

A Novel Visualization Tool for Evaluating the Accuracy of 3D Sensing and Reconstruction Algorithms for Automatic Dormant Pruning Applications

Fangda Li Somrita Chattopadhyay Shayan A. Akbar Noha M. Elfiky Avinash Kak
School of Electrical and Computer Engineering, Purdue University

li1208@purdue.edu, chattops@purdue.edu, sakbar@purdue.edu, nelfiky@purdue.edu, kak@purdue.edu

Abstract

Agricultural tree pruning is one of the most essential procedures in specialty crop production. During winter, skilled workers have to remove certain primary branches of the dormant trees in order to increase the productivity of those trees. The process is particularly challenging as it needs huge workforce and costs, besides, seasonal availability of skilled workers. In this paper, we introduce a novel multifunctional graphical user interface which operates in two different modes, namely, manual and automatic, and serves the following purposes - (1) validating 3D reconstruction algorithms, (2) automating the decision making process for selecting candidate branches to be pruned, (3) providing an easy interface with any robotic pruners, (4) training unskilled seasonal pruners, and (5) disseminating the knowledge. The paper describes the operating modes of the proposed tool and highlights various features and functionalities of the software by conducting experiments on an exemplar apple tree.

1. Introduction

Dormant pruning is the process of removing certain unproductive primary branches (i.e., branches that are directly connected to the trunk) of unfoliated trees in order to give significant boost to the commercial yield of an orchard in the upcoming production cycle. In modern high-density apple orchards, a large number of trained seasonal workers are employed to prune the trees manually during winter. The huge labor cost, unavailability of skilled seasonal pruners, and inconvenience to train them in physical orchards during winter have made dormant pruning one of the most expensive and challenging tasks in specialty crop production.

To alleviate dependence on this huge workforce, many researchers nowadays are focusing on automating the process of pruning. Our end goal being easy interface with robotic pruning, one important step is to build accurate 3D models of the trees. 3D models are needed to automate the

decision making process of selecting the candidate branches to be pruned. Moreover, quantitative pruning algorithms might produce different pruning decisions as the ones from human experts; thus, accurate 3D models can be used to visually and quantitatively evaluate the pruning decisions before actually applying them using real robots.

In order to address all the aforementioned issues, our paper introduces a novel multifunctional visualization software to perform pruning simulations on reconstructed apple trees. Our aim is to design a tool that (1) evaluates different 3D tree modeling algorithms, (2) makes automatic decisions of selecting candidate branches to be pruned, (3) provides easy interface with robotic pruning, (4) creates virtual pruning environment to train unskilled workers, and (5) serves as an educating and knowledge disseminating tool in classrooms, conferences, and other interested domains. We have developed two different operating modes for the software — a manual mode for evaluating, training, and educating purposes, and an automatic mode for making automatic pruning decisions and candidate branch selection to achieve our end goal of easy interface with robotic simulation.

The rest of the paper is organized as follows. In Section 2, we discuss the previous works in the areas of 3D tree reconstruction and visualization. Then, in Section 3, we explain the preprocessing procedures and implementation details of our 3D reconstruction technique. We introduce our visualization software for pruning simulation and provide detailed description of all its features in Section 4. In Section 5, we demonstrate the experiments performed using our software. Finally, we conclude our paper by highlighting few potential future works in Section 6.

2. Related Work

This section provides an overview of the current state-of-the-art in the field of 3D tree reconstruction, modeling, and visualization.

3D scene reconstruction has been an active area of research over the past few decades. Recently, reconstruction and modeling of trees has gained attention of the

computer vision scientists as an emerging research area due to its applications in robotics precision agriculture. In order to create the 3D model of a tree, existing literature [3, 12, 11, 7, 2, 4] use laser scanners or *Kinect* sensors to obtain depth images of the tree, and convert them into point clouds. Afterwards, the point clouds are aligned to form the final 3D model. Although acceptable results are obtained from scanning only one side of the tree in [3, 12, 11], their results lack to provide useful information about the branches that are present on the other side of tree. For automated pruning applications, having quality reconstruction of branches on both front and back sides of the tree is critical. Therefore, in this paper we use the reconstruction method suggested by Elfiky et al. in [7], where front and back point clouds of a tree are collected using the *KinectFusion* software [13], and merged to obtain final reconstruction of the entire tree.

Traditionally, state-of-the-art methods have represented the tree using either a graph structure or 3D geometric primitives. In [12], the tree structure is modeled as a weighted acyclic graph, where the thickness of the edges are estimated by the weights of its incident nodes. While a stacked-circle model is used in [7] in which circles are fitted onto trunk and primary branches while taking into consideration the centers and radii of the the neighbors.

For automatic pruning simulation, a visually appealing and interactive tool is needed in order to convey relevant information (e.g., radius and angle) about each branch, as well as visualize the tree before and after pruning. Commercial tree modeling software, such as *SpeedTree* [10] and *The Grove* [16], mainly focus on creating realistic visual effects, and do not support importation of customized models. On the other hand, open source visualization software, such as *Meshlab* [1] and *Blender* [8], cannot associate a branch model with its radius and angle.

All of the aforementioned 3D reconstruction techniques and visualization tools somehow lack to bundle the tree and its model together for pruning simulation. Realizing the limitations of the currently available software, in this paper we propose a new tool which creates clean and integrated visualization for both the tree and its model, with easy and effective automatic pruning interaction.

3. Preprocessing and Implementation Details

This section discusses the preprocessing operations required for the development of our multipurpose software. In Section 3.1, we mention the software used for this visualization tool. In Section 3.2, we explain our technique for 3D reconstruction of a dormant apple tree. This 3D tree model is used as an input to our multifunctional GUI. We describe the quantitative pruning criteria used to design different features of our tool in Section 3.3. Finally in Section 3.4, we explain the tree data that we are using to illustrate

the handling of our software.

3.1. Multifunctional GUI

The graphical user interface (GUI) of our software is written in C++ using *Qt* [5], which guarantees it to be compatible with both *Windows* as well as *Linux* operating systems. The software uses *OpenGL* [9] to render the 3D mesh of the tree. The *OpenGL*-based implementation utilizes the standard GPU rendering pipeline as mentioned in [14]. In this pipeline, the vertices from the mesh are first assembled into triangle primitives, which are then rasterized into fragments. Finally, the fragments are colored individually and stored as pixels in the screen buffer.

To enhance visual realism, diffused texturing and lighting effects are applied on the tree. Cube mapping is utilized to show a seamless color gradient in the background. To highlight branch contours in different colors, different stencil buffer values are used. Furthermore, we implement the technique described in [6] to render shadow of the tree on the ground plane.

3.2. 3D Reconstruction and Modeling

As input to our software, we create a 3D model of the dormant apple tree using a stacked-circle based 3D reconstruction scheme, as suggested by Elfiky et al. in [7].

A 360° model of the tree is reconstructed from only 2 views (i.e., front and back) of the tree. The method generates front and back point clouds of the tree from its corresponding depth images, and the final reconstruction is obtained by merging these two point clouds into a single 3D point cloud by exploiting various geometric attributes — such as angle of the branch with the trunk, and distances of the branch node to the root node and trunk node, extracted from the skeleton structure of the tree. An initial transformation estimate is first computed based on the mutual correspondence between the geometric attributes from front and back sides, and then the Iterative Closest Point (ICP) algorithm is used for further refinement.

The algorithm also exploits the fact that branches usually have inherent cylindrical structure, and employs an adaptive layered-circle-based modeling technique. Once reconstruction is done, modeling starts by dividing the point cloud into several layers and applying RANSAC based circle fitting to those points that are in a plane parallel to the ground plane. However, since the point cloud data includes points from both the trunk as well as branches, diameters of fitted circles differ greatly, even between adjacent layers. Thus, the algorithm implements an adaptive approach such that the points used to form a new circle stay within a bounding window of the neighboring circles.

The primary branches are also modeled in a similar fashion. Cylinders are first fitted onto each primary branch point cloud using RANSAC algorithm to determine the growing

direction of the branch. Each branch is then rotated towards the z -axis, and the same layered circle fitting algorithm as before is applied to find the stacked-circle model for each primary branch. These circles are then rotated to align with the growing direction of the branch.

At the end of preprocessing stage, the final 3D model comprises of the trunk and the primary branches modeled as sets of stacked-circles, each circle being represented by its coordinates and diameter.

3.3. Pruning by the Numbers

With our end goal being easy interface with robotic pruners, here we use the ‘‘Pruning by the Numbers’’ approach described in [15] to automate the pruning decisions using numerical evaluations rather than the traditional way of visual inspection. The article also introduces two different pruning algorithms and a novel evaluation metric called ‘‘pruning severity’’. Furthermore, quantitative parameters of the branches such as radii and angles are considered while making a pruning decision. The diameter of a primary branch is taken equal to the diameter of the first circle in its stacked-circle model, and the branch-to-trunk angle is obtained by calculating the angle between the normal direction \mathbf{N} of the concerned branch obtained from its cylinder model and the direction \mathbf{d} of the trunk at that location. The equation for angle θ calculation is given below:

$$\theta = \arccos \left(\frac{\mathbf{d} \cdot \mathbf{N}}{\|\mathbf{d}\| \|\mathbf{N}\|} \right) \quad (1)$$

In order to remain consistent with the terminologies used in [15], the terms *branch* and *limb* are used interchangeably in the rest of the paper.

Pruning Severity

This section describes the pruning severity metric for an apple tree. A high value of this metric indicates that a large number of branches should be pruned. Severity is quantified by the **Limb-to-Trunk Ratio (LTR)**, which is defined as:

$$\text{LTR} = \frac{\sum_{i=0}^n \text{LCSA}_i}{\text{TCSA}} \quad (2)$$

Where n is the total number of branches of the tree, LCSA_i is the **Limb Cross-Sectional Area** of the branch i , and TCSA is the **Trunk Cross-Sectional Area**. LCSA of each branch is measured at approximately 2.5 cm from its union to the trunk, and the TCSA is measured at about 30 cm above the ground.

Next we explain the two severity-based pruning algorithms: **Limb Trunk Ratio (LTR)** and **Maximum Limb Diameter (MLD)**, described in [15]. In **LTR**, branches of the tree are successively pruned starting from the thickest branch and continuing until the pruning severity or the **LTR**

value of the tree drops below a certain threshold. The value of this threshold is obtained empirically to produce maximum yield per tree as well as optimal fruit size.

The use of **LTR** algorithm is not feasible in case of manual pruning because of the impracticality to measure *all* the branches of *all* the trees in the orchard, and sort them in decreasing order of their diameters. Although the branch diameter rankings can be stored easily in automated pruning, it is inconvenient for human pruners to keep a record. Therefore, **MLD** algorithm is used as an approximation of the **LTR** algorithm, where sorting of branches with respect to their diameter values is not required. Rather the branches are pruned based on a simple comparison of their diameters with the maximum diameter **MD** value. In particular, we first apply the **LTR** algorithm on *few* trees within the same orchard, and as an additional step of the **LTR** algorithm, the diameter of the largest remaining limb for each tree is recorded. Then, we regress the largest diameter out of these remaining limbs against the trunk circumference **TC** and the desired **LTR** value to obtain Equation 3.

$$\text{MD} = 0.87 + 0.553 \text{TC} + 4.29 \text{LTR} \quad (3)$$

Where **MD** is the maximum branch diameter, **TC** is the trunk circumference measured in centimeters at about 30 cm above the ground, and **LTR** is the desired limb-to-trunk ratio.

3.4. Tree Data

We have illustrated the handling of our multifunctional software by conducting experiments on a dormant stand-alone apple tree, named ‘**Indoor 1**’, from our dataset¹. The images are acquired in an indoor laboratory setting using a Kinect2 sensor. We have collected the following different information for the tree. **Depth images**: The tree is scanned from front and back to yield two different views. **Labeled images**: We label the primary branches of the tree from bottom to top using a color-coded pattern for verifying the obtained reconstruction results of our approach explained in Section 3.2. **Ground truth measurements**: As ground truth, we collect the diameter values of the primary branches of a tree using a caliper from bottom to top. We also divide the tree into several sections starting from the root for ground truth measurements.

We apply the method explained in Section 3.2 on the above images to create a 3D tree model, consisting of two different information — (a) **stacked circle model** for the trunk and each primary branch, and (b) a **mesh file** for the whole tree. The tree trunk model in stacked circle format contains circles stored in individual text files with corresponding 2D coordinates (assuming trunk to be vertical and

¹The dataset is now publicly available at https://engineering.purdue.edu/RVL/WACV_Dataset

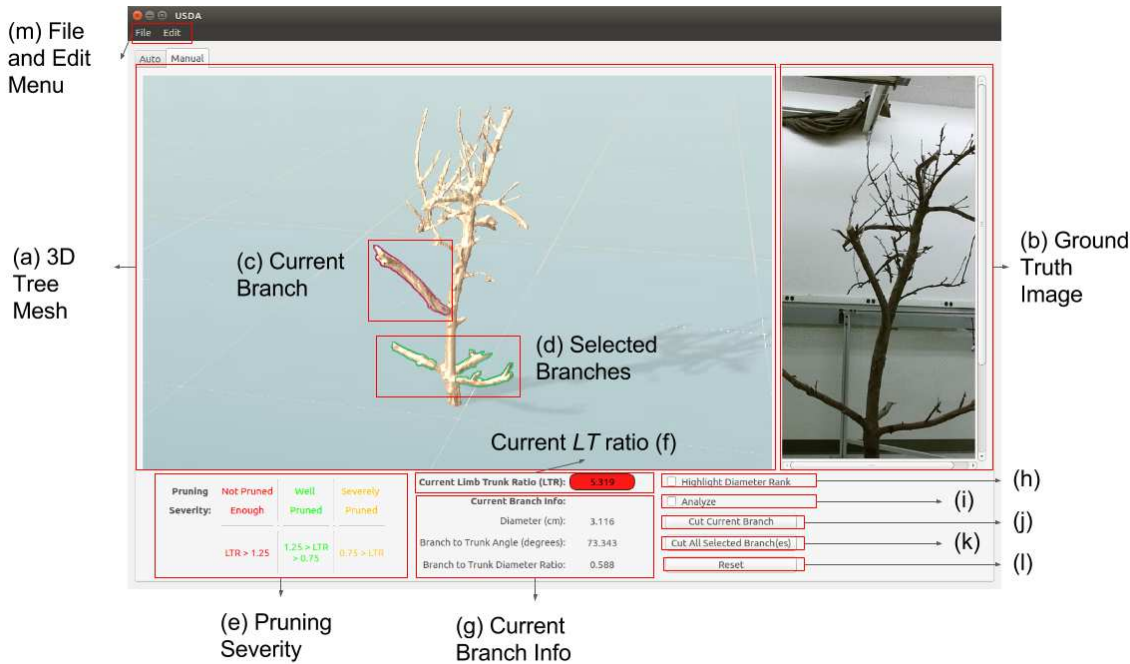


Figure 1: GUI for manual mode: (a) 3D Tree Mesh; (b) Ground Truth Image; (c) Current Branch Selected; (d) Multiple Branches Selected; (e) Pruning Severity Status; (f) Current **LTR** Value; (g) Current Branch Info contains information about Diameter, Branch to Trunk Angle Ratio, and Branch to Trunk Diameter Ratio; (h) Highlight Diameter Rank; (i) Analyze; (j) Cut Current Branch; (k) Cut All Selected Branch(es); (l) Reset; (m) File and Edit menu.

$z = 0$) and radii r . Each branch model contains circles stored in individual text files with their respective 3D coordinates and radii. Cylindrical model for each branch stored in a text file contains 3D coordinates of the lower circle, radius, height, and normal direction. In the mesh file, the whole tree (trunk and branch meshes) is stored in one object file, **.obj*. Each mesh in the object file contains the 3D coordinates of each vertex, the vertex indices of each face, and the normal associated with each face.

4. Multipurpose Graphical User Interface for Pruning Visualization

During winter season, pruners have to go to the orchards, and make the pruning decisions based on the appearance and position of the primary branches while measuring their parameters. To automate this pruning process, our paper introduces a new tool that can be used to accomplish the following tasks — (1) assess 3D sensing algorithms, (2) serve as a virtual training tool for unskilled pruners, preventing them from actually going to the orchard during winter for getting trained on real trees, (3) automate the decision making process of selecting candidate branches for pruning, (4) provide easy interface to robotic pruning, and (5) serve as an interactive knowledge disseminating tool for students and

other interested audience.

In this section, we provide a detailed description about our pruning software, and highlight the basic as well as advanced features associated with its two different operating modes, namely, manual and automatic. In the manual mode, the pruning decisions are made by human experts; whereas, in the automatic mode, the decisions are made based on some predefined quantitative criteria.

4.1. Functionalities of the GUI

Here we have listed the basic and advanced features of our novel software.

Basic Functionalities

- 1) **Loading a Tree:** The user can load a tree directory via the *Open Directory* dialog from the **File** menu, as shown in Fig. 1(m). The mesh files, the stacked-circle model, and the ground truth images of each tree are stored in a directory after preprocessing.
- 2) **Rotation, Zoom In and Zoom Out:** The user can rotate the 3D mesh of the tree, shown in Fig. 1(a), to get a 360° view. It can also be zoomed in and zoomed out according to observation needs.
- 3) **Automatic Synchronization between the 3D Model and the Real Tree:** In Fig. 1(b), we display the ground

truth image of a tree with the same camera rotation as the 3D model, displayed in Fig. 1(a). We capture ground truth images from 360° viewpoints in order to achieve synchronized views of the 3D model and the real tree.

- 4) **Selecting a Branch:** To select a branch as indicated in Fig. 1(c), the user has to click the branch of interest using a mouse button.
- 5) **Selecting Multiple Branches:** Fig. 1(d) illustrates that the user can also select more than one branch at the same time, with the last selected branch being the current branch.
- 6) **Analyzing a Branch:** The user can turn on the **Analyze** switch, shown in Fig. 1(i), to display only the selected branches in the 3D mesh of the tree. This functionality is useful for observing the occluded branches, as shown in Fig. 2.
- 7) **Extracting Information of the Selected Branch:** The user can find the parameters of the current branch from the **Current Branch Info** section, shown in Fig. 1(g).
- 8) **Cutting a Branch:** Once selected, the current branch can be pruned using the **Cut Current Branch** button, shown in Fig. 1(j).
- 9) **Cutting All Selected Branches:** All selected branches including the current branch can be pruned using the **Cut All Selected Branch(es)** button, shown in Fig. 1(k).
- 10) **Resetting the Tree Structure:** Fig. 1(l) displays the **Reset** button which can be used to restore all the pruned branches and retrieve the original model.
- 11) **Pruning Severity Status:**

- **Viewing Severity Status of the Tree:** In Fig. 1(e), we show the ranges for three different levels of pruning severity — **Not Pruned Enough**, **Well Pruned** and **Severely Pruned**. If necessary, the ranges can be adjusted using from the **Edit** menu, shown in Fig. 1(m).

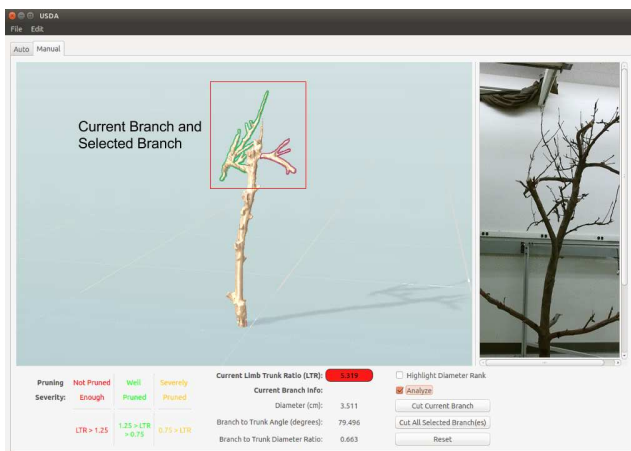


Figure 2: Use “Analyze” to show occluded branches

- **Extracting the Current Limb Trunk Ratio:** Fig. 1(f) gives the current **LTR**, calculated using only the remaining branches. The background color of the **LTR** button corresponds to the severity levels, as shown in 1(e). Red **LTR** background implies that the tree needs pruning, green **LTR** background means that the tree has been well pruned, and yellow **LTR** background implies that the tree has been over-pruned.

- 12) **Highlighting Branch Thicknesses:** The **Highlight Diameter Rank** switch, as shown in Fig.1(h), ranks all the branches in descending order of their diameter values, and applies color coding accordingly from red to green. If the **Highlight Diameter Rank** switch is checked, branches having different thicknesses are displayed with different colored contours. As shown in Fig. 4a, the thickest branch is highlighted with red contour, while green contour indicates the thinnest branch.

Advanced Functionalities

- 1) **Automatic Pruning Criteria:** The user here has choice of four different branch selection criteria:

- **Diameter:** The user can preset a range for branch diameters to select all the branches having diameters within that range, as shown in Fig. 3(a).
- **Branch to Trunk Angle:** This parameter indicates the angles between the trunk and each primary branch of the tree. As labeled in Fig. 3(b), the user needs to specify the minimum and maximum **branch to trunk angles** to select primary branches falling within that range.
- **Branch to Trunk Diameter Ratio:** This criterion is used to indicate the ratio between the diameters of the tree trunk and each primary branch. To select primary branches having certain **branch to trunk diameter ratio**, the ratio range needs to be prespecified by the user, shown in Fig. 3(c).
- **LTR after Pruning:** The user can set the desired **LTR** value after pruning (refer to Equation 2 for details) and one of the pruning algorithms discussed in Section 3.3 to select branches, as in Fig. 3(d).

- 2) **Combining Multiple Selection Criteria:** The user can choose multiple criteria for branch selection by turning on and turning off the **AND** switch, shown in Fig. 3(e). If checked, the switch takes the union of all selected criteria; whereas, the intersection of all selected criteria is considered when the switch is unchecked.

- 3) **Viewing Selected Branches:** The user can view the final selection of qualified branches by pressing the **Apply Criteria** button, shown in Fig. 3(f).

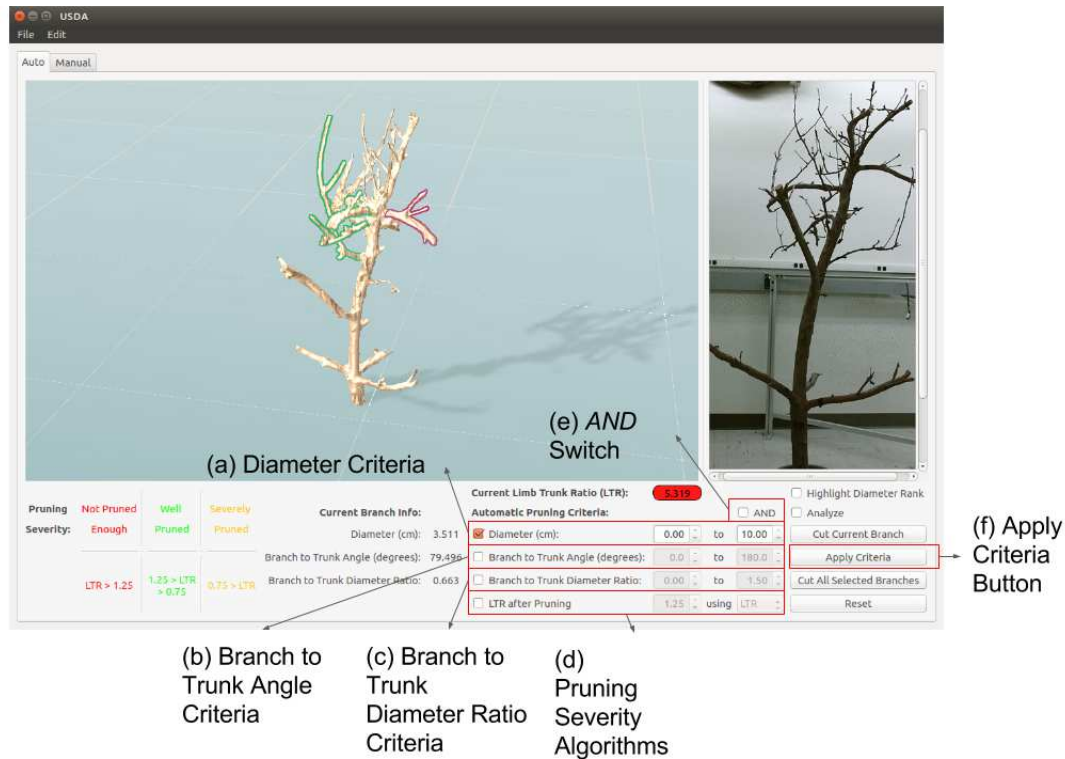


Figure 3: GUI for automatic mode. (a) Diameter; (b) Branch to Trunk Angle; (c) Branch to Trunk Diameter Ratio; (d) Pruning Severity Algorithms; (e) AND; (f) Apply.

4.2. Operation Modes of the GUI

The main goal of this GUI is to train unskilled pruners, evaluate different 3D reconstruction algorithms and perform automatic decision making to select candidate branches for creating an easy interface with robotic pruning, our end goal. In this section, we are going to describe two different operation modes of our software — manual and automatic.

Manual Mode

The manual mode is primarily designed to train unskilled workers to acquaint them with the entire pruning procedure, and to validate different reconstruction algorithms required to build 3D models of dormant trees. As mentioned before, this mode is operated based on intuition and visual inspection. The user makes pruning decision mainly based on the appearance and position of the branch while observing its quantitative parameters. The pruning process starts with loading a tree from the directory, and then selecting one or more of its branches by mouse clicking. Information about the selected branch/branches, such as diameter, branch to trunk angles, and branch to trunk diameter ratios, can be obtained from the **Current Branch Info** section. Human experts analyzing these information can decide whether or

not to prune the selected branches. One of most important criteria to prune the primary branches is the current **LTR** value. Color of the **LTR** background decides whether any more branches need to be pruned or not. Red **LTR** background indicates that the tree needs more pruning, green **LTR** background means the tree is properly pruned, and yellow **LTR** background shows that the tree has been over-pruned. Algorithm 1 provides a detailed illustration of the manual pruning procedure.

Algorithm 1 Manual Pruning Process

- 1) Load a tree.
 - 2) Observe the tree from different viewpoints. Zoom in to obtain a closer view.
 - 3) Hover on any branch to obtain its parameters.
 - 4) Select one branch or multiple branches.
 - 5) Use **Cut Current Branch/Cut All Selected Branch(es)** button for pruning.
 - 6) Examine current **LTR**.
 - 7) Repeat until desired values of **LTR** is reached.
 - 8) Restore the tree using **Reset** button, whenever needed.
-

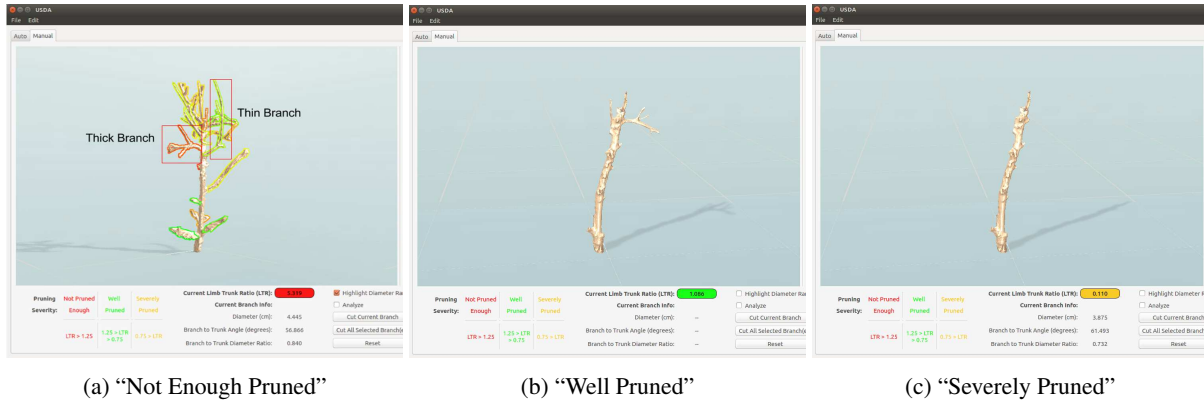


Figure 4: Pruning experiments in manual mode

Automatic Mode

The automatic mode is designed to automate the decision making process of selecting candidate branches for pruning, hence, facilitating easy interface with robotic pruning, the end goal of our process. In this mode, pruning is done based on some predefined criteria set by human experts. Similar to the manual mode, we start by loading a tree from the directory, and select one or more of its branches based on the criteria specified by the domain experts. User now has choice of 4 different pruning criteria — diameter, branch to trunk angle ratios, branch to trunk diameter ratios, and desired **LTR** value and pruning algorithm, as shown in Fig. 3. One or more of these criteria can be applied by checking the respective switches, and their immediate effect can be visualized by pruning the selected branches. Like manual mode, here as well the desired **LTR** value proves to be the most important criteria of pruning. Surveys have shown an optimal value of **LTR** to be 1.25 for quality growth and better yield of apple trees. Thus, the **LTR** criteria could be set to 1.25, and the effects of pruning can be observed using our GUI. A detailed procedure for the automatic pruning mode is given in Algorithm 2.

5. Experimental Results

In this section, we demonstrate the usage and effectiveness of our proposed visualization tool in two different operating modes — manual and automatic. All our experiments have been performed on a stand-alone dormant apple tree.

5.1. Manual Mode Experiments

Here we show the effect of manual pruning where the candidate branches are chosen primarily by appearance. The user can either choose the branches based on intuition, or can choose the option of locating the thick branches more efficiently by turning on the **Highlight Diameter Rank** switch. As shown in Fig. 4a, the thickest primary branch is indicated with red contour, while green contour specify

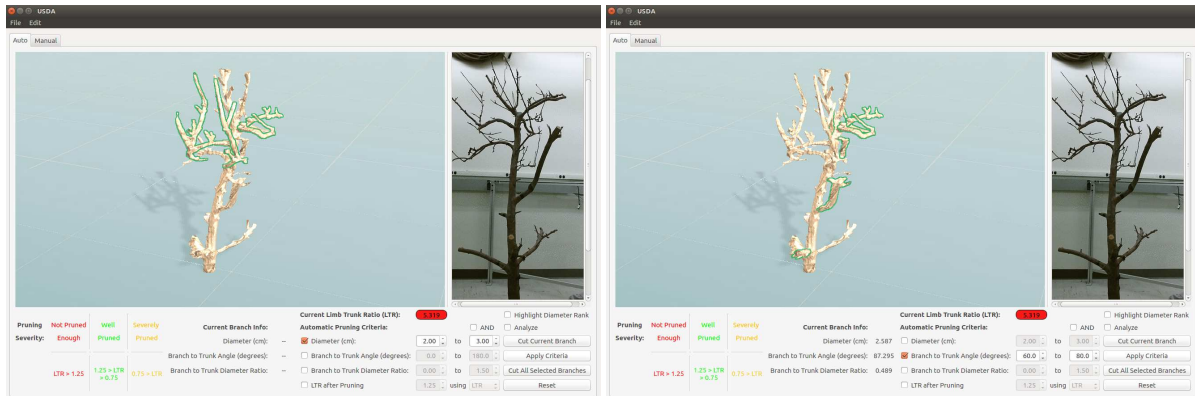
Algorithm 2 Automatic Pruning Process

- 1) Load a tree.
 - 2) Use specified criteria from domain experts.
 - 3) Check one or more pruning criteria from 4 different choices in the **Automatic Pruning Criteria** section and specify their ranges —(a) **Diameter**, (b) **Branch to Trunk Angle**, (c) **Branch to Trunk Diameter Ratio**, and (d) desired **LTR** and **severity pruning algorithm**.
 - 4) Visualize the branches selected for pruning by using the **Apply Criteria** button.
 - 5) Cut the selected branches using the **Cut All Selected Branch(es)** button to see the effect of pruning.
 - 6) Restore the original tree using **Reset** button, whenever needed.
-

the thinnest branch. Once all the candidate branches are selected, the user can start pruning the tree using the **Cut Current Branch** and the **Cut All Selected Branch(es)** buttons. It is observed that with increasing number of pruned branches, the **Current Limb Trunk Ratio** decreases. At the very beginning when none of the branches are pruned and **LTR** is very high, the red background in Fig. 4a indicates that the tree is not pruned enough. After pruning a certain number of branches, the **LTR** background turns green displaying a lower **LTR** value and indicating the tree to be sufficiently pruned, as shown in Fig. 4b. If the user continues pruning, the tree becomes severely pruned, indicated by a yellow **LTR** background in Fig. 4c.

5.2. Automatic Mode Experiments

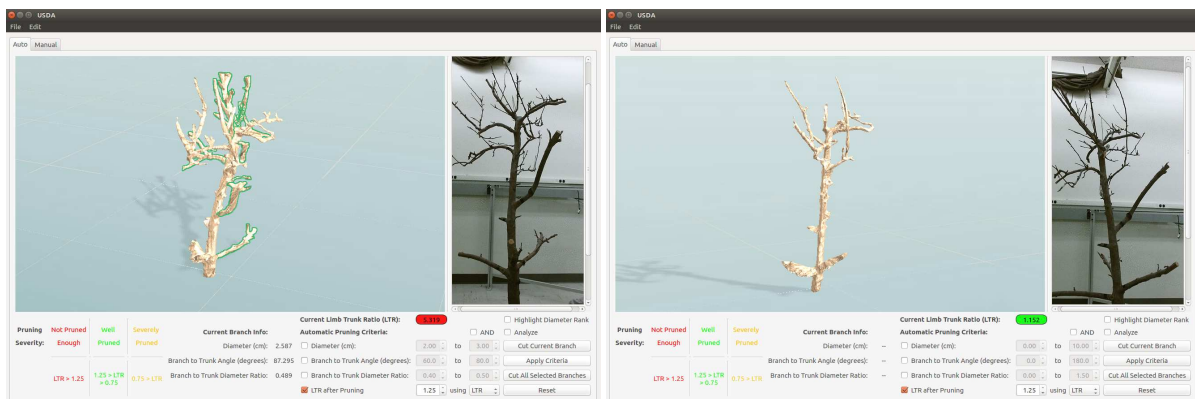
This section demonstrates the results of automatic pruning mode, in which the pruning decisions are made based on some predefined criteria by the domain experts. In Fig. 5a and Fig. 5b, we show that the branches can be selected based on diameter or branch to trunk angle criteria within



(a) Branches selected by diameter criteria (2.00 – 3.00 cm)

(b) Branches selected by angle criteria (60° – 80°)

Figure 5: Diameter and angle criteria in automatic mode



(a) Branches selected by LTR (*Desired LT Ratio* = 1.25)

(b) Pruning results of LTR (*Current LT Ratio* = 1.15)

Figure 6: Severity pruning in automatic mode

the user-specified range. While in Fig. 6a and Fig. 6b, the tree, before and after applying **LTR** algorithm, is displayed. It is seen that the **Current Limb Trunk Ratio** value after pruning is successfully lowered to the user-specified desired **LT** ratio, indicated by the **LTR** background color change from red to green.

6. Conclusion and Future Work

In this paper, we develop a tool for assisting the visualization, training and assessment of dormant apple tree pruning by incorporating 3D tree sensing and reconstruction. The tool is designed to serve multiple purposes — such as training amateur pruners, assessing different 3D sensing/reconstruction algorithms, automating the decision making process for candidate branch selection, and educating purposes. For the aforementioned applications, we have designed two different modes of our software — manual mode (mainly for training, educating and evaluating purposes) and automatic mode (mainly for automation of pruning decisions to facilitate easy interface with robotic

agents). The handling of the tool has been illustrated in details by performing experiments on a standalone dormant apple tree in an indoor laboratory setting.

In the future, we plan to build a 3D aligned and mesh generation framework instead of using KinectFusion software as mentioned in [7]. Also, in the current framework, the 3D meshes of the branches are segmented manually which needs to be automated. Moreover, we plan to add extra features to our software, such as computing locations of branch pruning points, incorporating a robotic plugin to simulate the pruning process, and many more.

7. Acknowledgment

This research is funded by the United State Department of Agriculture. We would like to thank Prof. Peter Hirst's group from the Department of Horticulture at Purdue University, Prof. James Schupp's group from the Department of Horticulture at Pennsylvania State University, and the Bear Mountain Orchards at Aspers, PA, for their help in the tree scanning process.

References

- [1] MeshLab. <http://meshlab.sourceforge.net/>. [Online; accessed 02-03-2016].
- [2] S. Akbar, N. Elfiky, and A. Kak. A novel framework for modeling dormant apple trees using single depth image for robotic pruning application. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
- [3] J. Binney and G. S. Sukhatme. 3d tree reconstruction from laser range data. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 1321–1326. IEEE, 2009.
- [4] S. Chattopadhyay, S. Akbar, N. Elfiky, H. Medeiros, and A. Kak. Measuring and modeling apple trees using time-of-flight data for automation of dormant pruning applications. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.
- [5] T. Q. Company. Qt. <http://www.qt.io/>. [Online; accessed 02-09-2016].
- [6] J. de Vries. Shadow Mapping. <http://learnopengl.com/>. [Online; accessed 02-09-2016].
- [7] N. M. Elfiky, S. A. 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 *Computer Vision and Pattern Recognition Workshops (CVPRW), 2015 IEEE Conference on*, pages 65–73. IEEE, 2015.
- [8] B. Foundation. Blender. <https://www.blender.org/>. [Online; accessed 02-03-2016].
- [9] T. K. Group. OpenGL. <https://www.opengl.org/>. [Online; accessed 02-09-2016].
- [10] I. Interactive Data Visualization. SpeedTree. <http://www.speedtree.com/>. [Online; accessed 02-03-2016].
- [11] 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*, 103:127 – 135, 2014.
- [12] Y. Livny, F. Yan, M. Olson, B. Chen, H. Zhang, and J. El-Sana. Automatic reconstruction of tree skeletal structures from point clouds. *ACM Transactions on Graphics (TOG)*, 29(6):151, 2010.
- [13] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.
- [14] OpenGL. Rendering Pipeline Overview. https://www.opengl.org/wiki/Rendering_Pipeline_Overview. [Online; accessed 02-09-2016].
- [15] J. Schupp. Pruning by the Numbers. <http://shaponline.org/wp-content/uploads/2012/03/Pruning-by-the-Numbers-Jim-Schupp.pdf>. [Online; accessed 02-03-2016].
- [16] W. van Keulen. The Grove. <https://www.thegrove3d.com/>. [Online; accessed 02-03-2016].