



Low Overlapping Plant Point Cloud Registration and Splicing Method Based on FPFH

Shengdong Lin, Yeping Peng^(✉), and Guangzhong Cao

Guangdong Key Laboratory of Electromagnetic Control and Intelligent Robots, College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518060, China
pyp8020@163.com

Abstract. Three-dimensional point clouds captured by sensors can quantify plant phenotype, which plays an important role in agricultural intelligence. Many scanned objects in agriculture and forestry are tall and obscured by leaves, so point clouds captured by either terrestrial or airborne methods may be incomplete. In order to obtain a more complete point cloud, this paper proposes a point cloud registration method based on the fast point feature histogram (FPFH), which aligns point clouds collected from different viewpoints. This method calculates the FPFH feature of each point and the Bhattacharyya distance between point pairs. Effective strategies pick out reliable sets of point pairs. Singular value decomposition is used to obtain the transformation relationship between point clouds. Experimental results show that the proposed method has high accuracy for plant point cloud registration in real scenes, and the root-mean-square error is smaller than that of common registration methods of SAC-IA and NDT.

Keywords: Fast point feature histogram · Bhattacharyya distance · Random sample consensus · Point cloud registration

1 Introduction

Plant phenotype is an objective expression of plant growth and plays an important role in plant research and agricultural production. In traditional agricultural and forestry research, plant phenotypic analysis relies on manual observation and measurement of plants. This approach is time-consuming, labor-intensive, and relies heavily on subjective experience. The integration of modern information technology and agriculture has promoted the formation of new production modes such as agricultural information intelligent perception, precise monitoring, and quantitative decision-making, which has promoted the development of precision agriculture [1]. Using modern information technology to obtain the phenotype of plants and crops without damage and quantify the growth state of crops is one of the important steps to achieve precision agriculture [2].

With the wide application of computer vision in the field of agriculture and forestry, plant disease detection [3] and fruit maturity judgment [4] can be achieved by using two-dimensional (2D) images to obtain plant information. Since the three-dimensional (3D) model has more information than the 2D image, it is a potential direction to use

the 3D point cloud model to monitor and manage agricultural production [5]. Wu et al. [6] used point clouds to generate the skeleton of maize plants, and estimated phenotypes such as plant height, leaf length, leaf inclination, and azimuth. Rueda-ayala et al. [7] estimated the plant height, biomass, and volume of sweet wheat grass and ryegrass through the point cloud of grassland. Cabo et al. [8] automatically identified tree stems by analyzing point clouds, and then estimated tree height and trunk diameter. The plant surface morphological information obtained through point clouds has many uses, such as judging the growth state of plants, predicting crop yield, calculating volume, and so on.

At present, most of the methods for the 3D reconstruction of plants are based on optical non-contact sensors. Wang et al. [9] used the terrestrial laser scanning (TLS) method to obtain the point cloud of corn plants, which is fast and simple to operate, but the scanning equipment is expensive. Vázquez-Arellano et al. [10] used a time-of-flight (TOF) camera to sequentially collect point clouds of corn seedlings in the experimental field for registration, and the registration algorithm used iterative closest point (ICP). The point cloud registration method in this paper relies on point clouds whose initial poses are roughly aligned and overlapped, and the price of the capture device is high. Chen et al. [11] used an RGB camera to take multiple photos of kale, wheat, and physalis placed on a turntable, calculated camera internal parameters and pose information according to incremental SFM, used MVSNet to generate a depth map, and finally generated 3D points cloud. The point cloud generated by this method is dense, but the method using the turntable is only suitable for low plants and not suitable for outdoor scenes. Ni et al. [12] used a binocular camera to capture two images of stereo plants, used the efficient large scale stereo matching (ELAS) algorithm to calculate the disparity map, and then obtained the 3D point cloud model of the plant through triangulation. When processing images were taken at close range, the texture of leaves can be clearly displayed, but the reconstruction effect is poor in the case of long-distance. Guan et al. [13] developed an imaging system consisting of an RGB camera and a PMD camera. Using the DBSCAN algorithm, the point cloud of the soybean plant canopy is extracted from the original single-view point cloud, and the point cloud models based on the side view and the top view are generated. But they are not spliced together, so the phenotypic information is not comprehensive enough.

The methods in the above-mentioned articles are almost all applied to obtain 3D point cloud models of low plants. However, for tall plants and trees, due to the limited field of view of the sensor, the above method cannot obtain a high-integrity point cloud model from top, bottom, and side views. To solve this problem, point clouds captured from multiple angles and efficient point cloud registration algorithms are required.

Among the point cloud registration algorithms, the most classic is the iterative closest point (ICP) proposed by Besl et al. [14]. In this method, the point pair with the closest Euclidean distance is used as the matching point, and the transformation matrix between the two point clouds is calculated by the least square method. Then, the matching points are repeatedly selected and calculated iteratively until the error is less than the threshold, and the transformation matrix is regarded as optimal. However, the disadvantage of this method is that it requires an initial value and is sensitive to noise, and it is easy to encounter local optimization [15].

In order to improve the accuracy of registration, many scholars divide the registration into two steps, that is, first use geometric features to obtain rough transformation, and then use ICP for fine registration. Accurately computing transformation matrices relies on accurate descriptions and correct matching of features [16]. At present, representative local feature algorithms include 3D shape context (3DSC) proposed by Frome et al. [17], the signature of histograms of orientations (SHOT) proposed by Salti et al. [18], point feature histogram (PFH) [19], and fast point feature histogram (FPFH) [20] proposed by Rusu et al.

FPFH is a 3D feature descriptor based on PFH. It uses the normal vector to calculate the relationship between the query point and the points in its neighborhood. It represents the features of the point neighborhood in the form of a histogram. Since FPFH has a fast operation speed, only 33 dimensions, and occupies less operation space, this paper proposes a point cloud registration algorithm based on FPFH features. This method is used to register point clouds from different viewpoints to obtain a more complete point cloud model. The method is applied to the multi-source real plant point cloud with a low overlap rate, and a more complete three-dimensional point cloud model of the plant can be obtained, and the error is at the millimeter level. This method is of great significance to agricultural intelligence. The remainder of the paper is organized as follows. The second section will introduce the detailed method and mathematical model. The third section contains experimental verification and a discussion of the results. Finally, the conclusion of this paper is given in the fourth section.

2 Methodology

The purpose of registration is to unify the two point clouds into the same coordinate system. The two point clouds to be registered are marked as a source point cloud and a target point cloud, respectively. This paper proposes a point cloud registration method based on FPFH: (1) Preprocess the two point clouds, including subsampling the point cloud with a voxel grid and statistical filtering to remove outliers; (2) Estimate each point cloud normals of points and compute their FPFH features. Histogram similarity is evaluated according to Bhattacharyya distance. Iteratively compares the FPFH features in the source point cloud and the target point cloud to obtain an initial set of matching point pairs; (3) Improve the accuracy of matching point pairs. For the special case where multiple points match one point, only the corresponding point with the closest distance is selected. Then, sort the remaining matching point pairs and pick the top matching point pairs. At the same time, the distance threshold is used to filter the points that are too concentrated; (4) The random sample consensus (RANSAC) is used to remove the wrong matching point pairs, and the calculation of singular value decomposition (SVD) is performed on the remaining matching point pairs to obtain the rotation matrix and translation vector. The specific implementation of this process is shown in Fig. 1.

2.1 Point Normal Estimation

Since a single point cannot reflect the geometric surface features around it, the features of a point need to be considered by combining the neighboring points within a certain range

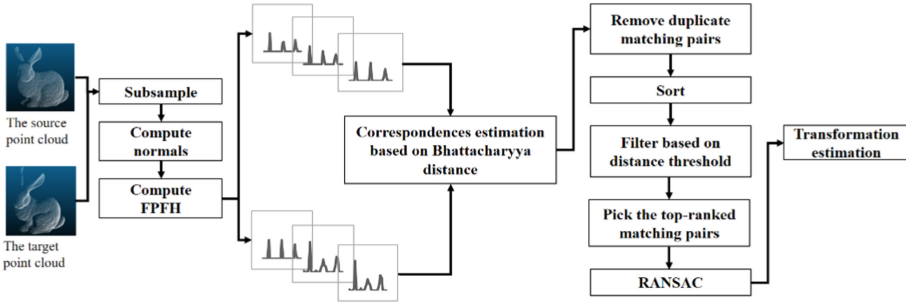


Fig. 1. The pipeline of pairwise point clouds registration.

around it. Typically, geometric surface features are represented based on the normals of points within a neighborhood.

The method of calculating the normal vector of the point cloud surface in this paper is to estimate the normal vector for the points in the neighborhood. Converts the problem of solving the normals of a point on the surface of the point cloud into the problem of estimating the normals of the tangent planes. By minimizing the objective function containing the normal vector, using the idea of the least-squares method, the result of the dot product of the vector formed by the point and each of its neighbors and the normal vector is zero.

First, the plane is represented as a point x and a normal vector \vec{n} , as shown in Fig. 2. The points in the neighborhood are represented by $p_i \in P^n$.

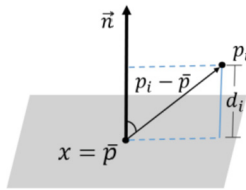


Fig. 2. Estimate the normal vector of a point

Calculate the centroid \bar{p} of the neighborhood points in the given radius r_n and assign it to x , which can be shown as

$$x = \bar{p} = \frac{1}{n} \sum_{i=1}^n p_i. \tag{1}$$

The vector from the centroid to the neighborhood point is defined as

$$y_i = p_i - \bar{p}. \tag{2}$$

The distance d_i from the neighboring point p_i to the plane is equal to the projection of $(p_i - x)$ onto the normal vector \vec{n} , which is expressed as

$$d_i = (p_i - x) \cdot \vec{n}. \tag{3}$$

The least-square plane estimation problem is constructed. To find a plane passing through the centroid \bar{p} with a normal vector \vec{n} , the objective function is

$$\min_{x, \vec{n}, \|\vec{n}\|=1} \sum_{i=1}^n \left[(p_i - x)^T \vec{n} \right]^2. \quad (4)$$

Substituting the centroid \bar{p} into the above formula can be simplified to

$$\min_{\vec{n}, \|\vec{n}\|=1} \vec{n}^T (YY^T) \vec{n}. \quad (5)$$

where $YY^T = \sum_{i=1}^n y_i y_i^T$ is the 3×3 covariance matrix, represented by C . And by the definition of y_i , there are

$$C = YY^T = \sum_{i=1}^n y_i y_i^T = \sum_{i=1}^n (p_i - \bar{p})(p_i - \bar{p})^T. \quad (6)$$

The relationship between the eigenvalue λ_e and the eigenvector \vec{v}_e of the covariance matrix is

$$C \vec{v}_e = \lambda_e \vec{v}_e, e \in \{0, 1, 2\}. \quad (7)$$

The eigenvalue λ_e and its corresponding eigenvector \vec{v}_e are obtained by singular value decomposition. If $0 \leq \lambda_0 \leq \lambda_1 \leq \lambda_2$, according to the principle of principal component analysis (PCA), the eigenvector \vec{v}_0 corresponding to λ_0 is approximately in the direction of the normal vector \vec{n} or $-\vec{n}$.

The ambiguity of normal vector cannot be solved mathematically. Therefore, we set a viewpoint V_p as the judgment basis to make the standard selection of normal direction consistent [21]. All normal directions should conform to the judgment formula, expressed as

$$\vec{n} \cdot (V_p - p_i) > 0. \quad (8)$$

2.2 The Calculation of FPFH

Both PFH and FPFH construct multi-dimensional histograms to describe point feature information. FPFH is the optimization algorithm of PFH. The main differences are the selection method, the calculation of point features, and the dimension of the histogram. The neighborhood influence range of PFH and FPFH is shown in Fig. 3. When calculating PFH, all of p_q 's neighbors enclosed in the sphere with a given radius r_p are selected and the relationship between the pairwise is calculated. The calculation of FPFH is to first calculate the relationship between point p_q and its neighborhood points within a given radius r_f . And then compute the relationship between these neighborhood points and their neighborhood points. Therefore, the computational complexity of FPFH is lower than that of PFH [20].

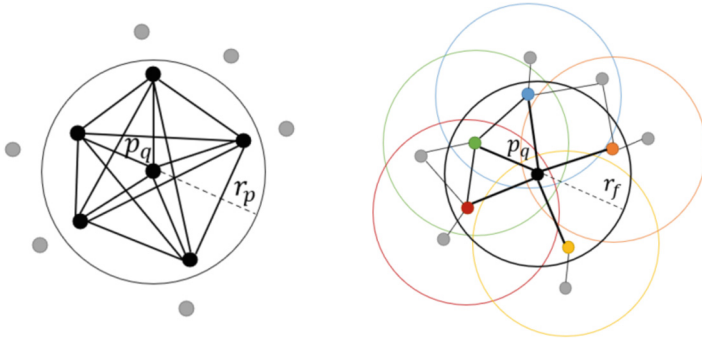


Fig. 3. The influence region of PFH (left) and FPFH (right)

The calculation of FPFH first defines the space within the given radius r_f as the neighborhood of the query point p_q . The query point p_q is then paired with the points in the neighborhood. In order to represent the relationship between each point pair in the local coordinate system, Darboux coordinate frame corresponding to the point pair is established, as shown in Fig. 4.

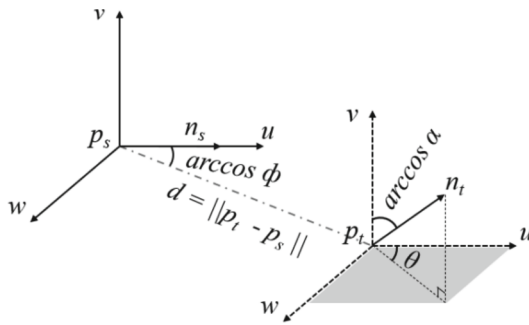


Fig. 4. Darboux coordinate frame

Specifically, Darboux coordinate frame $u - v - w$ is defined as

$$\begin{cases} u = n_s \\ v = \frac{p_t - p_s}{\|p_t - p_s\|} \times u \\ w = u \times v, \end{cases} \quad (9)$$

where p_s and p_t are the coordinates of source point and target point, and n_s and n_t are their estimated normals respectively; $p_t - p_s$ is the vector between the source point and target point. The direction of the normal n_s is defined as the direction of the u axis. The direction of the v axis is the direction of the cross product of $p_t - p_s$ and u . The w axis is the cross product of the u - and v - axes. Therefore, $u - v - w$ is a cartesian coordinate system with three perpendicular axes.

Four features describing the relationship between two points are defined, which reduces the number of values representing the two points and their normals from 12 to 4. They are expressed as

$$\begin{cases} \alpha = v \cdot n_t \\ \phi = u \cdot \frac{p_t - p_s}{d} \\ \theta = \arctan(w \cdot n_t, u \cdot n_t) \\ d = \|p_t - p_s\|, \end{cases} \quad (10)$$

where α is the dot product of the normal n_t and the v axis, ϕ is the dot product between the u axis and the normalized $p_t - p_s$, θ is the angle between the projection of the target normal n_t on the $u - v$ plane and the u axis, d is the Euclidean distance from p_s to p_t . In some cases, the fourth feature, d , is not very important when the distance between adjacent points increases from the viewpoint. Therefore, d is omitted from the calculation of FPFH [20]. These three angular features, $\{\alpha, \phi, \theta\}$, adopted by FPFH is spread over as much as possible of the available histogram range without exhibiting a bias for certain regions [22].

The calculation of FPFH of query point p_q is divided into the following steps: (1) Calculate the α, ϕ, θ of query point p_q , denoted as $SPFH(p_q)$; (2) Calculate the α, ϕ, θ of K neighborhood points $p_{qi} (i = 1, 2, \dots, K)$, denoted as $SPFH(p_{qi})$; (3) Each of the three angular features has 11 dimensions, so a histogram with a horizontal coordinate of 33 dimensions can be obtained. The $SPFH(p_q)$ and the $SPFH(p_{qi})$ with weights are counted into the histogram to obtain the $FPFH(p_q)$, which is expressed as

$$FPFH(p_q) = SPFH(p_q) + \frac{1}{K} \sum_{i=1}^K \frac{1}{\omega_i} SPFH(p_{qi}). \quad (11)$$

where K is the number of neighborhood points of p_q ; ω_i is the weight value inversely proportional to the Euclidean distance between p_q and p_{qi} , which means that the farther the distance between p_{qi} and p_q , the less influence p_{qi} has on the $FPFH(p_q)$.

2.3 Find the Matching Point Pairs

Sample Consensus Initial Alignment (SAC-IA) is a classical and common point cloud registration algorithm that calculates transformation matrix according to local descriptors [20]. The algorithm selects multiple sampling points from the source point cloud, finds one or more similar points in the target point cloud, and then randomly selects a point from the similar points as the corresponding point. According to the different selected point pairs, the transformation matrix is calculated, and an error metric is computed to evaluate the transformation. Finally, a good transformation is found among all the transformations. However, due to the method of selecting matching pairs and the standard of evaluating similarity, the stability of registration is poor in the case of low overlap rate and high noise, and the wrong results may occur. Therefore, the method proposed discards SAC-IA.

In statistics, there are metrics for evaluating the correlation of probability distribution functions, such as Bhattacharyya distance, Hellinger distance, Kullback-Leibler divergence, correlation. Several comparative experiments of these metrics are presented in

Sect. 3. Because of the symmetry and lower computational complexity of Bhattacharyya distance, the method proposed uses Bhattacharyya distance as the metric to evaluate FPFH similarity.

For discrete probability distributions p and q over the same domain X , Bhattacharyya distance $D_B(p, q)$ is defined as

$$D_B(p, q) = -\ln(BC(p, q)), \quad (12)$$

where

$$BC(p, q) = \sum_{x \in X} \sqrt{p(x)q(x)} \quad (13)$$

The FPFH histogram is essentially a combination of three 11-dimensional histograms. Therefore, when comparing the similarity of FPFH, the similarity of the three histograms is compared respectively. The three Bhattacharyya distances are calculated separately, and their sum is used to evaluate the similarity between the two FPFH. The FPFH similarity of the two points p_1 and p_2 is calculated as

$$D_{B-FPFH}(p_1, p_2) = D_{B-\alpha}(p_1, p_2) + D_{B-\phi}(p_1, p_2) + D_{B-\theta}(p_1, p_2). \quad (14)$$

Since the range of $BC(p, q)$ is $[0, 1]$, the range of $D_B(p, q)$ is $[0, +\infty)$, and the range of $D_{B-FPFH}(p_1, p_2)$ is also $[0, +\infty)$. The closer the Bhattacharyya distance is to zero, the more similar the histograms of the two points are, which means the more similar the neighborhood features of the two points are.

The points in the source point cloud respectively find the point with the smallest Bhattacharyya distance from the target point cloud as the matching point pair. These match point pairs form the initial set of matched point pairs.

2.4 Strategies to Improve the Accuracy of Matching Pairs

In order to filter some matching point pairs and improve the estimation accuracy of the transformation matrix, some strategies are added to our algorithm, as shown in Algorithm 1.

In the set of matching point pairs, it is inevitable that multiple points will choose the same point as their most similar point. When this happens, the point that has been selected multiple times is reversed to select its most similar point. The reconstituted match pair is then put into the collection, while the remaining point pairs are deleted.

Then, the matching point pairs are sorted according to distance, and the top matching point pairs are selected. At the same time, the distribution range of selected points is enlarged by setting a distance threshold to eliminate the points whose distribution is too concentrated. Finally, RANSAC is used to weed out incorrect matching pairs. The remaining set of matching point pairs is used for subsequent calculations.

Algorithm 1 Iterative selection of correct matching pairs

Input: S_0 : the initial matching pair set; N : the number of matching pairs to be selected; t_e : the Euclidean distance threshold;

Output: S_{acc} : the accurate matching pair set;

- 1: Initialization: $S_0 \leftarrow \emptyset$; $S_{acc} \leftarrow \emptyset$; $n = 1$;
- 2: **for** $i = 1 : size(S_0)$ **do**
- 3: **if** the point p_t in the target point cloud is selected multiple times, **then**
- 4: point p_t reversely selects its most similar point as the matching pair;
- 5: delete other duplicate matching pair(s) of point p_t ;
- 6: **end if**
- 7: **end for**
- 8: Sort the matching pairs in S_0 ;
- 9: Put the first matching pair p_1 into S_1 ;
- 10: Put the source point of the first matching pair to p_{tc} ;
- 11: **for** $j = 1 : size(S_0)$ **do**
- 12: **if** $D_{Euclidean}(p_{tc}, p_{tj}) > t_e$ **then**
- 13: $S_1 = S_1 \cup p_j$;
- 14: $p_{tc} = p_{tj}$;
- 15: **end if**
- 16: **if** $n = N$ **then**
- 17: break;
- 18: **end if**
- 19: **end for**
- 20: Apply RANSAC to S_1 , get the inlier set S_{acc} .

2.5 Calculation of Transformation

According to the matching point pairs, the singular value decomposition method is used to calculate the transformation between the two point clouds. The matching point pair set after filtering is expressed as

$$P = \{p_1, \dots, p_m\}, P' = \{p'_1, \dots, p'_m\}. \quad (15)$$

All point pairs conform to

$$p_k = Rp'_k + t. \quad (16)$$

where R is the rotation matrix and t is the translation vector. The error e_k is constructed as

$$e_k = p_k - (Rp'_k + t). \quad (17)$$

To obtain R , t , it is necessary to minimize the sum of error squares, which can be expressed as

$$E^2 = \sum_{k=1}^m \|(p_k - (Rp'_k + t))\|^2. \quad (18)$$

According to the least-squares fitting of the two point sets P and P' , (18) can be simplified as

$$E^2 = \sum_{k=1}^m \|q_k - Rq'_k\|^2, \quad (19)$$

$$t = p - Rp', \quad (20)$$

where p and p' are the centroids of the two point sets P and P' , respectively; q_k and q'_k

$$q_k = p_k - p, q'_k = p'_k - p'. \quad (21)$$

Singular value decomposition is used to estimate R . Finally, t is solved by substituting R into Eq. (20).

3 Experiment and Analysis

The experimental data include the public point cloud data and the plant point cloud data captured in the real scene. The experiments were implemented on a computer with 8 GB RAM and an Intel Core i5-10500 CPU. The program was developed based on C++ and a third-party open-source library, Point Cloud Library (PCL), which contains many point cloud basic processing functions.

3.1 Registration Experiment of Public Dataset

In order to verify the accuracy and validity of the method using Bhattacharyya distance, the methods using the similarity criteria of Hellinger distance, Kullback-Leibler divergence, and correlation were compared. The registration results of the public point cloud models from different perspectives are shown in Figs. 5, 6 and 7. The initial input point cloud models are selected from the dataset of Stanford 3D scanning repository.

It can be seen that these methods can generally register point clouds from different perspectives. In comparison, the proposed method has better alignment in details without obvious double shadow or offset. In the registration of Bunny 0° and 45° models, except for the proposed method, the alignment at the ear region is poor. In the registration of Bunny 45° and 90° models, the results of the registration method based on correlation show an overall deviation. In the registration of Armadillo 270° and 300° models, the results of the methods based on Hellinger distance, Kullback-Leibler divergence and correlation do not fit well in the claw and head of the model.

The registration error of each method is then quantitatively evaluated by the root mean square error (RMSE) criterion, as shown in Eq. (22).

$$RMSE = \sqrt{\frac{1}{M} \sum_{l=1}^M \|Rp_l + t - q_l\|^2} \quad (22)$$

where R and t are the true values of the rotation matrix and translation vector respectively, p_l and q_l are the points of the initial point cloud and the point cloud transformed by the registration algorithm respectively, and M is the number of points in the point cloud.

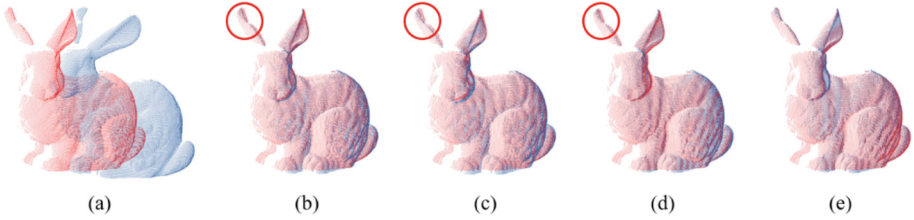


Fig. 5. Initial input of Bunny 0° and 45° and registration results using methods based on different metrics. (a) Bunny 0° and 45° ; (b) Hellinger distance; (c) Kullback-Leibler divergence; (d) Correlation; (e) The proposed method.

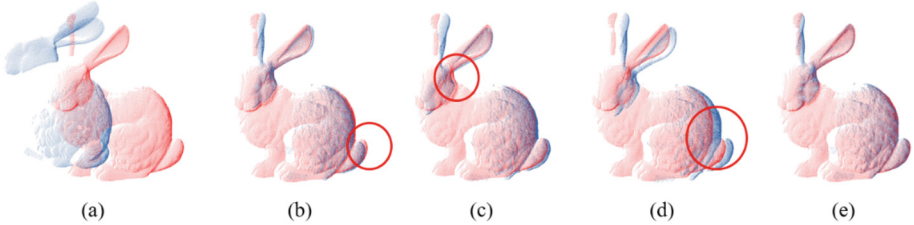


Fig. 6. Initial input of Bunny 45° and 90° and registration results using methods based on different metrics. (a) Bunny 45° and 90° ; (b) Hellinger distance; (c) Kullback-Leibler divergence; (d) Correlation; (e) The proposed method.

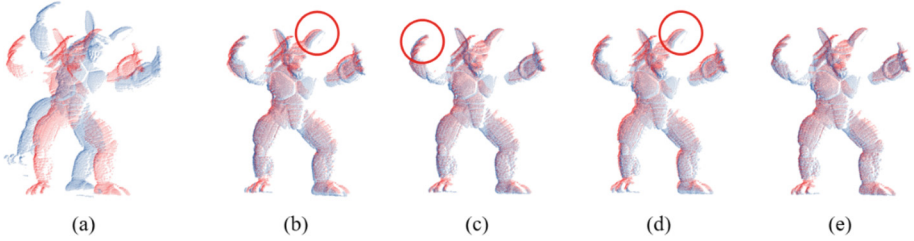


Fig. 7. Initial input of Armadillo 270° and 300° and registration results using methods based on different metrics. (a) Armadillo 270° and 300° ; (b) Hellinger distance; (c) Kullback-Leibler divergence; (d) Correlation; (e) The proposed method.

The RMSEs of each experiment are listed in Table 1. It can be seen that the RMSE of the method proposed is lower than that of other methods. The results are consistent with those in Figs. 5, 6 and 7, which illustrates that the proposed method is more accurate and effective when applied to the classic data set.

3.2 Registration Experiment of Noisy Plant Point Cloud

In order to verify the accuracy and anti-noise performance of the proposed method in registering low overlap plant point clouds, the proposed method was compared with typical point cloud registration algorithms SAC-IA and normal distributions transform

Table 1. Quantitative evaluation of classical data set registration experiments.

Data	Method	RMSE (mm)
Bunny 0° and 45°	Hellinger distance	1.051941
	Kullback-Leibler divergence	2.087637
	Correlation	1.283986
	The proposed method	0.389935
Bunny 45° and 90°	Hellinger distance	5.387755
	Kullback-Leibler divergence	11.445916
	Correlation	13.663601
	The proposed method	0.791662
Armadillo 270° and 300°	Hellinger distance	2.238819
	Kullback-Leibler divergence	2.066865
	Correlation	2.826190
	The proposed method	0.298217

(NDT). Experimental data are plant point clouds captured by RGB camera and modeled by COLMAP which is an open-source 3D reconstruction algorithm. At the same time, Gaussian noise with zero mean and variance of $3mr$ was added to the point cloud model to make the experiment closer to the real application scenario. The resolution of the point cloud is denoted as mr and is defined as

$$mr = \frac{1}{n} \sum_{i=1}^n \|p_i - p_{in}\|, \quad (23)$$

where p_{in} is the nearest point of p_i , and n is the number of points.

The inputs of the experiment and registration results of different methods are shown in Fig. 8. The inputs are point clouds with Gaussian noise, and their overlap rate is about 8%. The comparison of registration results shows that SAC-IA failed to register the two point clouds, and mistakenly spliced the point cloud of the trunk with that of the crown. Although NDT can roughly identify the transformation of the two point clouds, the registration result of the branch part still appears double shadow. The proposed method can register the two point clouds accurately, and the noise resistance and accuracy are better than the other two methods.

The RMSEs obtained from different methods are shown in Table 2. It can be seen that the RMSE obtained by the proposed method is 0.0168, which is the smallest than that of SAC-IA (1.5362) and NDT (0.0558). This indicates that the proposed method can obtain satisfactory registration of low overlapping plant point clouds, where the noise influence is effectively suppressed.

3.3 Registration Experiment of Multi-source Plant Point Cloud

Multi-source point clouds in real scenes have more noise, and the density between point clouds is diverse, which requires higher discrimination and robustness of registration

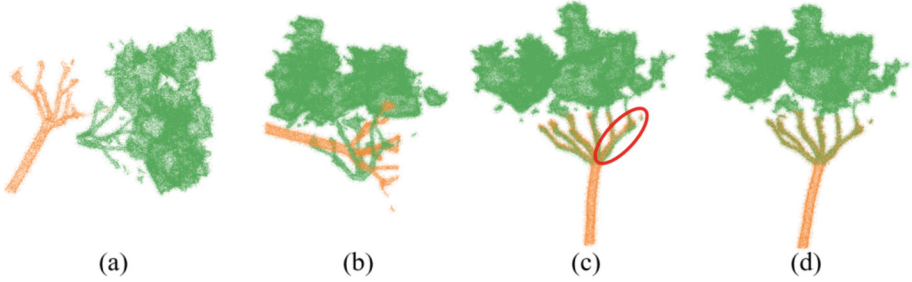


Fig. 8. The input of the experiment and the registration results of plant point clouds with different methods. (a) Input; (b) SAC-IA; (c) NDT; (d) The proposed method.

Table 2. Quantitative results of plant point cloud registration experiment with Gaussian noise.

Method	RMSE (m)
SAC-IA	1.5362
NDT	0.0558
The proposed method	0.0168

methods. The experiment in this section is to verify the effectiveness of the proposed method in registering plant point clouds captured by different sensors in real application scenes. Experimental data were captured by the RGB camera of unmanned aerial vehicle (UAV) and mobile phone respectively. As inputs, the point cloud captured on the ground by mobile phone and that captured by UAV is shown in Fig. 9. The overlap between the two point clouds is about 25%.

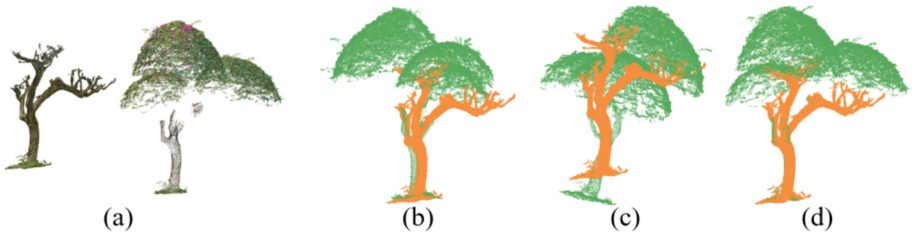


Fig. 9. The input of the experiment and the registration results of plant point clouds with different methods. (a) Input; (b) SAC-IA; (c) NDT; (d) The proposed method.

The proposed method was compared with SAC-IA and NDT, and the results are also shown in Fig. 9. The registration result of SAC-IA is shown in Fig. 9b. The position relation of the two point clouds is roughly correct, but the rotation angle has deviated. As shown in Fig. 9c, NDT gives a wrong transformation when registering the multi-source plant point cloud. The alignment result of the proposed method is more accurate, as shown in Fig. 9d. The branches in the point cloud are fitted without obvious position

and angle deviation. After manual measurement, the error at the maximum offset of the results of the proposed method is less than 1 cm, which is within the acceptable range.

According to the experimental results, in real application scenes, the proposed method can effectively register multi-source plant point clouds with a low overlap rate, and the error is within millimeter level. By splicing the point clouds captured by both terrestrial and airborne methods, more complete tree point clouds can be obtained, which lays a foundation for extracting more plant phenotypes.

4 Conclusion

Aiming at the difficulty of obtaining the complete point cloud model of tall plants, a method of point cloud registration in a real scene is proposed. The FPFH features are calculated according to the normal vector of the point cloud, and the similarity of features is evaluated by Bhattacharyya distance to obtain the initial matching pairs. Then, a filtering algorithm based on RANSAC is used to obtain the matching pairs with high accuracy. Finally, the transformation of the point cloud is obtained by singular value decomposition. The experimental results show that the proposed method is more accurate and helps to improve the success rate of point cloud registration for tall plants.

However, when using the proposed method, different application objects need to have different parameters. Therefore, providing the adaptive ability of parameters in the method will be our future research content.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (Grant Nos. 51905351 and U1813212), and the Science and Technology Planning Project of Shenzhen Municipality, China (Grant No. JCYJ20190808113413430).

References

1. Yang, X., et al.: A survey on smart agriculture: Development modes, technologies, and security and privacy challenges. *IEEE/CAA J. Automatica Sinica* **8**, 273–302 (2021)
2. Costa, C., Schurr, U., Loreto, F., Menesatti, P., Carpentier, S.: Plant phenotyping research trends, a science mapping approach. *Front. Plant Sci.* **9**, 1933 (2019)
3. Chen, J., Chen, J., Zhang, D., Sun, Y., Nanekaran, Y.A.: Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric.* **173**, 105393 (2020)
4. Santos Pereira, L.F., Barbon, S., Valous, N.A., Barbin, D.F.: Predicting the ripening of papaya fruit with digital imaging and random forests. *Comput. Electron. Agric.* **145**, 76–82 (2018)
5. Peng, Y., Yang, M., Zhao, G., Cao, G.: Binocular-vision-based structure from motion for 3-D reconstruction of plants. *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5 (2022)
6. Wu, S., et al.: MVS-Pheno: a portable and low-cost phenotyping platform for maize shoots using multiview stereo 3D reconstruction. *Plant Phenomics*. **2020**, 1–17 (2020)
7. Rueda-Ayala, V., Peña, J., Höglind, M., Bengochea-Guevara, J., Andújar, D.: Comparing UAV-based technologies and RGB-D reconstruction methods for plant height and biomass monitoring on grass Ley. *Sensors*. **19**, 535 (2019)
8. Cabo, C., Ordóñez, C., López-Sánchez, C.A., Armesto, J.: Automatic dendrometry: tree detection, tree height and diameter estimation using terrestrial laser scanning. *Int. J. Appl. Earth Obs. Geoinf.* **69**, 164–174 (2018)

9. Wang, Y., et al.: Maize plant phenotyping: comparing 3D laser scanning, multi-view stereo reconstruction, and 3D digitizing estimates. *Remote Sensing*. **11**, 63 (2018)
10. Vázquez-Arellano, M., Reiser, D., Paraforos, D.S., Garrido-Izard, M., Burce, M.E., Griepentrog, H.W.: 3-D reconstruction of maize plants using a time-of-flight camera. *Comput. Electron. Agric.* **145**, 235–247 (2018)
11. Chen, Z., Lv, H., Lou, L., Doonan, J.H.: Fast and accurate 3D reconstruction of plants using mvnnet and multi-view images. *Adv. Intell. Syst. Comput.* 390–399 (2021)
12. Ni, Z., Burks, T.: 3D dense reconstruction of plant or tree canopy based on stereo vision. *Agric. Eng. Int. CIGR J.* **20**, 248–260 (2018)
13. Guan, H., Liu, M., Ma, X., Yu, S.: Three-dimensional reconstruction of soybean canopies using multisource imaging for phenotyping analysis. *Remote Sens.* **10**, 1206 (2018)
14. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 239–256 (1992)
15. Zhang, C., Wei, Z., Xu, H., Chen, Y., Wang, G.: Scale variable fast global point cloud registration. *Chinese J. Comput.* **09**, 1939–1952 (2019). (in Chinese)
16. Tang, M., Zhao, H., Ding, H.: Research on binarized local feature descriptors of point clouds. *J. Mech. Eng.* **02**, 219–229 (2021). (in Chinese)
17. Xu, G., Pang, Y., Bai, Z., Wang, Y., Lu, Z.: A fast point clouds registration algorithm for Laser Scanners. *Appl. Sci.* **11**, 3426 (2021)
18. Salti, S., Tombari, F., Di Stefano, L.: Shot: Unique signatures of histograms for surface and texture description. *Comput. Vis. Image Underst.* **125**, 251–264 (2014)
19. Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M.: Aligning point cloud views using persistent feature histograms. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (2008)
20. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3D registration. In: 2009 IEEE International Conference on Robotics and Automation (2009)
21. Ge, L., Cai, Y., Weng, J., Yuan, J.: Hand PointNet: 3D hand pose estimation using point sets. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018)
22. Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J.: 3D object recognition in cluttered scenes with local surface features: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**, 2270–2287 (2014)