2016 IEEE International Conference on Robotics and Automation (ICRA)
Stockholm, Sweden, May 16-21, 2016

A Novel Framework for Modeling Dormant Apple Trees using Single Depth Image for Robotic Pruning Application

Shayan A. Akbar[1], Noha M. Elfiky[2], and Avinash Kak[3]

Abstract— **Dormant pruning is one of the most costly and labor-intensive operations in specialty crop production. During winter, a large crew of trained seasonal workers has to carefully remove the branches from hundreds of trees using a set of pre-defined rules. The goal of automatic pruning is to reduce this dependence on a large work-force that is currently needed for the job. Automatically applying the pruning rules entails construction of 3D models of the trees in their dormant condition (that is, without foliage) and accurate estimation of the pruning points on the branches.**

This paper investigates the contribution of modeling the trunk and primary branches of a tree using semi-circles in a 3D reconstruction scheme. It involves three main steps; estimating the diameter-error, filtering the depth images, and semi-circle-based modeling which estimates the required pruning data. The obtained results demonstrate the effectiveness of the proposed framework for automatic reconstruction using only single depth image. We further propose a novel empirical model that estimates the diameter of the primary branches under varying depth values. Both qualitative and quantitative evaluations are performed on a new challenging apple trees dataset under various indoor and outdoor environments. Our results show that the proposed scheme provides a performance of 89% for correctly estimating the diameter of the primary branches.

Index Terms— **Dormant Pruning, 3D Model of Apple Trees, Depth Sensing, Semicircle Estimation**

I. Introduction

Accurate 3D reconstruction of apple trees using depth images is a critical problem and is essential for the automation of dormant pruning which involves cutting of certain primary branches that are connected directly with the trunk of a tree. Dormant pruning is an important process in the overall cycle of specialty crop production as it helps in improving the quality as well as the yield of the fruit. In agriculture literature, there is a set of rules which dictates if a branch of a tree needs to be pruned [1], [2]. These rules mostly require accurate measurements of the diameters of branches and trunk. Errors in these measurements can lead to incorrect cutting of "good" branches which had a potential to produce large quantity of fruit. This can result in a significant reduction in the overall yield of the orchard. Therefore, in order to automate the process of pruning, accurate 3D reconstruction of apple trees is of primary importance and is the focus of research presented in this paper.

3D reconstruction requires depth information of the scene (i.e., tree in our case) with respect to position of the camera. Many modern reconstruction schemes [3], [4], [5], [7], [8], [9], [14], [15] capture this depth information using depth sensors or cameras. One such portable, and inexpensive depth camera is Kinect 2 which provides a depth image of the surrounding scene using Time of Flight technique. But a major problem that inhibits the process of accurate 3D reconstruction is the presence of noise near and on the edge pixels of the tree in the depth images obtained from Kinect 2 sensors. This noise is problematic as it reduces the accuracy of reconstruction and affects the estimation of diameters.

In a modern high-density orchard environment there are several other factors which affect the accuracy of reconstruction of trees. Background clutter, illumination variations, complexity of tree structures, partial occlusions, and close proximity of the consecutively planted trees are some of the many problems which make accurate 3D reconstruction a challenging problem. In this paper we focus on resolving some of these issues by proposing a new framework for reconstruction of trees.

We propose a method for accurate 3D reconstruction of trunk and primary branches that relies on the fact that dormant trees have a predominantly cylindrical structure — that is, any cross-section when viewed from either front or back side, can be approximated by a semicircle. Exploiting this fact, the method divides the point cloud of tree obtained from a depth image into several cross-sections and fits a semicircle on each cross-section in least-squares sense using Levenberg-Marquardt algorithm [6].

We also study the accuracy of Kinect 2 sensor to represent the thickness or diameter of a certain cross-section with varying depth values. We observe that its accuracy reduces as the depth at which the cross-section is present increases. Hence, the branches that are present at greater depths would contain more error in their measurements. In this work, we also propose an empirical model for adjusting the calculated diameter value, subsequently, making it more accurate.

The rest of the paper is organized as follows. In Section II, we will review the related work. Then, we will discuss the proposed framework in detail in Section III. In Sections IV and V we will describe the experimental setup and evaluate the performance of the algorithm on both indoor and outdoor trees, respectively. Finally, we will conclude the paper and highlight potential future works in Section VI.

*This work was supported by the US Department of Agriculture (USDA)

[1]Shayan A. Akbar is a PhD student in the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, US sakbar@purdue.edu

[1]Noha M. Elfiky is a Research Scientist in the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, US nelfiky@purdue.edu

[2]Avinash Kak is a Professor in the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, US kak@purdue.edu

II. Related Work

The inexpensive and portable Kinect sensors have recently become very popular in the computer vision community as they provide depth images of the scene at a very high frame rate (i.e., 30 fps). Although, the quality of these depth images is comparable with those obtained from the expensive and heavyweight LASER scanners, it is still not very accurate. Several studies have been performed recently for evaluating the accuracy of Kinect sensors and for modeling different types of noise present in the depth images [10], [11], [12], [13]. However, none of these studies examine the erroneous measurements in the diameters of cylindrical objects which is critical to our 3D reconstruction application.

Reconstructing a 3D model of a scene is an active research area in computer vision. KinectFusion [7] software is a tool associated with Kinect sensor that reconstructs the surrounding scene in real-time from depth images. But, KinectFusion as well as its extensions [8], [9] reconstruct generic scenes without putting any constraint on the object in the scene. Subsequently, the reconstructed output can be inaccurate especially for challenging real-world applications —such as 3D reconstruction of trees for dormant pruning. Also, there exists in the literature several methods that focus on 3D reconstruction of trees [3], [4], [5], [14], [15]. However, these methods estimate the tree model without taking into consideration the erroneous diameter-measurements of the cross-sections. In addition, they don't exploit the fact that trees are predominantly cylindrical in structure.

In this work, we propose a new framework that builds 3D reconstruction of apple trees that accounts for the above mentioned error while utilizing the predominantly cylindrical structure of trees. In particular, each side of a tree is modeled using a set of stacked semicircles which are defined by their centers and diameters. We estimate the diameters of the semicircles using the error model that we have developed in this study while the centers are estimated based on least-squares fitting method [6]. Moreover, in order to obtain the final reconstruction we rely on a single depth image, as opposed to the methods mentioned above.

To this end, we will highlight the novelties of this paper which are threefold. First, we propose a new framework for reconstructing dormant apple trees using semicircle fitting in 3D space. Second, we put forward a new error correction model that accurately estimates the diameter of each cross-section of a tree —even when the data is noisy. Lastly, we publish a new dataset [1] for six dormant apple trees which are collected from various orchards. The dataset consists of the depth images of each tree as well as the ground truth measurements of the diameters of its primary branches.

III. Framework

In this section we will discuss the proposed framework for obtaining the 3D reconstruction of dormant apple trees in detail. In particular, we will explain its three main steps

[1]The dataset can be downloaded from: https://engineering.purdue.edu/RVL/ICRA_Dataset/

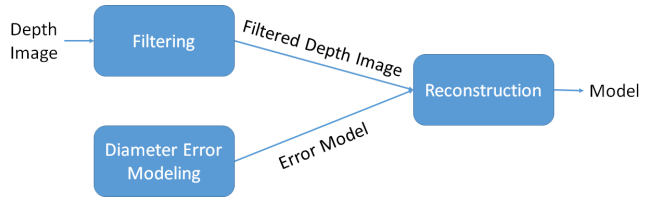


Fig. 2: A block diagram of the framework is shown.

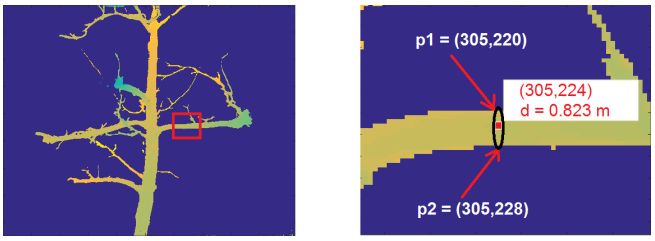


Fig. 3: A section of a tree is shown on left and a cross-section of the tree is highlighted on right with black oval.

(A) modeling the errors in diameter calculations obtained from depth images , (B) filtering the depth images to remove the lens distortion and background clutter, and (C) 3D reconstruction based on semicircles. The overall framework is shown in Fig. 2.

A. Modeling the Errors in Diameter Calculations Obtained from Depth Images

Kinect sensor does not capture points that lie near the edge of the trunk and primary branches in the depth images. Consequently, there exists some error between the ground truth diameter-measurements of the primary branches D_A and their corresponding diameter-calculations obtained from the depth image D_C. To measure D_A of a tree cross-section we use a caliper, and we calculate its corresponding D_C value from the depth image as follows:

$$D_C = \frac{d}{f}\|p_1 - p_2\| \tag{1}$$

Where d is the depth value at the center of the a cross-section (shown as black oval in Fig. 3), f is the focal length of the sensor, and p_1 and p_2 are the edge points of the indicated cross-section.

To model the error between D_C ands D_A under different depth distances from

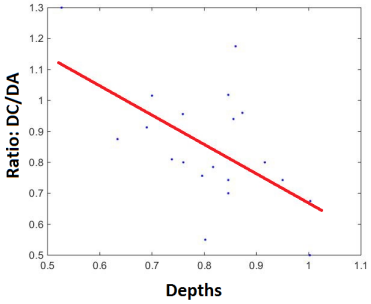


Fig. 4: Ratio of calculated and actual diameters with varying depths for twenty cross-sections is shown with blue points. The estimated line is also given with red color.

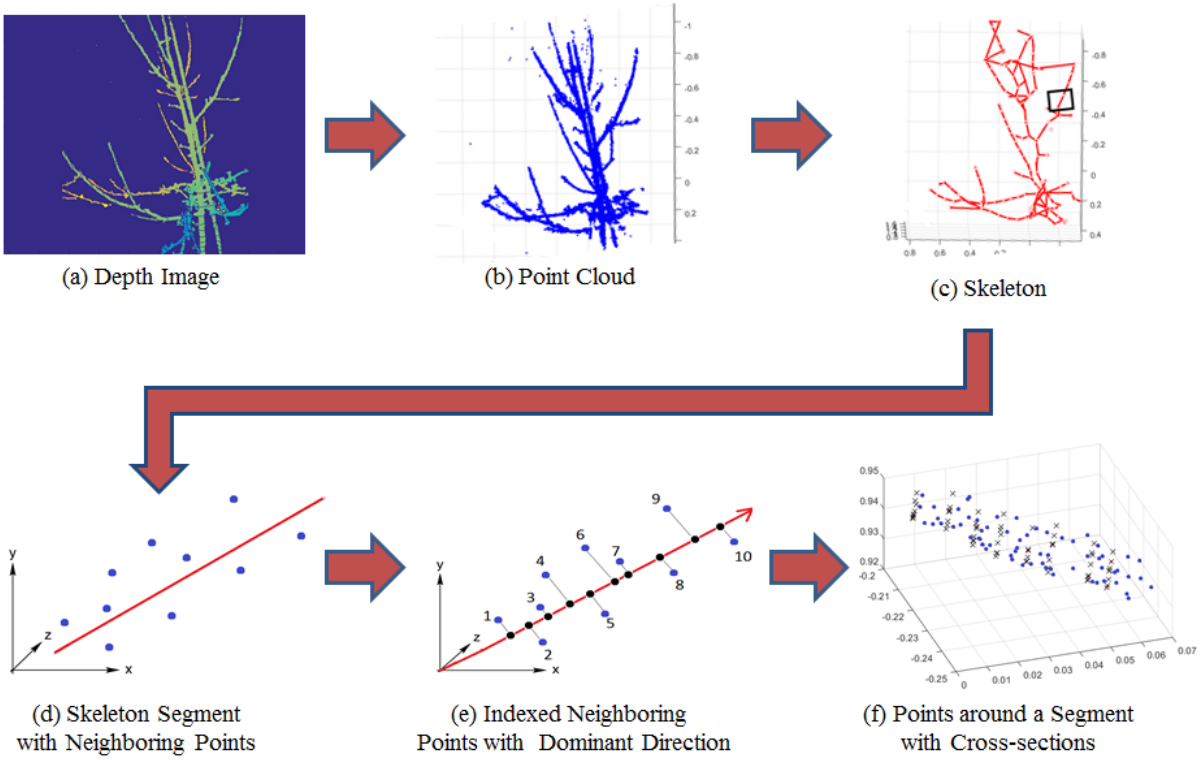


Fig. 1: The pipeline for obtaining cross-sections from depth image is shown.

the sensor, we examine various cross-sections of a tree. In particular, for each cross-section we record the depth value of its center pixel as well as the ratio between D_C and its corresponding D_A. In Fig. 4, we show the ratio values between D_C and its corresponding D_A (on the y-axis) under varying depth values (on the x-axis) for twenty cross-sections. From the figure, it can be inferred that as depth value increases the error between D_C and D_A increases. Consequently, we fit a linear model on the data of Fig. 4 based on the following equation:

$$\frac{D_C}{D_A} = -0.95d + 1.6 \tag{2}$$

Lastly, we use the obtained model in order to estimate the diameter of a semicircle as shown in the block diagram (Fig. 2).

B. Filtering Depth Images to Remove the Lens Distortion and Background Clutter

The main focus of this section is to discuss the two necessary operations that are required to clean the depth images before further processing. These operations include lens distortion removal that affects the accuracy of the reconstruction [18], [19] and background subtraction. To remove lens distortion, we calculate its parameter vector $\mathbf{k}$ by calibrating the camera as in [17]. Subsequently, the undistorted depth map can be obtained as follows:

$$\mathbf{X}_t = \begin{bmatrix} 2k_3xy + k_4(r^2 + 2x^2) \\ k_3(r^2 + 2y^2) + 2k_4xy \end{bmatrix} \tag{3}$$

$$\mathbf{X}_u = (1 + k_1r^2 + k_2r^4 + k_5r^6)\mathbf{X} + \mathbf{X}_t \tag{4}$$

Where $\mathbf{X} = (x, y)$ is the input depth map, $\mathbf{X}_t$ is the tangentially undistorted map, $\mathbf{X}_u$ is the radially and tangentially undistorted map, and $r^2 = x^2 + y^2$. The parameter vector $\mathbf{k} = (k_1, k_2, k_3, k_4, k_5)$, where k_1, k_2, and k_5 are the radial distortion parameters while k_3, and k_4 are the tangential distortion parameters.

Apart from lens distortion, depth images also contain noisy background that needs to be removed. For this, we segment the foreground object from the background clutter by thresholding the pixel values of the depth image that are greater than a certain threshold (τ meters) to zero, as discussed in Section IV-A.

C. 3D Reconstruction Based on Semicircles

Our 3D reconstruction method is based on fitting semicircles on the cross-sections of tree point cloud. The steps involved in the method are discussed below in detail.

1) Converting Depth Image to Point Cloud: The first step in our approach is to obtain the point cloud representation of a tree as shown in Fig. 1(b). To do this, we convert the depth image of a tree to a point cloud $\mathbf{V}$ based on following equation as in [7]:

$$\mathbf{V}(\mathbf{u}) = D(\mathbf{u})K^{-1}\mathbf{u} \tag{5}$$

Where $\mathbf{u} = (x, y, 1)$ is an image pixel in the homogeneous coordinate system with a depth value $D(\mathbf{u})$ and K is the camera intrinsic matrix.

2) Extracting Skeleton from the Point Cloud: Once we obtain the point cloud of a tree, we extract its skeleton representation using the extraction process in [16]. In particular, we provide the point cloud of a tree $\mathbf{V}$ to the skeleton extraction algorithm F and obtain the skeleton nodes S as well as their interconnections C as shown in Fig. 1(c).

$$[S, C] = F(\mathbf{V}) \tag{6}$$

3) Finding Neighboring Points of a Skeleton Segment: After obtaining the skeleton of the tree, we then find the n neighboring points $\mathbf{R} \subset \mathbf{V}$ of each skeleton-segment based on the nearest neighbor algorithm, as shown in Fig. 1(d). We use σ parameter (discussed in Section IV-A) to filter the skeleton segments that have n number of points lower than σ (i.e., 15).

4) Finding Dominant Direction of Neighboring Points: We estimate the orientation of a segment by calculating the dominant direction $\mathbf{d}$ from its neighboring points. To obtain the dominant direction $\mathbf{d}$, we apply the Principal Component Analysis (PCA) on the clustered points of a segment $\mathbf{R}$ using the following equation:

$$\mathbf{R}\mathbf{R}^T\mathbf{w} = \lambda\mathbf{w} \tag{7}$$

$$\mathbf{d} = [\mathbf{w}_{i1}], i = 1, 2, 3 \tag{8}$$

Where λ and $\mathbf{w}$ represent the eigenvalues and the eigenvectors, respectively.

5) Projecting Neighbouring Points onto Cross-Sections: In this section, we aim to form cross-sections from the neighnoring points $\mathbf{R}$. In this direction, the first step is to project each point in $\mathbf{R}$ on to the dominant direction $\mathbf{d}$ by applying the dot product of each point with $\mathbf{d}$. Once we project the points of a cross-section $\mathbf{R}$ on to $\mathbf{d}$, we sort and index each point based on its distance from the origin along $\mathbf{d}$, as shown in Fig. 1(e). Then, we combine the n indexed points into several sets. We use the γ parameter with a value of 8 (as discussed in Section IV-A) to determine the number of points in each set. After that, we project the points of each set on to a plane (i.e.,a cross-section). To project a point (x, y, z) on a plane that is defined by the position $(\mathbf{a} = (j, k, l))$ and the normal vector $(\mathbf{d} = (s, t, u))$, we use the following (x+Fs, y+Ft, z+Fu) where,

$$F = \frac{sj - sx + tk - ty + ul - uz}{s^2 + t^2 + u^2} \tag{9}$$

Note that we denote the projection of γ number of points onto a cross-section as $\mathbf{U}$ from now on. Fig. 1(f) illustrates the projection of the neighboring points $\mathbf{R}$ of a segment (shown in red) into eight cross-sections (shown in black). Note that the number of cross-sections or planes p is adaptive as it depends on both γ and the number of points n in $\mathbf{R}$. The relationship between n, p, and γ is:

$$p = \lfloor n/\gamma \rfloor, \tag{10}$$

6) Fitting a Semi-Circle on a Cross-Section: The final step is to fit semicircles on cross-sections of tree point cloud. For this purpose, we first find the observation vector ν of the camera which is approximated to pass through the mean μ of the points of a cross-section $\mathbf{U}$. Therefore,

$$\nu = \frac{\mu}{|\mu|} \tag{11}$$

After obtaining $\mathbf{U}$, μ, and ν, we fit a semicircle on the points of a cross-section $\mathbf{U}$. Specifically, we estimate the semi-circle parameters — its radius $\mathbf{r}$ and center $\mathbf{C}$. To do this, we calculate the diameter D_C of a cross-section based on its ν vector. To calculate the diameter D_C of a cross-section, we first find the farthest pair of points in $\mathbf{U}$, calculate their respective perpendicular distances $d_{\perp_1}$ and $d_{\perp_2}$ from ν, and sum these distances. Fig. 5 illustrates the two perpendicular distances $d_{\perp_1}$ and $d_{\perp_2}$. To calculate $d_{\perp_x}$ between the ν and a point $\mathbf{x} \in \mathbf{U}$ we use the following:

$$d_{\perp_x} = \frac{|\mathbf{W} \times \mathbf{V}|}{|\mathbf{W}|} \tag{12}$$

$$\mathbf{W} = -\nu \tag{13}$$

$$\mathbf{V} = \nu - \mathbf{x} \tag{14}$$

Once we obtain the calculated diameter D_C we correct it based on the proposed model discussed in Section III-A. In particular, we estimate the value of actual diameter D_A with depth value $d = |\mu|$ using (2) and obtain the radius $\mathbf{r}$ as $\mathbf{r} = D_A/2$.

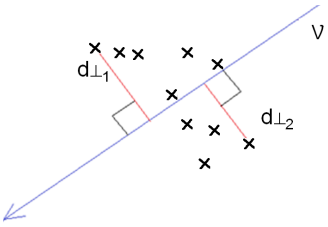


Fig. 5: The perpendicular distances $d_{\perp_1}$ and $d_{\perp_2}$ between farthest pair of points and observation vector ν is shown.

To find the optimal center $\mathbf{C}$ of the semicircle, we start by finding an initial estimate C_0 as:

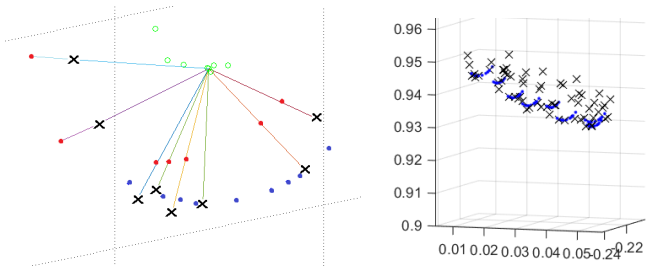
$$C_0 = \mu + \mathbf{r}\nu \tag{15}$$

Next, we construct a new set of points $\mathbf{z}_i \in \mathbf{Z}$ (marked as red circles in Fig. 6(a)) w.r.t. the original points of a cross-section $\mathbf{U}$ (marked in black crosses in Fig.6(a)) using both the center $\mathbf{C}_0$, and the radius $\mathbf{r}$ parameters of the cross-section, as follows:

$$\mathbf{z}_i = \mathbf{C} - \mathbf{r}\mathbf{D}_i \tag{16}$$

Where i iterates over all points in a cross-section, $\mathbf{D}_i$ is a direction vector from $\mathbf{C}_0$ to every point $\mathbf{x}_i \in \mathbf{U}$. To estimate the optimized center parameter $\mathbf{C}_f$, we use Levenberg-Marquardt [6] to minimize the distance between points in $\mathbf{Z}$ and points in $\mathbf{U}$. The distance function is:

$$\sum_i (\mathbf{x}_i - \mathbf{z}_i)^2 \tag{17}$$



(a) (b)

Fig. 6: We show the points at one cross-section and at one segment of the tree. In (a), we show points at a single cross-section. The projected points on the plane or cross-section are shown with black crosses. The red and blue points represent the initial estimate of points fitted on the data and final semicircle arc, respectively. The green circles are shown to illustrate the centers estimated while fitting procedure. In (b), we show the points at one segment of the tree. The black crosses are the projected points on planes while blue points are final semicircle points.

Note that $\mathbf{C}$ is the only parameter that is estimated using the Levenberg-Marquardt algorithm as $\mathbf{r}$ is estimated earlier using (2). Once we estimate the optimal $\mathbf{C}_f$, we use it along with $\mathbf{d}$, v, and $\mathbf{r}$ to obtain the optimized semicircle points (marked in blue points in Fig. 6(a)) using Rodrigues rotation formula [20] with an arc angle α that ranges from $-\pi/2$ to $\pi/2$. Fig. 6 shows the points at a cross-section in (a) and the points that belong to a segment in (b).

IV. Experimental Setup

In this section, we will first describe the important parameters of our approach together with their recommended tuning. Next, we will explain the data acquisition procedure used to collect our dataset of dormant apple trees. Finally, we will describe the metrics used for evaluating the performance of the proposed framework.

A. Parameter tuning

In our approach, we have three main parameters that affect the reconstruction results. These parameter are (i) Depth threshold (τ), (ii) Number of points around an edge (σ), and (iii) Number of points in a cross-section (γ).

In Section III-B we mentioned that it is required to set a bounding box on the object that is being reconstructed. We use the parameter τ to define the maximum depth beyond which all the depth values in the depth image are set to zero. The value of τ should be altered depending upon the position of the object from the sensor. In our experiments, we set the value of τ to 2m.

As discussed in Section III-C, the skeletonization procedure results in the existence of some redundant edges that are surrounded by few sparse points. Consequently, we need to threshold the least number of points around an edge that is determined by the parameter σ. In our experiments, we set the value of σ to 15. After identifying a candidate edge (i.e., less than σ) we partition its surrounding points into several cross-sections. We use the γ parameter to identify the number of points required to form a cross-section. In our experiments, we use γ values that range from 5 to 8.

B. Dataset and its Acquisition

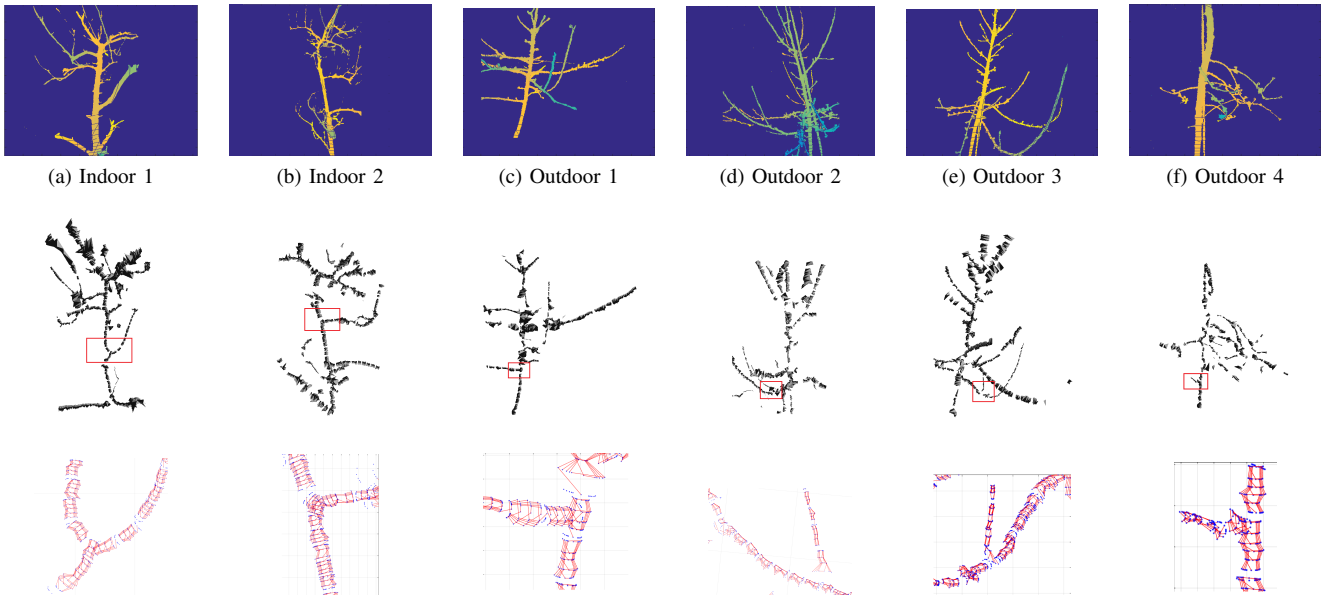
We collect depth images of a tree at a very high frame rate while moving Kinect 2 sensor around the tree. The images are collected from both far and near viewpoints. The far viewpoint images capture the entire tree while near viewpoint images encapsulate the finer details of branches of the tree. It is important to mention that in a high-density orchard it is not possible to go around the tree while scanning because of the close proximity of the trees. Therefore, we collect data from front and back sides of the tree and store them, separately. In particular, there are six trees in our dataset that are collected from different orchards. Out of these six trees, there are two indoor laboratory trees namely, 'Indoor 1', and 'Indoor 2' while the rest are obtained from the field, namely, 'Outdoor 1', 'Outdoor 2', 'Outdoor 3' and 'Outdoor 4'. For each tree, we collect three types of information listed below.

Depth images: We capture the depth images from both front and back sides of a tree. Six depth images are shown in the first row of Fig. 7. **Labeled images**: We label the primary branches of a tree from bottom to top as shown is Fig. 8 for verifying the obtained reconstructions of our approach. **Ground truth measurements**: We use the ground truth measurement of the diameters for evaluating our approach. Therefore, we collect the diameter values of the primary branches of a tree using caliper from bottom to top. In third column of Table I, we show the ground truth values of the examined trees.

Fig. 8: Ground truth image with labeled branches is shown. The labels '1' to '9' are clearly shown at the originating location of the respective branches.

C. Evaluation Metric

We use two metrics to evaluate our approach, namely, *Estimation Error* and *Confidence Value*. The *Estimation Error* is calculated based on the absolute difference between the ground truth diameter (obtained using caliper) and the estimated diameter (obtained using our approach). The *Confidence Value* indicates the percentage of the primary branches whose *Estimation Error* are within a certain threshold value ε and is given as follows:



(a) Indoor 1 (b) Indoor 2 (c) Outdoor 1 (d) Outdoor 2 (e) Outdoor 3 (f) Outdoor 4

Fig. 7: The first row shows the depth images of trees 'Indoor 1', 'Indoor 2', 'Outdoor 1', 'Outdoor 2','Outdoor 3', and 'Outdoor 4', respectively. In the second row, we show the reconstruction of each tree and highlight a selected region with a red box. Third row shows the magnification of the selected region of each tree.

$$Confidence\ Value = \frac{P}{Q} * 100 \tag{18}$$

Where P is the number of branches with *Estimation Error* less than ε and Q is the total number of branches.

V. Experimental Results and Evaluation

In this section, we show the obtained reconstruction results for each tree in our dataset. In particular, Fig. 7 demonstrates for each tree, its depth image (in the first row), obtained reconstruction (in the second row) using a single depth image, and a zoomed-in reconstruction for a highlighted region (in the third row).

Table I shows a detailed evaluation of each tree. Specifically, the first and second columns represent the tree name and its primary branches that are evaluated, respectively. The third column shows the measured ground truth diameter-value (using caliber) for each primary branch, while the fourth column shows the estimated diameter using our 3D reconstruction model. Finally, the fifth and sixth columns demonstrate the absolute and the percentage errors between the estimated diameter and its corresponding ground truth measurement, respectively.

To calculate the accuracy of our approach, we also use the *Confidence Value* metric described in Section IV-C. In particular, we calculate the percentage of the correctly estimated diameters for all the primary branches based on various ε threshold values. Using ε values of $2mm$, $3mm$, and $5mm$, we were able to correctly estimate the diameters of $50\%, 67\%$, and 89% of primary branches, respectively.

TABLE I: Performance evaluation of proposed framework, see text for details

Tree name	Branch number	Actual diameter (mm)	Estimated diameter (mm)	Estimation Error (mm)	Percentage Error (%)
Indoor 1	1	25.0	20.0	5.0	20.0
	9	15.0	18.0	3.0	20.0
	12	18.0	16.5	1.5	8.3
Indoor 2	11	16.0	14.5	1.5	9.3
	8	20.0	19.1	0.9	4.5
	12	13.0	14.7	1.7	13.0
Outdoor 1	1	13.3	14.6	1.3	9.0
	5	11.0	13.8	2.8	25.4
	10	10.9	8.0	2.9	26.6
Outdoor 2	1	13.8	12.7	1.1	7.9
	3	11.6	15.4	3.8	32.7
	8	10.5	9.6	0.9	8.5
Outdoor 3	1	13.8	15.4	1.6	11.5
	4	9.4	12.8	3.4	36.1
	8	18.0	26.0	8.0	44.4
Outdoor 4	1	5.8	9.6	3.8	65.5
	8	7.9	10.3	2.8	35.4
	10	9.9	10.8	0.9	9.0

VI. Conclusion

In this work, we presented a new method that accurately reconstructs dormant (i.e., without foliage) apple trees using a single depth image. This method is based on the least-squares fitting of semicircles on cross-sections of the tree point cloud. The proposed approach allows us to estimate the diameter of a primary branch using a novel empirical model, which is useful for dormant pruning. We also demonstrated both qualitative and quantitative results on several trees under various indoor and outdoor environments. The obtained

results allowed us to accurately estimate the diameters of 89% of primary branches with ε value of $5mm$.

For future work we plan to extend our empirical error model in order to account for possible variations in both illumination and primary branch thickness as well as improving the robustness of the proposed diameter-error model (i.e., improving the R^2 value from $\sim$0.4 to higher values).

ACKNOWLEDGMENT

We would like to acknowledge the United State Department of Agriculture (USDA) that provides the funding of this research. We would also like to thank Professor Peter Hirst's group from the Department of Horticulture and Landscape Architecture at Purdue University, Professor James Schupp's group from the Department of Horticulture at Pennsylvania State University, as well as the Bear Mountain Orchards at Aspers, PA, for their help in the tree scanning process and the development of the dataset. We would also like to acknowledge Mr. Bret Wallach from Vision Robotics Corporation for his valuable advises in the course of this research.

REFERENCES

[1] C. Forshey and D. Elfving (1989) *The Relationship Between Vegetative Growth and Fruiting in Apple Trees*, in Horticultural Reviews, Volume 11 (ed J. Janick), John Wiley & Sons, Inc., Hoboken, NJ, USA. doi: 10.1002/9781118060841.ch7

[2] C. Forshey, D. Elfving, and R. Stebbins, *Training and pruning apple and pear trees*, American Society for Horticultural Science, Alexandria, Virginia, 1992, ISBN 0-9615027-1-1

[3] B. Adhikari and M. Karkee. "3d reconstruction of apple trees for mechanical pruning". In 2011 ASABE International Meeting, August 2011.

[4] M. Karkee, B. Adhikari, S. Amatya, and Q. Zhang, "Identification of pruning branches in tall spindle apple trees for automated pruning", Computers and Electronics in Agriculture, 103(0):127 135, 2014.

[5] Q. Wang and Q. Zhang, "Three-dimensional reconstruction of a dormant tree using rgb-d cameras", In 2013 ASABE International Meeting, July 2013.

[6] D. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters", SIAM Journal on Applied Mathematics 11 (2), 1963, 431441.

[7] S. Izadi, D. Kim, O. Hilliges, D.Molyneaux. R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison. and A. Fitzgibbon, "KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera", in Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, New York, NY, USA, 2011, pp559-568

[8] T. Whelan , M. Kaess , M. Fallon , H. Johannsson , J. Leonard and J. McDonald, "Kintinuous: Spatially extended KinectFusion", Proc. Workshop RGB-D, Adv.Reason. Depth Cameras, 2012

[9] H. Roth and M. Vona, "Moving volume KinectFusion", Proc. Brit. Mach. VisionConf., pp.1 -11 2012

[10] C. Nguyen, S. Izadi, and D. Lovell, "Modeling Kinect Sensor Noise for Improved 3D Reconstruction and Tracking". 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission (3DIM/3DPVT), pages 524-530, Oct. 2012. 3, 5

[11] J. Smisek, M. Jancosek, and T. Pajdla, "3D with Kinect", in ICCV Workshop on Consumer Depth Cameras for Computer Vision, 2011.

[12] T. Mallick, P. Pratim Das and A. K. Majumdar, "Characterizations of Noise in Kinect Depth Images: A Review", IEEE Sensors Journal, vol. 14, no. 6, 2014, pp. 1731-1740.

[13] S. Meister , S. Izadi , P. Kohli , M. Haemmerle , C. Rother and D. Kondermann, "When can we use KinectFusion for ground truthacquisition?", Proc. Workshop Color-Depth Camera Fusion Robot., 2012

[14] Y. Livny , F. Yan , M. Olson , B. Chen , H. Zhang , J. El-Sana, "Automatic reconstruction of tree skeletal structures from point clouds", ACM Transactions on Graphics (TOG), v.29 n.6, December 2010

[15] N. Elfiky, S. Akbar, J. Sun, J. Park, and A. Kak,"Automation of Dormant Pruning in Specialty Crop Production: An Adaptive Framework for Automatic Reconstruction and Modeling of Apple Trees", in Proc. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June, 2015

[16] J. Cao, A. Tagliassachi, M. Olson, H. Zhang, and Z. Su. "Point cloud skeletons via laplacian based contraction. Proc. IEEE Int. Conf. on Shape Modeling and Applications", 187197, 2010.

[17] A. Staranowicz, G. Brown, F. Morbidi, and G. Mariottini, "Easy-to-use and accurate calibration of rgb-d cameras from spheres", In Image and Video Technology. Springer, 2014, 265–278.

[18] S Jin, L. Fan, Q. Liu, and R. Lu , "Novel Calibration and Lens Distortion Correction of 3D Reconstruction Systems", International Symposium on Instrumentation Science and Technology, Institute of Physics Publishing, 2006.

[19] R. Szeliski, "Image Alignment and Stitching: A Tutorial", Found. Trends. Comput. Graph. Vis., Now Publishers Inc., Hanover, MA, USA, January 2006

[20] http://www.mathworks.com/matlabcentral/fileexchange/34426-rotate-vectors-about-axis/content/rodrigues_rot.m