud {Q}. To improve the matching efficiency, we must select a portion of appropriate matching points from the point cloud.

Weighted sampling For an irregular surface, a raised area is more useful for feature extraction. In this paper, we define the concave and convex points as shown in Fig. 1. We remove concave points because they affect the registration accuracy.

To ensure the integrity of the search space used to determine matching feature points for the reference point set, we only sample the target point set. That is, $N = \eth N$, where $\eth$ indicates weighting factors, N is the size of the target point cloud P, $\eth = \beta \frac{sum}{N}$, and sum is the number of points that have a larger than average Gaussian curvature. $\beta (0 < \beta \leq 1)$ is an adjustable parameter, defined as

$$\beta = \begin{cases} 0 < \beta \leq 0.5, & \text{if} \quad sum \geq 50\,\%, \\ 0.5 < \beta \leq 1, & \text{if} \quad sum < 50\,\%. \end{cases}$$

After weighting the sample of the target point set, the time complexity for calculating the matching points reduces from $O(N_P N_Q)$ to $O(\beta N_P N_Q)$.

In the initial registration process, we adapt the function defined in Section 3.3 to establish matching point pairs. The improved initial registration algorithm is described in Algorithm 1.

Algorithm 1: Improved 3D point cloud initial registration algorithm

```
Input: $P = \{p_i \in R^3\}$    $Q = \{q_j \in R^3\}$
for $p \in \{p_i\}$ do
    $N_{Kneighborhood}$ = KNeighborhood(p), $C_{center}$ = CenterofNeighbor($N_k$)
    foreach $N_{kneighborhood}$ do
        Fit the surface $S_{xy}$, and calculate $k_1, k_2, K, H, n$.
Calculate the average value of Gaussian curvature: $\overrightarrow{K_q} = \frac{1}{N}\sum_{i=1}^{N} K_{q_j^k}$ Sample:
removing concave points and $K_{p_i^k} < \overrightarrow{K_p}$.
foreach $p_i^{ks}$ do
    $(p_i^{ks}, q_j)$=searchP2Q$(p_i^{ks}, Q)$
Establish the correspondence of feature points according to the feature
similarity.
Compute transformation matrices $R$ and $t$ using matched points.
Apply transformations $R$ and $t$ to the 3D point cloud.
return Initial completed
Output: $afterIni = \left\{p_i^{'} \mid p_i^{'} \in R^3; i = 1, 2, \cdots, N^{'}\right\}$
        $afterIniQ = \left\{q_j^{'} \mid q_j^{'} \in R^3; j = 1, 2, \cdots, M^{'}\right\}$
```

4.2 Exact registration

The ICP algorithm is the most widely used exact registration algorithm. It searches for nearest point pairs in two data point sets. Then, the rotation and translation transformations are applied to register of the nearest point pair. The transformations are applied iteratively until the matching criterion is met.

Two groups of points in a unified coordinate system are obtained after the initial registration, as shown in Fig. 2. Meanwhile, we calculate the relationships between the matched points in the local area. During the exact registration process, the initial registration results are used to determine the angles (φ) of the vectors between the adjacent matching point pairs ($p_i^{'}$ and $q_j^{'}$) and the origin of coordinate system after the initial registration. The angles (Θ) of the normal vectors of the two matching points are also calculated. In this paper, we

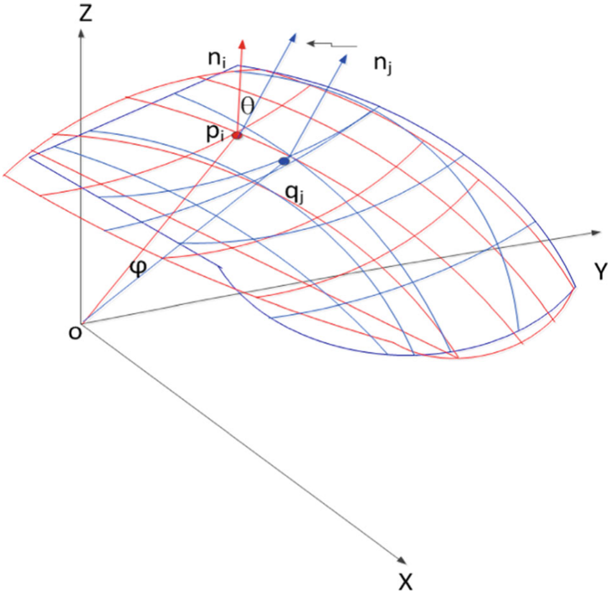


Fig. 2 Two geometric features after the initial registration

include these two features ($varphi$ and Θ) in the ICP iteration process. To further improve the efficiency of the ICP, we optimize the matching process by considering two geometric features. This method is called the improved ICP algorithm, and is given in Algorithm 2.

In Fig. 2, p_i and q_i represent two nearest neighbor matching points, o is the original coordinate system after initial registration, and n_i and n_j represent the normals of p_i and q_j, respectively. We calculate the angle using the cosine law. Let $varphi$ represent the angle between $\overrightarrow{op_i}$ and $\overrightarrow{oq_j}$., and let Θ represent the angle between the normal vectors n_i and n_j. Here, V_f controls the number of valid matching points, and the container C_f is used to save the optimal matching point pair. If the angle between two points satisfies 1) $\sin\varphi < \tau_1 \&\& \sin\theta < \tau_2$, and 2) $|\cos\varphi - \cos\theta| < \tau_3$, then the matching point pair is added to C_f.

We represent the point set after the initial registration step as follows. Let $afterIniP = \left\{p_i' \mid p_i' \in R^3; i = 1, 2, \cdots, N'\right\}$ denote the target point set, and let $afterIniQ = \left\{q_j' \mid q_j' \in R^3; j = 1, 2, \cdots, M_\prime\right\}$ denote the reference point set. The feature point pair after the initial registration step is regarded as the initial value for the exact registration set.

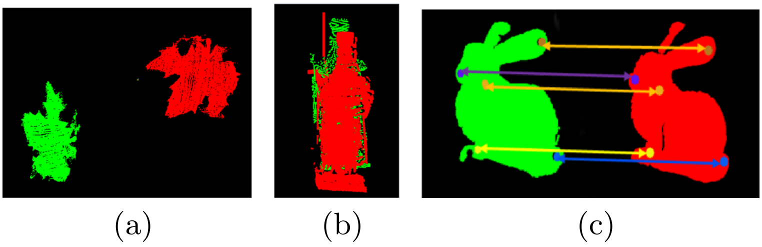


Fig. 3 Point clouds from two different views and the correspondence registration

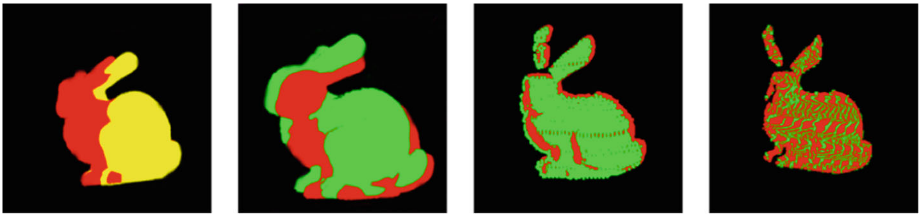
Let $afterIniP^f = \left\{p_i^f \mid p_i^f \in R^3; 0 < i < N'\right\}$ denote the target point feature point set, and let $afterQ^f = \{q_j^f \mid q_j^f \in R^3; 0 < j < M'\}$ denote the reference feature point set.

The improved exact ICP algorithm is described in Algorithm 2.

```
Algorithm 2: Improved exact registration ICP algorithm
Input: $afterIniP = \left\{p_i^{'} \mid p_i^{'} \in R^3; i = 1, 2, \cdots, N^{'}\right\}$
       $afterIniP^f = \left\{p_i^f \in R3; 0 < i < N^{'}\right\}$
       $afterIniQ = \left\{q_j^{'} \in R^3; j = 1, 2, \cdots, M^{'}\right\}$
       $afterIniQ^f = \left\{q_j^f \mid q_j^f \in R^3; 0 < j < M^{'}\right\}$
       $V_f = [3, 9, 12, 16, 50, 100, 180] \qquad C_f = \phi$
for $v_i \in V_f$ do
    while $(e_k - e_{k+1}) < \gamma$ do
        for $p \in afterInip$ do
            $(p_i^f, q_i^f = seachNearest(p_i^f, afterIniP^f))$
        Calculate the angle using the cosine law, the geometric features of
        vector $\overrightarrow{op[i]}, \overrightarrow{oq_j}$, the angle $\varphi$ between $\overrightarrow{op_i}$ and $\overrightarrow{oq_j}$, and the normal vector
        angle $\theta$.
        if $(\sin\varphi < \tau_1)$ && $(\sin\theta < \tau_2)$ && $(|\cos\varphi - \cos\theta| < tau_3)$ then
            if $C_{f-num} < V_{fi}$ then
                $C_f V = Add(p_i^f, q_i^f)$
        if $afterIniP^f$ and $afterIniQ^f$, the registration error is too large then
            Use the quaternion method to calculate transformation matrices $R$
            and $T$ that minimize the mean square error defined as
            $d_k = \frac{1}{N_p}\sum_{i=1}^{N_p} ||q_i^f - Rp_i^f - Q^T||^2$,
            calculate transformation $(\overrightarrow{T}, e_{ms}) = \Phi(P, Q)$,
            and apply the transformation
            $afterIniQ^f(l + 1) = Tran(\overrightarrow{T})afterIniQ^f(l)$
        Apply the transformation to the target point cloud
        return Registration completed
Output: Point cloud data registration
```

5 Experiments

In these experiments, we applied our method to three 3D point clouds: 1) the standard 3D point cloud data of a bunny captured by the computer graphics laboratory of Stanford University [22]; and 2) noisy point clouds of the terracotta warrior model and maple leaf that were captured by our equipment [31]. The experiments were implemented on an Intel (R) core (TM) 2.6 HSF CPU with 4GB RAM. Figure 3 shows the point cloud data for two different views, Fig. 3a has 49,250 points and Fig. 3b has clouds with 90,678 and 104,748 points. Figure 3c demonstrates that the corresponding points of the two views generally refer to the same part of the bunny, which verifies the effectiveness of our algorithm. Throughout these experiments, $\tau_1 = 0.1$, $\tau_2 = 0.2$, $and\tau_3 = 0.005$.



(a) Traditional registration (b) The classical ICP (c) Initial + classicalICP (d) our algorithm

Fig. 4 Registration results using different algorithms

5.1 Comparison of different registration algorithms

The purpose of the first experiment was to verify the global optimality of our new algorithm. We compared four algorithms: 1) traditional initial registration [29]; 2) classical ICP registration [1]; 3) initial registration combined with classical ICP [28]; and 4) improved initial registration combined with improved ICP. We considered how our algorithm performed with the same inputs. The registration results are shown in Fig. 4.

Table 1 contains the least squares errors and time taken to execute the 3D point cloud data registration process. The different algorithms were implemented on a 2.6HZ Intel(R) core with 4.00 GB RAM. In all the comparisons, the models started in arbitrary positions. We reported the average run times over X runs of the algorithm. Figure 4b and Table 1 show that, without an optimal initial registration, the classical ICP registration is not accurate or efficient. Figure 4d and Table 1 demonstrate that our algorithm is more efficient and accurate. The key point set was extracted before registration, and we weighted and sampled the set to reduce the number of registration points, so there were more computations but the algorithm ran much quicker. Furthermore, this method was more accurate, as shown in Table 1.

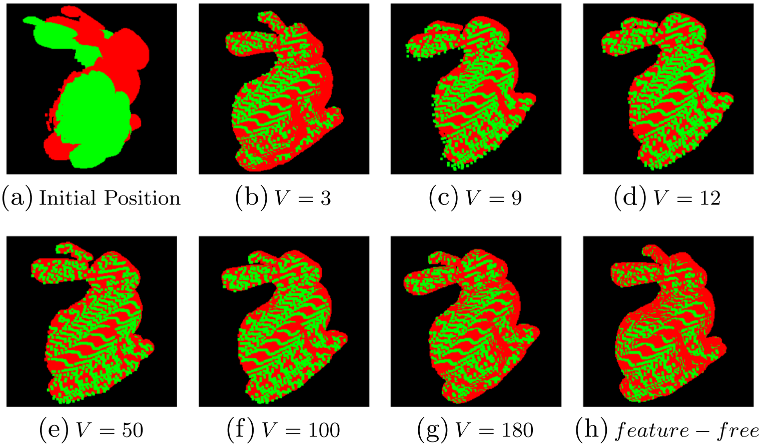
5.2 Noise and outliers

This experiment tested the performance of our algorithm in the presence of noise and outliers. The ICP algorithm is based on a least-squares fit, so it is sensitive to noise and outliers. In this experiment, we used the Stanford standard bunny scan data and the terracotta warrior model shown in Fig. 3, which contain noise. We tested the running time and matching errors of our algorithm.

Following the algorithm in Section 4, we set V_f using different feature points. We let $V_f = [3, 9, 12, 16, 50, 100, 180]$ denote the set of different feature points, and then

Table 1 Performance comparison for different algorithms

Algorithm	Matching error (d/mm)	Run time(s)
Traditional initial registration	7.01324	89.43
Classical ICP algorithm	3.40153	258.57
Initial registration + classical ICP	1.20365	42.14
Proposed algorithm	0.01248	38.46



(a) Initial Position (b) $V = 3$ (c) $V = 9$ (d) $V = 12$

(e) $V = 50$ (f) $V = 100$ (g) $V = 180$ (h) $feature - free$

Fig. 5 Comparison of the results for different numbers of corresponding points

used the corresponding matching points to calculate the transformation. We applied this transformation to the registration process.

We registered the point clouds of the bunny and terracotta warrior that had the same points as those obtained by the same sampling method under various conditions. Table 2 shows the robustness of the registration algorithm to the number of feature points. According to Table 2, three matches were sufficient to calculate the optimal transformation. An increase in the number of point pairs corresponded to a small reduction in the matching error, but an increase in time. Figure 5 shows the registration results using the standard point clouds. Figure 5a shows the initial orientation of the point cloud and Fig. 5b shows the orientation after registration using 3 matching point pairs. Although the point cloud data was registered, there were errors and the point clouds did not completely overlap. Figures 5c–g are the results obtained using 9, 12, 50, 100, and 180 matching point pairs. Figure 5 is the registration result that did not use extracted feature points. We can compare these results to evaluate the performance of the algorithm. Table 3 shows the performance of the registration algorithm in the presence of noise, which we can compare with the results in Table 2. The increase in corresponding point pairs improved the robustness of the algorithm. However, the point pair selection was affected by the noise. Figure 6 shows the registration results in the presence of noise. Comparing Fig. 6b and Fig. 5b, we can see some obvious registration

Table 2 Performance comparison when registering the bunny data using different numbers of corresponding points. The best results are highlighted

Number of matching point pairs	Matching error (d/mm)	Runtime (s)
3	0.01347	36.23
9	0.01248	38.46
12	0.01242	40.01
16	0.01143	42.56
50	0.01041	67.31
100	0.00913	70.18
180	0.00911	90.54

Table 3 Performance comparison when registering point clouds of the terracotta warrior data, which contain noise. The best results are highlighted

Number of matching point pairs	Matching error (d/mm)	Run time (s)
3	0.05448	40.31
9	0.05132	46.23
12	0.05042	49.05
16	0.04943	53.41
50	0.04041	83.21
100	0.04013	91.23
180	0.04001	113.53

errors. Figures 6c, d were obtained using 9 and 12 matching point pairs. Compared with Fig. 5c, d, we can see that an increase in point pairs reduced the error, although it was still present. Figures 6e–g are not obviously different to Figs. 6c, d. Thus, increasing the number of feature point pairs improved the algorithms resistance to noise, and the registration error gradually leveled off.

Selecting V_f The criteria for choosing V_f were determined using multiple experiments, because we could not derive all the parameters mathematically. Our experimental results show how the performance of our algorithm varies with Z, which represents a gradual trade-off between performance and robustness. Z should be set based on the scene. Figure 7 lists the various performance characteristics of the algorithm. For comparison, we used point

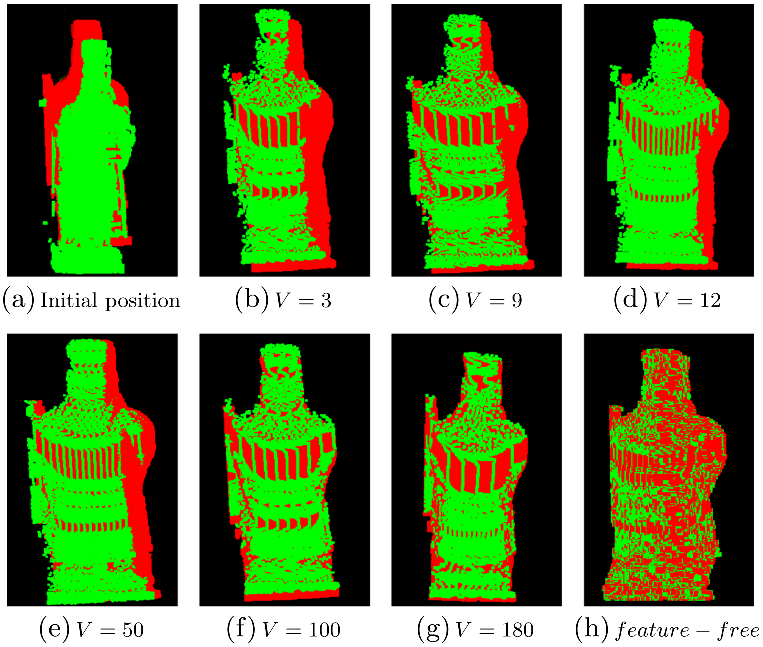
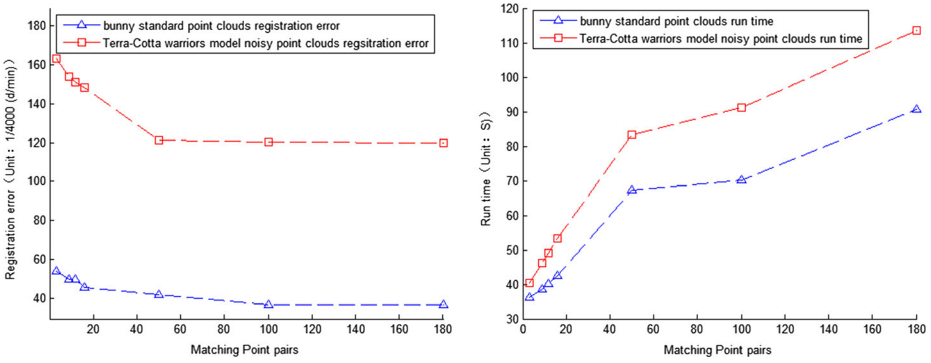
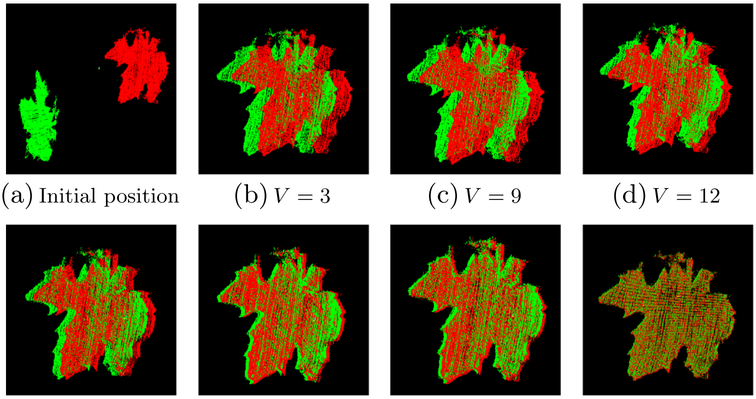

(a) Initial position (b) $V = 3$ (c) $V = 9$ (d) $V = 12$
(e) $V = 50$ (f) $V = 100$ (g) $V = 180$ (h) $feature - free$

Fig. 6 Results for different numbers of corresponding points, using noisy point clouds



(a) Registration error (b) Time

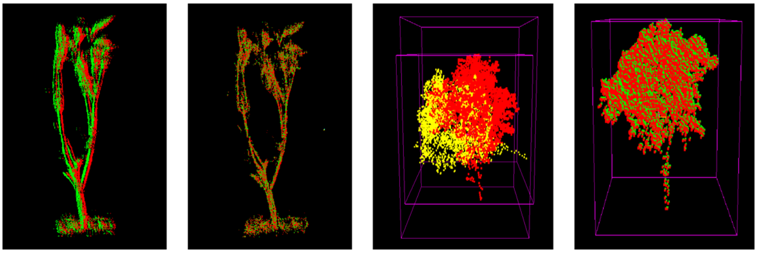
Fig. 7 Performance of our algorithm for different noise levels, and for different numbers of feature points



(a) Initial position (b) $V = 3$ (c) $V = 9$ (d) $V = 12$

(e) $V = 50$ (f) $V = 100$ (g) $V = 180$ (h) $feature - free$

Fig. 8 Maple leaf registration using the proposed method



(a) Initial Position (b) $V = 12$ (c) Initial position (d) $V = 9$

Fig. 9 Registration of tree point cloud data

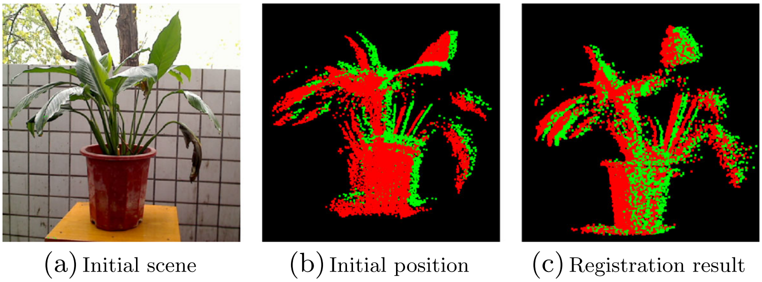

(a) Initial scene (b) Initial position (c) Registration result

Fig. 10 Registration of plant data

clouds with and without noise, and different numbers of matching point pairs. An increase in the number of matching point pairs slowed down the feature extraction process, and the convergence speed was affected by noise. However, the matching error gradually decreased and then stabilized.

5.3 Application

Registering the maple leaf data First, we registered the leaf point clouds acquired using our equipment in an outdoor environment [31]. There were 49251 points. We then applied the same method as in Section 5.2. Figure 8 shows that our algorithm produced a near perfect registration, even for noisy point clouds and few feature points. For large, dense datasets, a small fraction of uniformly sampled points are sufficient to compute the registration; the full dataset is only used for the transformation. Our approach is robust to uniform sampling, and residual matching point pairs are easily removed after a few ICP iterations.

Registering the tree model data We applied our method to two point-clouds of a tree model, one acquired using our equipment in an outdoor environment (Figs. 9a, b), and one provided by the University of Queensland (Figs. 9c, d). In this experiment, we set $V_f = 12$ and $V_f = 9$.

Registering broad leaved plant data Finally, we directly registered broad leaved plant point cloud data that contained 116397 points, which were obtained by our scanner in an outdoor environment. The data were captured in an uncontrolled environment, so they contained some noise and outliers. In this experiment, we set $V_f = 50$. The algorithm took approximately 92.51 s. The registration results are shown in Fig. 10.

6 Conclusions

We proposed a new point clouds registration method for large-scale, 3D point clouds, which is based on a geometrical feature neighborhood constraint. The key point set is selected to reduce the search space. We use the similarities of features to optimize the matching point pairs, and angle features to remove incorrect registrations during the exact registration process.

Our experiments demonstrate that the proposed method is an order of magnitude faster than existing registration techniques, and produces a more accurate registration. Although noise has some effect, it did not affect the final performance of the algorithm. Our method is robust to noise because it uses a geometrical feature neighborhood constraint.

We effectively integrated the proposed method with our scanner, which works in an outdoor environment. Our technique can also be used when it is difficult to extract feature points. In the future, we will investigate methods for computing adaptive thresholds, and provide a reference for virtual plant reconstruction.

Acknowledgments This work was partially supported by the National High Technology Research and Development Program of China (863 Program) (No. 2013AA102304), Science and Technology Innovation Project (QN2013056), and Science and Technology Innovation Project (2014YB067).

References

1. Basdogan C, Oztireli AC (2008) A new feature-based method for robust and efficient rigid-body registration of overlapping point clouds. Vis Comput 24(7-9):679–688
2. Besl PJ, McKay ND (1992) Method for registration of 3-D shapes. In: Robotics-DL tentative (pp. 586-606). International Society for Optics and Photonics
3. Chen Y, Medioni G (1991) Object modeling by registration of multiple range images. In: Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on (pp. 2724-2729). IEEE
4. Cheng L, Tong L, Li M, Liu Y (2013) Semi-automatic registration of airborne and terrestrial laser scanning data using building corner matching with boundaries as reliability check. Remote Sens 5(12):6260–6283
5. Cheng ZQ, Wang YZ, Li B, Xu K, Dang G, Jin SY (2008) A Survey of Methods for Moving Least Squares Surfaces. In: Volume Graphics, pp 9–23
6. Dai J, Yang J (2011) A novel two-stage algorithm for accurate registration of 3-D point clouds. In: Multimedia Technology (ICMT), 2011 International Conference on (pp. 6187-6191). IEEE
7. Daniels Ii J, Ochotta T, Ha LK, Silva CT (2008) Spline-based feature curves from point-sampled geometry. Vis Comput 24(6):449–462
8. Diez Y, Mart J, Salvi J (2012) Hierarchical normal space sampling to speed up point cloud coarse matching. Pattern Recogn Lett 33(16):2127–2133
9. Dos Santos DR, Dal Poz AP, Khoshelham K (2013) Indirect georeferencing of terrestrial laser scanning data using control lines. Photogramm Rec 28(143):276–292
10. Du S, Zheng N, Meng G, Yuan Z (2008) Affine registration of point sets using ICP and ICA. Signal Processing Letters. IEEE 15:689–692
11. Du S, Zheng N, Ying S, Liu J (2010) Affine iterative closest point algorithm for point set registration. Pattern Recogn Lett 31(9):791–799
12. Guan Y, Cheng X, Shi G (2008) A robust method for fitting a plane to point clouds [J]. J Tongji Univ (Nat Sci) 7:024
13. Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W (1992) Surface reconstruction from unorganized points (Vol. 26, No. 2, pp. 71-78). ACM
14. Huang T, Zhang D, Li G, Jiang M (2012) Registration method for terrestrial LiDAR point clouds using geometric features. Opt Eng 51(2):021114-1
15. Jiang J, Cheng J, Chen X (2009) Registration for 3-D point cloud using angular-invariant feature. Neurocomputing 72(16):3839–3844
16. Jing LD, Zhi YC, Xiu ZY (2007) The application of ICP algorithm in point cloud alignment. J Image Graph 12(3):517–521
17. Liao Y, Xu F, Zhao X, Hagiwara I (2014) A Point Cloud Registration Method Based on Point Cloud Region and Application Samples. In: AsiaSim 2014 (pp. 216-227). Springer, Berlin Heidelberg
18. Lou B, Zhong Y, Li R (2004) Data registration in3D scanning systems. J Tsinghua Univ (Sci Techn) 44(8):1104–1106
19. Mian AS, Bennamoun M, Owens RA (2006) A novel representation and feature matching algorithm for automatic pairwise registration of range images. Int J Comput Vis 66(1):19–40
20. Rusinkiewicz S, Levoy M (2001) Efficient variants of the ICP algorithm. In: 3-D Digital Imaging and Modeling, 2001 Proceedings. Third International Conference on (pp. 145-152). IEEE

21. Schindhelm CK (2012) Evaluating slam approaches for microsoft kinect. In: Proceedings 2011 The Eighth International Conference on Wireless and Mobile Communications (ICWMC 2012), Venice, pp 402–407
22. The standford 3D Scanning Repository (2003-04-01), Standford University Computer GraphicsLaboratory. http://www.graphics.stanford.edu/data/3Dscanrep/
23. Wei X, Jiexin P (2011) Point cloud integration base on distances between points and their neighborhood centroids. J Image Graph 5:886–891
24. Weinmann M, Jutzi B (2011) Fully automatic image-based registration of unorganized TLS data. Int Arch Photogramm Remote Sens Spat Inf Sci 38:1–6
25. Wu M, Zhou LS, Wang ZD, An LL (2003) Research of multi-view registration and integration on measured point cloud data. J Nanjing Univ Aeronaut Astronaut 35(5):552–557
26. Xie Z, Xu S, Li X (2010) A high-accuracy method for fine registration of overlapping point clouds. Image Vis Comput 28(4):563–570
27. Yang HJ, He DJ (2012) A simplified and accurate registration for splat-based fruit scans. ICIC Express Lett Part B: Appl 3(4):733–741
28. Yang J, Li H, Jia Y (2013) Go-icp: Solving 3d registration efficiently and globally optimally. In: Computer Vision (ICCV), 2013 IEEE International Conference on (pp. 1457-1464). IEEE
29. Yanjuan Z (2006) Registration of scattered cloud data. J Comput Aided Des Comput Graph 18(4):475
30. Zhang X, Xi J, Yan J (2005) Research on digital measurement technology based on point cloud data of complex surfaces. Comput Integr Manuf Syst-beijing 11(5):727
31. Zhang Z, Yuan L (2012) Building a 3D scanner system based on monocular vision. Appl Opt 51(11):1638–1644
32. Zhu J, Du S, Yuan Z, Liu Y, Ma L (2012) Robust affine iterative closest point algorithm with bidirectional distance. IET Comput Vis 6(3):252–261

Nan Geng ShaanXi Province, China. Birthdate: Apr, 1971. He is Electronic and Mechanical Engineering Ph.D., graduated from College of Mechanical and Electronic Engineering, Graduate School of Northwest A&F University, China. And research interests on Image analysis and Machine vision, Computer virtual technology and Graphics, Agricultural Informatization. He is a professor of College of Information Engineering, Northwest A&F University and Director of the institute of graphic image at ShaanXi province.

Fufeng Ma ShanDong Province, China. Birthdate: Apr. 1989. He is graduate student from College of Information Engineering of Northwest A&F University, China. And research interests on Computer virtual technology and Graphics.

Huijun Yang (1974-), born in Wanrong, Shanxi, China. Femal, Ph.D., Associate Professor. Research field: computer graphics, point cloud reconstruction and three-dimensional modelling. She has published over 18 academic papers in the significant journal and preside 3 projects, published 2 textbooks as the paternal, participated 6 projects as the key researcher and obtained 4 software copyright. College of Information Engineering, Northwest A&F University.

Boyang Li ShaanXi Province, China. Birthdate: July. 1990. She is graduate student from College of Information Engineering of Northwest A&F University, China. And research interests on Image analysis and Machine vision.

Zhiyi Zhang ShanXi Province, China. Birthdate: Sep, 1974. He is Electronic and Information Engineering Ph.D., graduated from Faculty of Engineering, Graduate School of Engineering Iwate University, Japan. And research interests on Computer Aided Design, Computer Aided Geometric Design, Computer Graphics. He is a associate professor of College of Information Engineering, Northwest A&F University.