Junction-Based Correspondence Estimation of Plant Point Cloud Data Using Subgraph Matching

Ayan Chaudhury, Mark Brophy, and John L. Barron

***Abstract*—Laser scanner-captured 3-D point cloud data analysis is becoming more commonly used for remote sensing and plant science applications. Because of nonrigidity and complexity, reconstructing a 3-D model of a plant is extremely challenging. Existing algorithms often fail to find correct correspondences for plantlike thin structures. We address the problem of finding 3-D junction points in plant point cloud data as a first step of this correspondence matching process. Temporarily, we transform the 3-D problem into 2-D by performing appropriate coordinate transformations to the neighborhood of each 3-D point. Our proposed method has two steps. First, a statistical dip test of multimodality is performed to detect the nonlinearity of the local 2D structure. Then, each branch is approximated by sequential random-sample-consensus line fitting and a Euclidean clustering technique. The straight line parameters of each branch are extracted using total-least-squares estimation. Finally, the straight line equations are solved to determine if they intersect in the local neighborhood. Such junction points are good candidates for subsequent correspondence algorithms. Using these detected junction points, we formulate a correspondence algorithm as a subgraph matching problem and show that, without using traditional descriptor similarity-based matching, good correspondences can be obtained by simply considering geodesic distances among graph nodes. Experiments on synthetic and real (*Arabidopsis* plant) data show that the proposed method outperforms the state of the art.**

***Index Terms*—Correspondence matching, junction point detection, laser scanning, plant point cloud, 3-D registration.**

I. Introduction

REMOTE sensing and terrestrial applications often need to process botanic tree point cloud data, for example, for biomass estimation. With the advancements of remote sensing technologies and near-infrared laser scanners, 3-D automatic noninvasive analysis of growing plants is becoming possible. Recently, a new body of literature has appeared on point cloud imaging-based plant analysis for terrestrial applications [2], [3], [21].

Many of these applications require the construction of a 3-D model from multiple scans, and much work has already been done on this area. However, modeling the 3-D structure of rigid objects is less demanding than modeling complex plant structures. Problems such as nonrigidity, complex morphology, thin stem structures, etc., make the problem much more complicated. Point to point matching (*pairwise registration*) is a basic task in aligning two data sets and building the 3-D model. Very often in the computer vision literature, *feature points* are used to match two images. These points are usually points where there is a sharp discontinuity in some local feature. However, in the case of a 3-D plant point cloud, this kind of generalized feature can result in ambiguities because of the complex morphology of the plant. One motivation of this letter is to address this problem. We argue that *junction* points can be more effective than traditional features for registering plantlike structures (thin self-repetitive structures having branches). We show the efficacy of our method by matching corresponding junction points by a subgraph matching optimization technique.

This letter is an improvement of the work proposed by Bucksch and Khoshelham [2]. We present three major contributions. **First**, we consider the case of pairwise registration when the initial alignment is not known in advance between two views of the plant. **Second**, we propose an efficient junction detection algorithm for matching of feature points. **Third**, we formulate the correspondence estimation as a subgraph matching optimization technique and demonstrate better results than that of the state of the art.

Manuscript received August 25, 2015; revised December 3, 2015 and April 28, 2016; accepted May 10, 2016. Date of publication July 7, 2016; date of current version July 20, 2016. This work was supported by an National Science and Engineering Council of Canada Discovery grant.

The authors are with the Department of Computer Science, University of Western Ontario, London, ON N6A 5B7, Canada (e-mail: achaud29@csd.uwo.ca, mbrophy5@csd.uwo.ca, barron@csd.uwo.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2016.2571121

II. Related Work

Pairwise registration is a fundamental task in processing 3-D point cloud data. Registration algorithms can be broadly classified into two groups. The first type is a global registration method such as Coherent Point Drift (CPD) [15] or Gaussian Mixture Models (GMM) [10]. The second type is based on the computation of local feature descriptors and matching [11], [13], [14]. Global registration methods often fail because of the local deformation between two views. On the other hand, the basic idea of feature point extraction is to find changes in local structure (intensity or local geometry). We consider only raw plant point cloud data, without any intensity or color information. Applying a standard 3-D key-point detector algorithm relies on local descriptors [14], [16] for feature extraction, and matching is performed based on descriptor similarity. This idea works well for indoor/outdoor scene point cloud with color information and models with well-defined surface and curvature. However, applying this idea to plantlike structures, which contain many similar geometrical features, usually results in ambiguous matches in a correspondence calculation. Matching correspondences in forest scenes are studied in some recent articles [1], [9]. However, the problem of finding efficient correspondences still remains unsolved.

Bucksch and Khoshelham [2] addressed this problem and proposed a localized registration algorithm. The method is highly dependent on the skeletonization, which might not work

in the presence of noise and plant leaves. Also, their method is based on the assumption that an initial guess is available, i.e., they assume rough manual alignment of the junction feature points in two trees. For large rotation angle differences, the algorithm may fail to register two views correctly. This letter removed the need for manual alignment. Recently, Zhou *et al.* [21] presented an extended approach [2]. However, their method is still dependent on skeletonization, which still involves manual intervention and does not handle occlusion and local deformation. Also, the idea of cylinder fitting does not work for thin stems.

We hypothesize that, in the case of plant structures, the detection of junction points will provide good feature points for matching. A junction is the point where two or more branches meet. That is, the local neighborhood of a junction point can be considered to be a nonlinear, rather than the local neighborhood of a point on a linear single stem, or uniform leaf structure.

In the next section, we describe the rotations and translations needed to convert the 3-D problem into an equivalent 2-D problem. Then, we discuss the dip test for multimodality. After demonstrating RANdom SAmple Consensus (RANSAC) fitting and total least squares (TLS) approximations, we demonstrate subgraph matching for correspondence estimation. Finally, experimental results and analyses for synthetic and real data sets are given.

III. Coordinate Transformation

We use a kd-tree algorithm to compute the nearest neighbor points within a certain radius (set at 2.0 mm). Given such points in a local neighborhood about some 3-D point, we transform the data so that the surface normal of the plane fitting the data is a line-of-sight vector (0, 0,1).

More specifically, we compute the center of mass $(x_{\rm cm}, y_{\rm cm}, z_{\rm cm})$ of the neighborhood 3-D points. To reformulate as a 2-D problem, we perform the following steps: translate the origin to the center of mass by $-(x_{\rm cm}, y_{\rm cm}, z_{\rm cm})$, rotate about the x-axis onto the $x-z$ plane by Euler angle α, rotate about the y-axis onto the longitudinal axis (0, 0, 1) by Euler angle β, and finally transform the origin back to the previous location by $(x_{\rm cm}, y_{\rm cm}, z_{\rm cm})$. Next, a plane of the form $ax+by+cz+d=0$ is fitted to the neighborhood data using Crammer's rule. The parameters $\vec{n} = (a, b, c)$ are the plane's surface normal. We checked the residual of this fit for many randomly chosen 3-D neighborhoods, and the residual is always very small, conforming our local planarity assumption.

Since these transformations result in vertical surface normals, we need only be concerned with the structure in the $x-y$ plane, i.e., the problem is now 2-D.

IV. DIP Test for Multimodality

The detection of multimodality in numeric data is a well-known problem in statistics. A probability density function having more than one mode is denoted as a multimodal distribution. Hartigan and Hartigan [8] proposed a dip test for unimodality by maximizing the difference between the empirical distribution function and the unimodal distribution function. In the case of a unimodal distribution, the value for the dip should asymptotically approach 0, while for the multimodal case, it should yield a positive floating point number. Inspired by the work of Zhao and Bhotika [19], we use the dip test for detecting nonlinearity in the data. Points having a nonlinear local neighborhood are potential candidates for a junction point. The idea is to perform the dip test for a local neighborhood of a point. If it is a stem or a leaf, the data should be uniform, and the distribution should only be unimodal. For a junction point likely due to a bifurcation, it should exhibit multimodality. We use the dip value as a measure of multimodality (see Fig. 1).

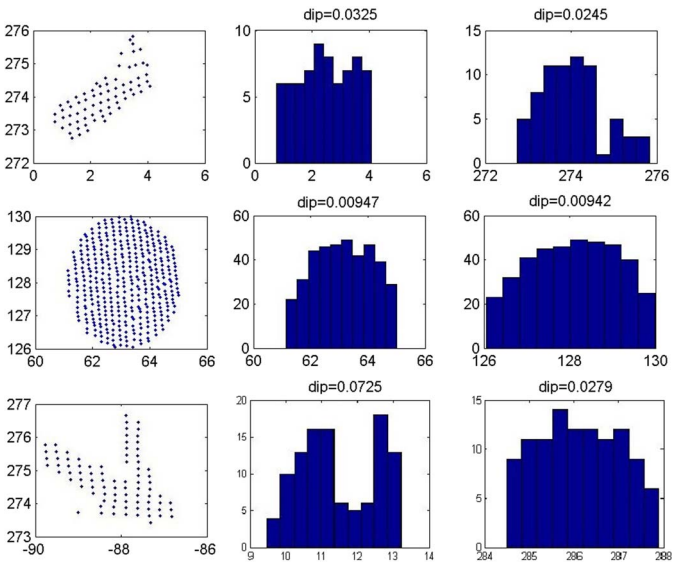

Fig. 1. Distribution of data: Column 1 is the neighborhood point clouds under consideration, column 2 is the histograms of the x-coordinate distribution, and column 3 is the histograms of the y-coordinate distribution for (first row) a single stem, (second row) a leaf, and (last row) a stem with two branches. The latter is potentially a junction point.

We perform this dip test along the x and y directions (note that, as we have reduced the dimensionality from 3-D to 2-D, the z-coordinates can be ignored) and obtain the maximum dip value. The neighborhood is determined to be multimodal if the dip value is over some threshold. However, the threshold value of the dip is highly dependent on the data and should be tuned carefully. In our case, we empirically set it to 0.04 to obtain useful results.

We use the dip measurement for the initial filtering of non-junction neighborhood data. Note that nonlinearity and high dip values in the local neighborhood do not guarantee that those points are junction points. For some leaf and stem data, we notice that, sometimes, the data show high dip values. Instead of relying blindly on dip test results, we do further processing, described in the next section.

V. RANSAC Fitting and TLS Approximation

We consider, at most, three branches at an intersection point. Three branches may intersect at a single point [the red dot in Fig. 2(c)] or at two different points [the red dots in Fig. 2(d)]. We also assume that the main stem will be thicker than the branches. We extract this thick stem using RANSAC straight line fitting using a high distance threshold for inliers (empirically chosen as 0.8 mm for our case). Sequential RANSAC

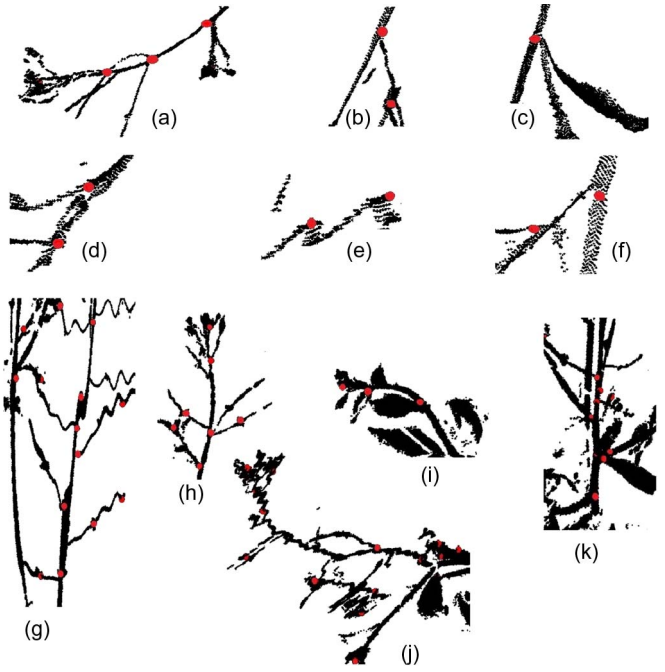

Fig. 2. (Red dots) Examples of detected junction points on the real Arabidopsis plant data.

is used to fit a straight line for every branch (set to 0.5 mm for nonstem branches). However, there may be other points due to additional branches, a leaf or a noise event (or some combination of the three). After removing the RANSAC-fitted main stem, we perform Euclidean clustering on the rest of the data and choose the biggest connected component(s) to extract the subbranches. Two sets of points, $\mathcal{X}_i = \{p_i \in \mathcal{P}\}$ and $\mathcal{X}_j = \{p_j \in \mathcal{P}\}$, form two different clusters if the following condition holds: $\mathbf{min}\|p_i - p_j\|_2 \geq \tau$, where τ is the distance threshold (set to 1.0). The branches may be straight or curved, but by using RANSAC, we can estimate the principal direction of the branch [17]. A criterion is imposed to estimate a broken branch shape (due to occlusion): We merge two branches if they are spatially close to each other and have roughly the same direction.

After estimating the points for each branch, we use TLS to approximate the straight line represented by a set of points in a branch and extract the parameters by minimizing $E = \sum_{i=1}^{n}(ax_i + by_i + c)^2$ and check if the intersection point is contained in the local neighborhood or not. Note that the obtained intersection point is 2-D, so we can apply the inverse coordinate transformation determined earlier to compute the actual 3-D point. Finally, we perform nonmaximal suppression based on the highest dip value to reduce the number of points. The distance threshold for nonmaximal suppression is chosen as 4 mm (the same threshold is used also in [14]). The procedure is described in Algorithm 1.

Algorithm 1 Junction Point Detection

Require: Neighborhood point set $\mathcal{P} = \{x_1, x_2, \ldots, x_n\}$
1: Translate $\mathcal{P}$ to the center of mass
2: Rotate about x-axis
3: Rotate about y-axis
4: Retranslate $\mathcal{P}$ back to the previous origin
5: Compute dip along x- and y-axes: dip_x, dip_y
6: **if** $\max(dip_x, dip_y) > \tau$ **then**
7: Fit RANSAC to $\mathcal{P}$ & store the point set into $\mathcal{M}_1$. Remove the points from $\mathcal{P}$, resulting in $\mathcal{P}_1 = \mathcal{P} - \mathcal{M}_1$
8: **if** $|\mathcal{P}_1| > n_t$ {sufficient number of points} **then**
9: Extract the two largest Euclidean clusters $\mathcal{O}_1$, $\mathcal{O}_2$ of $\mathcal{P}_1$ {Locate subbranch(s)}
10: **if** $|\mathcal{O}_1|, |\mathcal{O}_2| > n_k$ {sufficient number of points} **then**
11: Fit RANSAC to $\mathcal{O}_1$ (and $\mathcal{O}_2$) & store the point set in $\mathcal{M}_2$ (and $\mathcal{M}_3$)
12: Apply TLS to $\mathcal{M}_1$, $\mathcal{M}_2$ (and $\mathcal{M}_3$) and obtain the straight line parameters
13: Solve the straight line equations using the computed parameters for $\mathcal{M}_1, \mathcal{M}_2$ (and $\mathcal{M}_1, \mathcal{M}_3$). Store the computed point(s) in J_1 (and J_2)
14: Check if the corresponding 3-D coordinates of J_1 (and J_2) are contained in $\mathcal{P}$. If J_1 and J_2 are close enough, merge them into $J = (J_1 + J_2)/2$
15: **return** J (or J_1, J_2) {One or two junction points}
16: **end if**
17: **end if**
18: **else**
19: **return** NULL {No junction point}
20: **end if**

VI. Correspondence Matching

The detected junction points from the last phase are potential candidates for correspondences and can be used as feature points for matching. For raw 3-D point cloud data, local surface normals, neighborhood information, etc., are typically used for encoding the local structure, and points are matched based on the descriptor similarities. We observed empirically that this idea fails for plant data because the thin structures do not allow for good local surface normal calculations, and because of deformations, the local structure can change abruptly in adjacent images. This problem was also addressed by Duchenne *et al.* [6] where triplets of feature points were used to compute mutual angles. However, they assume that the coordinate axes are known in advance. Also, the smart selection of triangles is an important factor in computing good correspondences. There are many other properties that can be considered for mutual matching, but we show that simple geodesic distance can be used efficiently in finding correct correspondence, emphasizing the simplicity and effectiveness of the proposed algorithm.

A. Graph Formation

First, we triangulate the data using the Delaunay triangulation in 3-D (note that we converted the problem temporarily to 2-D just for the detection of junction points). Using the vertex information from triangulation, we construct a graph connecting all the points. To handle the cases of missing or occluded data, we connect the points to the nearest triangle vertex so that all the points are included in a single graph. Then, for each junction point, we apply Dijkstra's shortest path

algorithm to compute the geodesic distance to all other junction points. The same procedure is followed for the second point cloud as well. Then, we exploit pairwise distances to be the criteria for matching. This kind of local descriptorless approach was also proposed by Leordeanu *et al.* [12]. However, they assumed some prior knowledge of the object categories. In our case, we do not have any prior knowledge or make any assumptions about the data.

B. Subgraph Matching Formulation

Given all the pairwise distances of all junction points, we formulate our correspondence matching problem as a subgraph matching problem. Consider two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Each junction point is considered to be a node of the graph. Each node stores the geodesic distances to all other nodes. In the end, this yields a set of edges. The compatibility of two nodes in G_1 and G_2 is defined as a closest distance match. For example, let us suppose that two graphs G_1 and G_2 have n_1 and n_2 nodes. Each node V_{1_i} in G_1 stores all distances to all other nodes. We denote this as the set of attributes of node V_{1_i}: $\mathcal{D}_{v_{1_i}} = \{d_{v_{1_i}v_{1_j}}\}, \forall j \in n_1$. Similarly in G_2, the set of attributes of node V_{2_i} is defined as $\mathcal{D}_{v_{2_i}} = \{d_{v_{2_i}v_{2_k}}\}, \forall k \in n_2$. The compatibility of two nodes, V_{1_i} and V_{2_i}, is formulated as the sum of the squares of the difference of the nearest distances, multiplied by the number of matches. Suppose that G_1 and G_2 contain five and seven nodes, respectively. Let the attributes of a node V_{1_i} contain the following distances: $\{d_1, d_2, d_3, d_4\}$ (ignoring self-distance). Similarly, V_{2_i} contains the distances $\{d'_1, d'_2, d'_3, d'_4, d'_5, d'_6\}$. We use a threshold $\epsilon(= 0.2)$ for the match of two distances. Suppose that there are three distance matches given by $d_1 \sim d'_4$, $d_3 \sim d'_2$, and $d_4 \sim d'_1$. Then, the affinity of the two vertices is computed as

$$\mathcal{A}_{v_{1_i}v_{2_i}} = 3 * \left[(d_1 - d'_4)^2 + (d_3 - d'_2)^2 + (d_4 - d'_1)^2\right]. \quad (1)$$

The logic for using this kind of distance matching is that any outlier is likely to be eliminated by a lower number of matches. On the other hand, compatible points will only have the maximum number of distance matches.

C. Matching of Nodes

Using the compatibility of two vertices, we obtain the initial node correspondence by using the Hungarian algorithm. The outliers are likely to get rejected by unmatched distance attributes. However, there still may be nonoptimal matches of the vertices. We follow the approach of attribute graph matching proposed by Cour *et al.* [4]. Given two graphs $G_1 = (V_1, E_1, A_1)$ and $G_2 = (V_2, E_2, A_2)$, where each edge $e = V_iV_j \in E$ has an attribute A_{ij}, the objective is to find $\mathcal{N}$ pairs of correspondences (V_i, V_j), where $V_i \in V_1$ and $V_j \in V_2$, the affinity A_{ij} [see (1)] defines the quality of the match between nodes V_i and V'_i. Denoting the similarity function of pairwise affinity as $f(\cdot, \cdot)$, the matching score can be computed as

$$\lambda(\mathcal{N}) = \sum_{ii', jj' \in \mathcal{N}} f(\mathcal{A}_{ij}, \mathcal{A}'_{i'j'}). \quad (2)$$

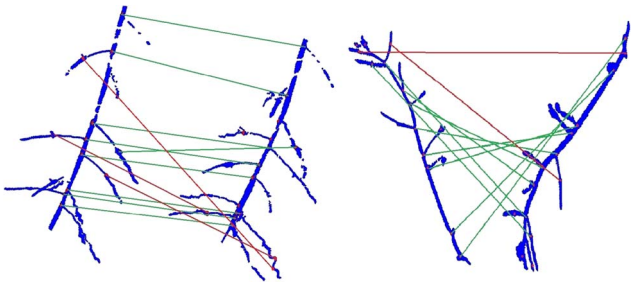

Fig. 3. Two examples of correspondence matching (green lines are good, and red lines are bad matches which are eliminated by RANSAC).

Representing $\mathcal{N}$ as a binary vector x so that $x(ii') = 1$ if $ii' \in \mathcal{N}$, the aforementioned equation can be written as

$$\max \ \lambda(x) = x^T W x \quad (3)$$

where $W_{ii',jj'} = f(\mathcal{A}_{ij}, \mathcal{A}'_{i'j'})$. The optimal solution of the aforementioned equation is given by

$$x^* = \arg\max(x^T W x). \quad (4)$$

The permutation matrix provides the correspondence among the vertices (or the junction points in our case). Finally, the outliers (or wrong matches) are pruned out using RANSAC.

VII. Experimental Results

First, we tested our method on a synthetic data set. We have used *Vascusynth* [18] to generate artificial artery data, used their ground truth for junction points, and have obtained 100% correct matches.

Second, we show the result of junction point detection (red dots) on a variety of Arabidopsis data sets in Figs. 1 and 2. The data set is extremely challenging due to occlusion, missing data, and deformations. Results show the robustness of the algorithm in detecting junction points correctly. We have validated our junction detection algorithm by showing efficient correspondence matching based on these feature points. Fig. 3 shows two examples from the data set. The left image shows matching on the data set having local deformations and missing data, whereas the right image shows the matching when the data set is rotated 180° relative to each other. The few bad matches (red lines) are eliminated by RANSAC in both cases.

We compare our results with three state-of-the-art algorithms: CPD [15], GMM [10], and Scale Invariant Spin Image (SISI) [5]. CPD and GMM are well-established methods for global registration, and SISI is a state-of-the-art algorithm for feature-based correspondence matching, which has been shown to perform better than Scale Invariant Feature Transform (SIFT) [13] and the spin image descriptor [11]. We observe that both CPD and GMM perform poorly in the cases of occlusion, deformation, and missing data. We infer that, since these algorithms are based on global optimization, they often cannot handle ambiguities in local structure, whereas our algorithm performs well for these cases (the left image in Fig. 3 is an example of such a case). It is also observed that, for large rotation angles between the two images, the performance of CPD and GMM fails drastically. Fig. 4 shows a few point-to-point correspondences obtained from the registration results

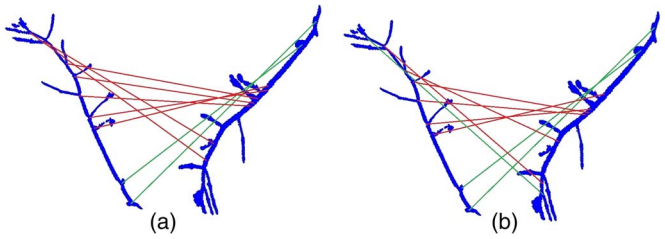

Fig. 4. Point-to-point correspondence results obtained from (a) CPD and (b) GMM for large rotation angle. Good matches are shown in green lines, and bad matches are in red lines.

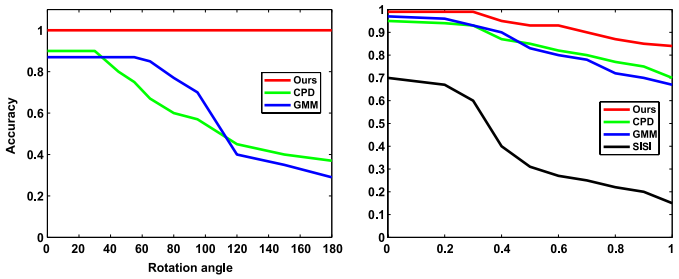

Fig. 5. Effect of rotation angle and performance: (a) Rotation angle (in degrees) is plotted against (normalized) accuracy. (b) Precision–recall curve comparing CPD, GMM, SISI, and our method.

for CPD and GMM for the rotated data in the right image in Fig. 3. For GMM-based algorithms, we have experimented with different parameter settings and reported results for the best case. More specifically, we have used the default values of [10] in their publicly available code. For CPD [15], we have set the regularization weight $\lambda = 3$ and the width of the Gaussian kernel (smoothness) $\beta = 2$. We have tried other different values of λ but only recovered poorly warped registrations.

CPD and GMM fail to perform well because the algorithms get stuck in local minima for large rotation angles. However, using our graph-based matching method, the rotation angle does not affect the performance. We present a quantitative comparison of CPD, GMM, and our method for varying rotation angle in the left graph of Fig. 5. We perform a similar analysis on increasing rotation angle as in [20].

We have also performed experiments using the SISI descriptor matching [5]. The algorithm failed to find meaningful correspondences between two views. We believe that the reason for this is that the algorithm's dependence on local descriptors based on surface normals tends to be unreliable for our plant data. The precision–recall curves in the right image in Fig. 5 show the superior performance of our method over CPD, GMM, and SISI.

VIII. Conclusion and Future Work

We have presented an approach to detect junction points in plant point cloud data and showed its efficacy for correspondence matching over the state of the art. The proposed junction point detection algorithm is simple and fast and does not need any *a priori* knowledge. The time complexity of the algorithm is $O(n^2)$. One drawback of the proposed method is that it is parameter sensitive. Some of the parameters such as the RANSAC threshold for the inlier, distance threshold for graph matching, cutoff value for the dip test, and threshold for nonmaximal suppression would need to be tuned for different applications. An immediate research direction will be investigating the cases where the branches are highly curved and contain complex leaf structures for a larger variety of plant types.

Acknowledgment

The authors would like to thank the anonymous reviewers whose thoughtful comments and suggestions led to a much improved manuscript.

References

[1] A. Bienert and H. Maas, "Methods for the automatic geometric registration of terrestrial laser scanner point clouds in forest stands," in *Proc. Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, 2009, vol. 38-3/W8, pp. 93–98.

[2] A. Bucksch and K. Khoshelham, "Localized registration of point clouds of botanic trees," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 3, pp. 631–635, May 2013.

[3] A. Bucksch, R. Lindenbergh, M. Z. A. Rahman, and M. Menenti, "Breast height diameter estimation from high-density airborne LiDAR data," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 6, pp. 1056–1060, Jun. 2014.

[4] T. Cour, P. Srinivasan, and J. Shi, "Balanced graph matching," in *Proc. NIPS*, 2007, pp. 1–8.

[5] T. Darom and Y. Keller, "Scale invariant features for 3D mesh models," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2758–2769, May 2012.

[6] O. Duchenne, F. Bach, I. Kweon, and J. Ponce, "A tensor-based algorithm for high-order graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 33, pp. 2383–2395, Dec. 2011.

[7] D. Fehr, A. Cherian, R. Sivalingam, S. Nickolay, V. Morellas, and N. Papanikolopoulos, "Compact covariance descriptors in 3D point clouds for object recognition," in *Proc. ICRA*, 2012, pp. 1793–1798.

[8] J. Hartigan and P. Hartigan, "The dip test of unimodality," *Ann. Statist.*, vol. 13, no. 1, pp. 70–84, 1985.

[9] J. Henning and P. Radtke, "Multiview range-image registration for forested scenes using explicitly-matched tie points estimated from natural surfaces," *ISPRS J. Photogramm. Remote Sens.*, vol. 63, no. 1, pp. 68–83, Jan. 2008.

[10] B. Jian and B. Vemuri, "Robust point set registration using Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1633–1645, Aug. 2011.

[11] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.

[12] M. Leordeanu, M. Hebert, and R. Sukthankar, "Beyond local appearance: Category recognition from pairwise interactions of simple features," in *Proc. CVPR*, Jun. 2007, pp. 1–8.

[13] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[14] A. Mian, M. Bennamoun, and R. Owens, "On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes," *Int. J. Comput. Vis.*, vol. 89, no. 2/3, pp. 348–361, Sep. 2010.

[15] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, Dec. 2010.

[16] B. Steder, R. Rusu, K. Konolige, and W. Burgard, "Point feature extraction on 3D range scans taking into account object boundaries," in *Proc. ICRA*, 2011, pp. 1–8.

[17] M. Uhercik, J. Kybic, H. Liebgott, and C. Cachard, "Model fitting using RANSAC for surgical tool localization in 3D ultrasound images," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 8, pp. 1907–1916, Aug. 2010.

[18] VascuSynth. [Online]. Available: http://vascusynth.cs.sfu.ca

[19] F. Zhao and R. Bhotika, "Coronary artery tree tracking with robust junction detection in 3D CT angiography," in *Proc. ISBI*, 2011, pp. 2066–2071.

[20] F. Zhou and F. De La Torre, "Deformable graph matching," in *Proc. CVPR*, 2013, pp. 2922–2929.

[21] G. Zhou, B. Wang, and J. Zhou, "Automatic registration of tree point clouds from terrestrial lidar scanning for reconstructing the ground scene of vegetated surfaces," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 9, pp. 1654–1658, Sep. 2014.