Proceedings of the 2012 IEEE
International Conference on Robotics and Biomimetics
December 11-14, 2012, Guangzhou, China

Image-based Tree Pruning

Wei Liu[1,*] George Kantor[*] Fernando De la Torre[*] Nanning Zheng[1,†]

[1] Xi'an Jiaotong University, AI&R Institute.

[*] Robotics Institute, Carnegie Mellon University. 5000 Forbes av, Pittsburgh, PA 15213, USA

liuwei204@cmu.edu, kantor@ri.cmu.edu, ftorre@cs.cmu.edu, nnzheng@mail.xjtu.edu.cn

***Abstract*— There is an increasing awareness and development of agricultural robots to take the toil of farming by automating growing plants and trees. Pruning is an expensive and labor intensive step in growing trees, that greatly affects its productivity. Moreover, pruning requires knowledge about what, where and how to cut. To partially solve the limitations of manual pruning methods, this paper presents an automatic image-based pruning system. Our system uses a high-resolution and a Kinect camera mounted on a mobile robot to capture the 3D structure of trees in the field. The robot goes around a tree and synchronously captures high-resolution and depth images. The visual and depth information across images is fused to estimate a 3D "stick" representation of the tree. The output of our system suggests the operator which branches to cut based on pre-existing rules. Several challenges contribute to the difficulty of image-based pruning: (1) fusing spatial and temporal information in depth images, (2) capture and segment small branches, (3) quantitative estimation of the angles and length for each branch. The number of suggested branches to cut in several trees have high agreement with the ones suggested by an expert, that illustrates the validity of our approach.**

Keywords: tree pruning, SLAM with kinect, agricultural robotics

I. Introduction

Proper tree trimming and pruning are essential to increasing a tree's health, strength and productivity. However, tree pruning is time consuming, manual labor is expensive and difficult to hire (due to the existing immigration policies). Moreover, proper pruning requires a good knowledge of the particular tree and what, where and how to cut. For instance, in the case of wine making, the vineyard has to be cut at a particular angle to produce better wine. Hiring someone who does not know when or how to prune correctly can prove to be disastrous to the tree and the productivity. Because pruning is an expensive and key step in agricultural business, developing robotics systems that can effectively and optimally prune trees would have a huge impact in current agricultural systems. Robots can make perfect cuts, work continuously (even at night) and provide consistent cuts. Beyond pruning trees for agricultural purposes, automatic pruning robotic systems will become fundamental for sustainable forest management. This paper presents a system that takes as input 3D information of a tree, as well as high-resolution images, and it provides a recommendation system of which branches to cut. Fig. 1 shows an image of our system.

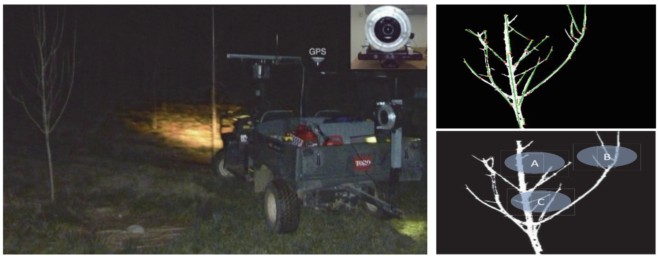

Fig. 1. Mobile robotics system for image-based tree pruning. The system is equipped with a high-resolution flash camera, a Kinect camera and a GPS. Right) In the top the recovered "3D stick" tree representation. In the bottom the suggested pruning.

Tree modeling has been an active area of research in the last few years due to advances in new sensors and computational resources. For instance, in the computer graphics community [1] [2] [3], there have been several approaches to recover the tree structure from images. However, most of them involved expensive computational methods and require manual labeling. In the computer vision literature, Adrian [4] used two orthogonal images of the tree to estimate the three structure achieving similar performance as a laser based method. Beyond tree modeling, there has been some work on robotics systems for tree pruning. Nielsen [5] used a stereo based-system to 3D blossom mapping for automated thinning of peach blossoms on perpendicular V architecture trees. Recently, the Vision Robotics corporation [6] has developed a grape pruning robot system, that use a pair of stereo cameras to reconstruct grape vines for pruning. Existing methods for tree-modeling or pruning only use two or more images to build the 3D model or laser information. These techniques allow to capture the coarse 3D structure of the tree but it is unclear how to reconstruct small branches that are required for pruning. In this paper we propose a cheap vision system that combines low resolution 3D Kinect camera with a high-resolution camera.

II. Method

This section describes in detail our system, that is based on four modules: (1) localization of the tree's trunk and tree segmentation using depth and color image, (2) branch detection in a single 3D image using a Hough transform-based method, (3) construction of the 3D tree model by merging several 3D images using the Iterate Closes Point (ICP) algorithm, (4) pruning strategy. Fig. 2 illustrates the main idea of our system.

978-1-4673-2127-3/12/$31.00 © 2012 IEEE

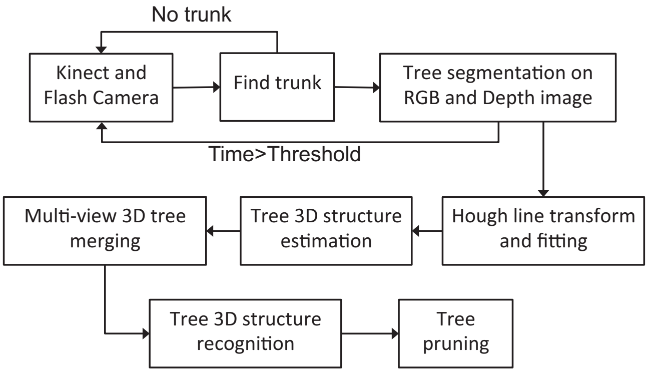
No trunk
Kinect and Flash Camera
Find trunk
Tree segmentation on RGB and Depth image
Time>Threshold
Multi-view 3D tree merging
Tree 3D structure estimation
Hough line transform and fitting
Tree 3D structure recognition
Tree pruning

Fig. 2. Main modules in our system.

A. Sensors

Our system uses a Kinect camera that provides 640×480 pixel depth images. A main drawback of the kinect sensor to capture outdoors depth images is the interference with the light spectrum, other 3D sensors such as laser and radar can work during the day, but we want to keep our system cheap and kinect provides an affordable hardware. So we capture the data at night. However, the visible camera cannot work at night without a special lighting. As illustrated in Fig. 1, our solution combines the kinect with a high-resolution flash camera that allows capture at night, and provides better color quality images than the visible kinect camera, that will be crucial for the color-based segmentation. To solve for correspondence between the RGB images and the kinect depth image, we calibrate the cameras. We used the kinect internal camera parameters provided by the manufacturer, and the Matlab calibration toolbox to know the 3D rotation and 3D translation between the optical centers of the cameras. The flash camera takes pictures at 2 f.p.s, the kinect works at 30 f.p.s. and the GPS at 100 f.p.s., so our system is synchronized at 2 f.p.s. The synchronization among the sensors is done by recording the UTC time.

B. Tree Segmentation

This section describes the algorithm for segmenting the tree in the high-resolution RGB images. We tested several algorithms for segmentation including Grab-cut [7] , Grow-cut [8] and Mean-Shift. However, these methods were likely to produce segmentations containing the background. Instead, we used a simple region growing algorithm.

The first step in the region growing algorithm is to select a seed using the high-resolution RGB image I_c. We assume there is only one tree in the image, and we automatically located the lower part of tree that contains only the trunk (no branches), see Fig. 3. There are three main steps to segment the trunk: (1) select the lower fourth part of the color image I_{cl} can convert to gray I_{gl}, (2) filter the image with a Gaussian filter, I_{gls}, (3) use canny edge detection to get a binary image I_{edge}, (4) detect two vertical lines L_1, L_2 that are longer and more vertical than a threshold, and with a minimum horizontal distance (radius r) between them. Finally, get the middle point P_s between the two lines.

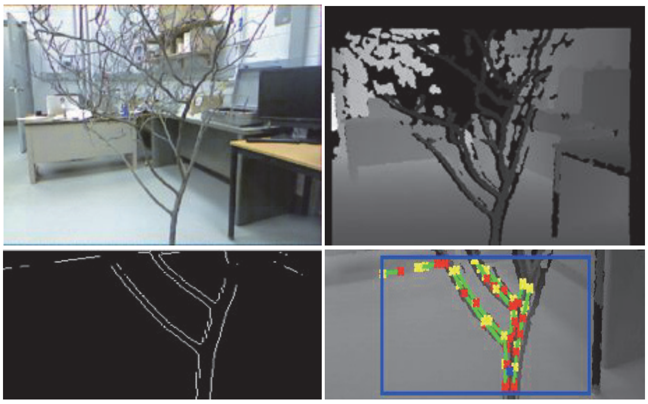

Fig. 3. Top left) Original color image. Top right) Depth image. Bottom left) Edges on Gaussian filter smoothed image, Bottom right) Detected lines (boundaries of the trunk) using the Hough transform, the blue cross is the selected seed.

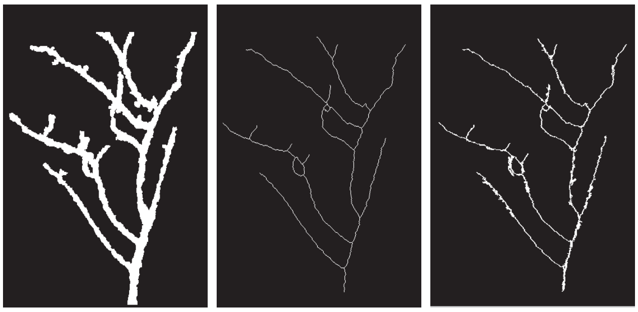

Fig. 4. Left) Region growing on the depth image. Middle) Tree skeleton. Right) Segmentation in the high-resolution RGB image combined with the depth image.

Once the trunk area is detected, we selected a pixel of the trunk and use it as a seed for the region growing algorithm. See Fig. 3 for an illustration of the method.

Figure 4 shows results for the tree segmentation. Observe that the lighting condition for each branch suffer from large variations, especially for the small branches and the branches edges. While the kinect camera provides low resolution and it can miss the small branches, it provides a good initialization for the region growing algorithm to segment the high resolution image.

Once the seed is automatically selected, the next step is to add neighboring pixels that correspond to the same segmentation. Observe that the brightness of the RGB pixels will be biased for different lighting conditions depending on the distance from the lighting source and normal to the lighting source (that is uniform). So, growing the region using only color image is likely to do not result in accurate segmentations. On the other hand, using only the depth, is likely to miss the small branches and get a bigger diameter than it should be. For these reasons, we first grow the tree using the depth image with the seed found from the RGB image (observe that we know the correspondence between a pixel in the depth image and the RGB image). Given an initial skeleton of the tree estimated from the depth image, we used the whole skeleton as seeds for a further

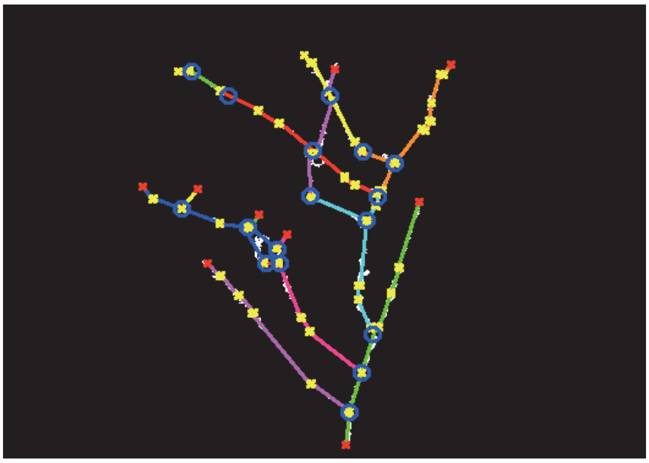

Fig. 5. 2D structure of the tree, blue circles represent the crotches, red crosses are branch ends. Same color represents the same branch.

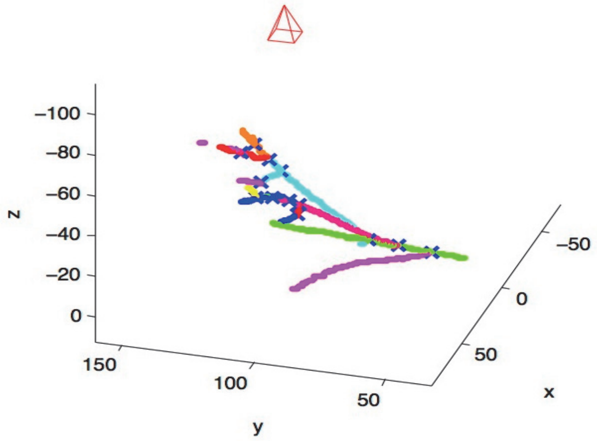

Fig. 6. 3D structure of the tree from one frame. Measures are in centimeters.

segmentation using the RGB and depth images. Observe that with multiple seeds that contain different lighting conditions it is likely that the region growing correctly segments the tree. Fig. 4 illustrates the process.

C. 3D tree estimation

Once we have accurately segmented the tree in the RGB and depth image, we can select these pixels in the depth image and obtain a "stick" figure of the tree. With the RGB and the depth images, we built an estimate 3D cloud representation of the tree. We are interested in detecting all branches and being able to accurately measure the length, diameter and angles of the branches, and we used the Hough line transform [9]. We proceed iteratively in the binary segmentation image. Every iteration labels the line with most pixels, and excludes this line's location area with a width according to the branch diameter that is an pre-estimated threshold. We do not proceed the Hough line transform once for all, because by doing this, a lot of unexpected lines will be introduced. Which are very hard to be rejected, considering the variability of the branch lengths, diameters, orientations and curvatures.

It is very hard to model the shapes of the branches, so an integrated branch could not be recognized with only Hough line transform. Other than this, the branches often come across with each other, one may split the other by using the proposed line detection method. So the following two operations are introduced for these problems. First, extend each line in a restricted region. The region is the area with a distance to the line smaller than the branch diameter and has positive binary values in the segmentation image. Second, group the lines belong to the same branch. Every two lines will be grouped under the conditions that their cross angle and the distance of their nearest points are smaller than certain thresholds. Otherwise, if the distance is smaller but the angle is bigger, they are considered belonging to different branches and their cross point is recognized as a crotch. Then we can obtain all the branches, and the most vertical branch which contains the first seed is the trunk. Fig. 5 shows the result of the branch recognition.

After we recognize all the branches, we will try to get their measurements. We have chosen several locations in the branch to estimate its diameter. In each location, we get the pixel width of the branch in the segmentation image. As we know the camera visual angle and resolution, and the metric distance between the branch and the camera, we can obtain the branch diameter in this location using triangulation. The branch has different diameters in different locations, we use the average value as its common diameter. We can also get the branch lengths, orientations and crotch locations in the branches using the same method.

There are several ways to describe the 3D structure of the branches. One is to get the 3D locations of all the branch end points and crotches, but it is prone to be crashed by a single wrong value. The other is to get every pixel's depth of the branches, this contains a large number of redundant values. We sample the pixels in every one centimeter along the branches. But the depth values are not smooth and continuous because of occlusions and resolution. So a curve fitting operator is activated. This could also benefit the small branch tails which are visible in high resolution optical camera but invisible in kinect depth camera. Fig. 6 shows the 3D structure of the tree.

D. Multi-view merging of 3D images

In previous section, we have described the method to detect the branches from a pair of RGB-depth images. However, it is difficult to model all branches using only one image due to occlusions and lack of high resolution depth. In order to build more accurate 3D models, this section describes the process to merge the information of several pairs of RGB-depth images.

It is important to notice that most branches are not straight lines, and just directly merging the "stick" figures across views will not provide the most accurate merging. A proper way to describe the shape of the tree is a non-parametric approach using a point cloud. We will merge the point clouds

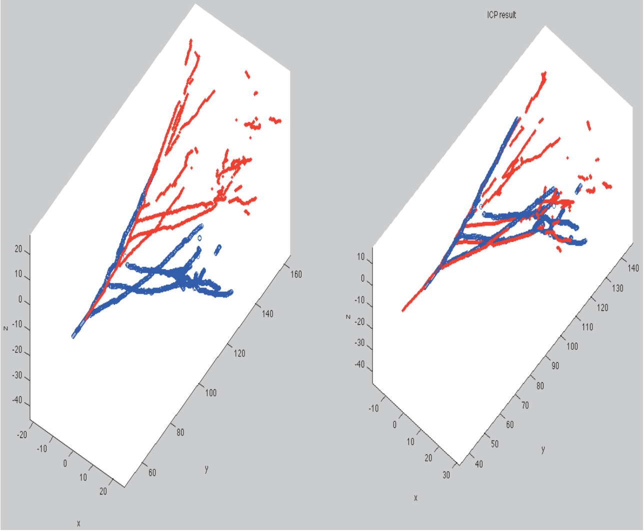

Fig. 7. Left) The 3D trees from different visual points, Right) The merged result.

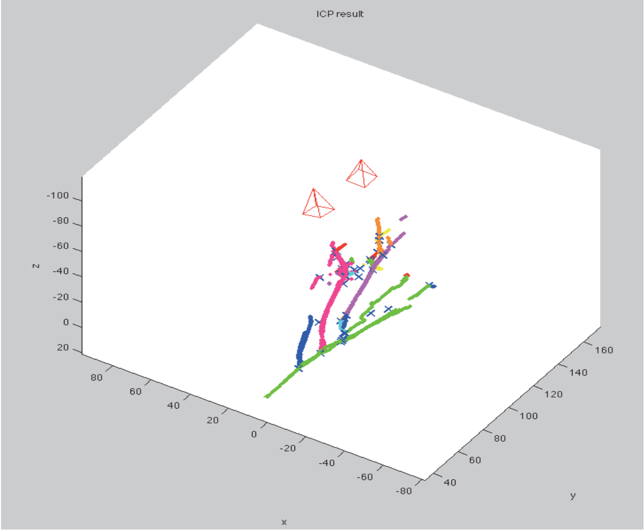

Fig. 8. The merging result from two different visual point. Different colors represent different branches. The bifurcation in same color is cause by error. The small blue pyramids are the camera from different locations.

that correspond to the trees at different views. The most popular method to align 3D point clouds is the Iterative Closest Point (ICP) algorithm [9].

Consider two point clouds $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, ...\mathbf{m}_n\} \in \Re^{3\times n}$ (see footnote for notation) [1] and $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, ...\mathbf{d}_m\} \in \Re^{3\times m}$. Then ICP algorithm will find the 3D Rotation (**R**) and translation (**T**) by minimizing:

$$[\mathbf{R},\mathbf{T}] = \underset{\mathbf{R},\mathbf{T}}{\text{argmin}} \sum_{i=1}^{K} \|\mathbf{R}(\mathbf{d}_i) + \mathbf{T} - \mathbf{m}_i\| \tag{1}$$

Using the difference of the average point locations as the initial translation ($\mathbf{T}_0$), and rotation as initial rotation ($\mathbf{R}_0$). K is the number of matched points, which is smaller than m and n. d_i and m_i are the corresponding points that have the closest distance in the two points clouds. Observe the point clouds contain different number of points and the two objects are not exactly the same. So, if more than 80% of points in the cloud are matched, it will be consider as a good match. Fig. 7 shows the point cloud be merged from two different view points. The pyramids represent the camera locations.

In fact, we could only get the surface information, the branches should be cylindrical after reconstruction. But in our project, the structures and sizes are most considered. So we could use the surface points average value instead of a fully surface reconstruction.

Figure 8 shows the 3D trees from two visual points. The matched points are merge together and the other points are remained.

E. Tree Pruning

Once we have detected all branches in the tree, we are ready to provide recommendation of which branches to prune. Figure 9 shows a hierarchical tree decomposition.

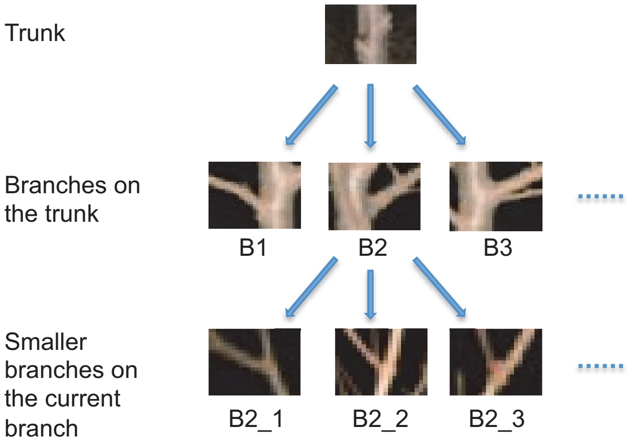

Fig. 9. Hierarchical structure of the tree. Every branch contains the location, diameter, length and orientation.

The tree is described hierarchically for pruning, as shown in figure 9.

Every branch contains four sources of information, location Lo, diameter D, length Le and orientation O. The trunk's location is its metric coordinate with respect to the trunk, orientation is its slant angel. Except for this, the branch location Lo is the crotch position in its mother branch, A is the branch angle with its mother branch, D is branch average diameter, Le is the branch total length.

Once the hierarchy is built, we can start the pruning following [10], who has professional knowledge about tree pruning. In Fig. 1 bottom right, branch A should be cut off because it goes into the main trunk, B is because the small branches is too close to each other, C is because two close branches come across. We will provide some more result in the experiment section.

[1] capital letters denote matrices **D**, bold lower-case letters a column vector **d**. $\mathbf{d}_j$ represents the j^{th} column of the matrix **D**. All non-bold letters represent scalar variables. d_{ij} denotes the scalar in the row i and column j of the matrix **D** and the scalar i-th element of a column vector $\mathbf{d}_j$.

III. EXPERIMENT RESULT

The data is collected in the Raemelton farm located in Adamstown, Maryland. We have selected several kinds of trees that need to prune. Fig. 10 shows the Queen Elizabeth Hedge Maple and Fig. 11 shows the Wildfire Tupelo. We can see that the on-field tree structures are more complicated than the demo tree, but their backgrounds are much more clean. As shown in the figures, we can get the trees' 3D structures with our system.

Some branches of the Maple are chosen to show the metric results in TABLE I, the best results are marked with bold. The ground truth values are measured with rulers and protractors, which are marked with a subscript G. From the table we can see that the branch lengths are much more smaller than the ground truth. This is because the lost of the tiny tails of the branches. The differences of the branch diameters may be caused by several problems. First, the branch diameter varies a lot in different locations. Second, the crotches and the missing tails. Third, the uncertainty of the sensors. The branch locations are represented with the crotch locations in their mother branches. These values are more stable, although they are affected by the recognition precision of the crotches. The branch angles are also very stable, they are very accurate in 2D, but in 3D, they are affected by the precision of the depth values.

The metrical values of the camera locations come from the ICP results. Their ground truth values are given by a two mini-meters precision GPS sensor on the robot. It is not necessary to get the tree's global location, so the GPS are used to get the relative locations when the robot moves. The first location from ICP is set to the first GPS location. Then the following locations can be evaluated by the GPS results. The second pose and position results are shown in TABLE II. We can see that the ICP results are very accurate for a rigid tree object.

Then the professional rules are applied to the reconstructed trees. The pruning locations are shown in the bottom right image in Fig. 10. The tree contains a lot of tiny branches, which are hard to be reconstructed because (1) the tiny branches are colorful, which is caused by the lighting condition and the edge effect, (2) the depth camera has a low definition. But the pruning starts from close-trunk to far-from-trunk, so the tiny tails will not cause big problems.

TABLE I
TREE MEASUREMENT RESULT

	Le	Le_G	D	D_G	Lo	Lo_G	A	A_G
B1	**85.6**	**93.8**	**0.9**	**1.1**	82.4	78.7	**44.8**	**45.5**
$B1\_1$	14.2	26.3	0.61	0.58	**55.1**	**53.8**	35.3	36.2
$B1\_2$	12.1	47.5	0.82	0.75	68.5	65.2	17.9	18.5

IV. CONCLUSION

Despite the amount of research done on robotic automation for fruit care and harvesting done in the last few decades, there remains no clear viable, cost-effective approach toward robotic mechanization of many agricultural tasks. In this paper, we focused on a fundamental and expensive task that involves tree pruning. We developed a image-based pruning system that is automatic and recommends which branches to prune. In future work, we will explore extending the algorithm to work in real-time, as well as providing recommendations of which angles to use for cutting the trees.

TABLE II
CAMERA LOCATIONS WITH RESPECT TO THE CURRENT TREE.

Position(m)	X	X_G	Y	Y_G	Z	Z_G
L2	4.08	4.25	1.71	1.80	0.54	0.65
Position($^\circ$)	α	α_G	θ	θ_G	γ	γ_G
L2	1.24	1.10	1.42	1.31	1.24	1.12

ACKNOWLEDGMENTS

The authors would like to thank Steve Black, Qi Wang and Ji Zhang for the help of data collections, and Steven Nuske for the help of camera construction. The second author is supported by USDA Cooperative State Research, Education & Extension Service with matching support from industry and the Pennsylvania Infrastructure Technology Alliance. The first author is supported by the National Natural Science Foundation of China under Grant No.91120006.

REFERENCES

[1] J. Weber and J. Penn, "Creation and rendering of realistic trees", *in Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, New York, 1995, pp. 119–128.

[2] I. Shlyakhter, M. Rozenoer, J. Dorsey and S. Teller, "Reconstructing 3d tree models from instrumented photographs", *Computer Graphics and Applications, IEEE*, 2001, vol. 21, no. 3, pp. 53–61.

[3] B. Neubert, T. Franken and O. Deussen, "Approximate image-based tree-modeling using particle flows", *ACM Transactions on Graphics*, 2007, vol. 26, no. 3.

[4] A. I. Hapca, F. Mothe and J.-M. Leban,"A digital photographic method for 3d reconstruction of standing tree shape", *Ann. For. Sci.*, 2007, vol. 64, no. 6, pp. 631–637.

[5] M. Nielsen, D. Slaughter and C. Gliever, "Vision-based 3d peach tree reconstruction for automated blossom thinning", *Industrial Informatics, IEEE Transactions on*, 2011, vol. PP, no. 99, p. 1.

[6] [Online]. Available: http://www.visionrobotics.com.

[7] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images", *IEEE International Conference on Computer Vision*, 2001, vol. 1, pp. 105–112.

[8] V. Vezhnevets and V. Konouchine, ""grow-cut" - interactive multi-label n-d image segmentation", *Graphicon*, 2005.

[9] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, February 1992, vol. 14, pp. 239–256.

[10] S. Black, "http://www.raemelton.com/", Raemelton Farm, Adamstown, Maryland.

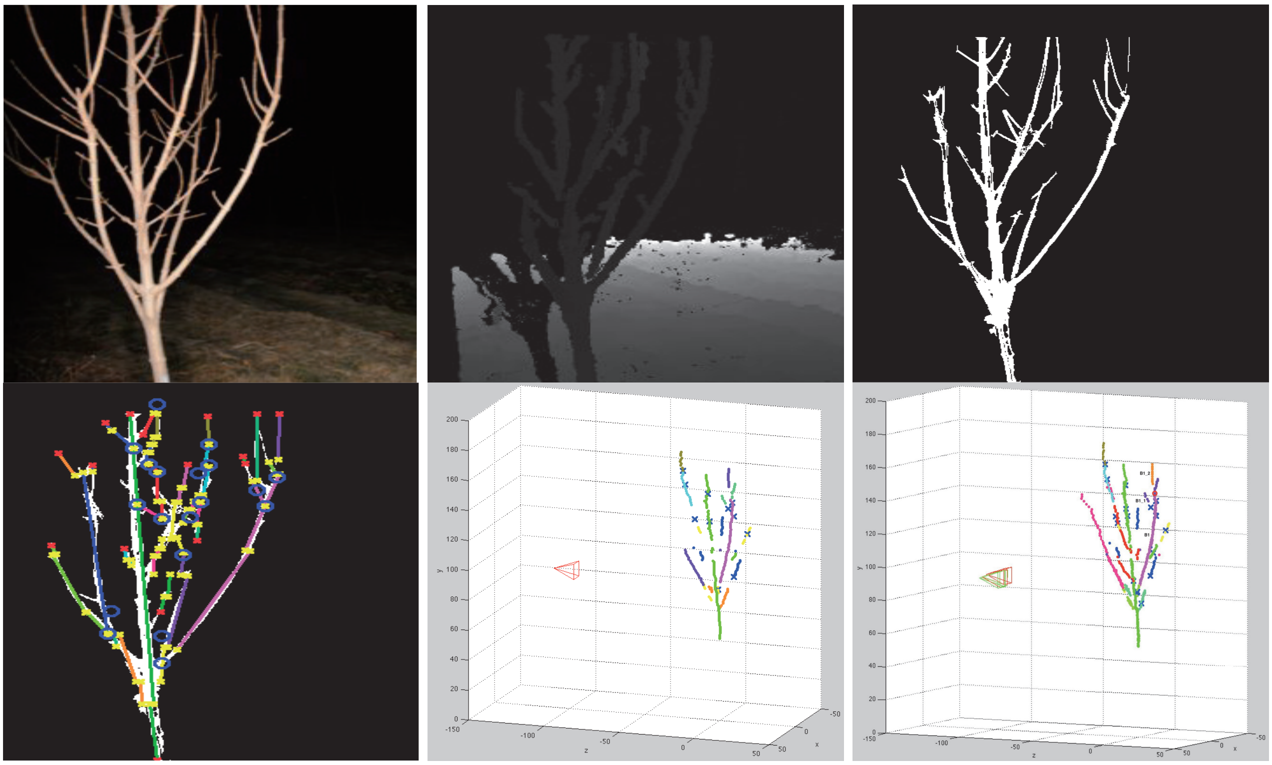

Fig. 10. Queen Elizabeth Hedge Maple. Top left) Hight-definition color image. Top middle) Kinect depth image. Top right) Segmentation result, Bottom left) Tree 2D structure. Bottom middle) 3D structure of the tree from one frame, Bottom right) 3D structure of the tree from different visual positions (Green is GPS location, which is used for evaluating the estimated positions from the visual information, Red is ICP result) with suggested pruning locations in red circles.

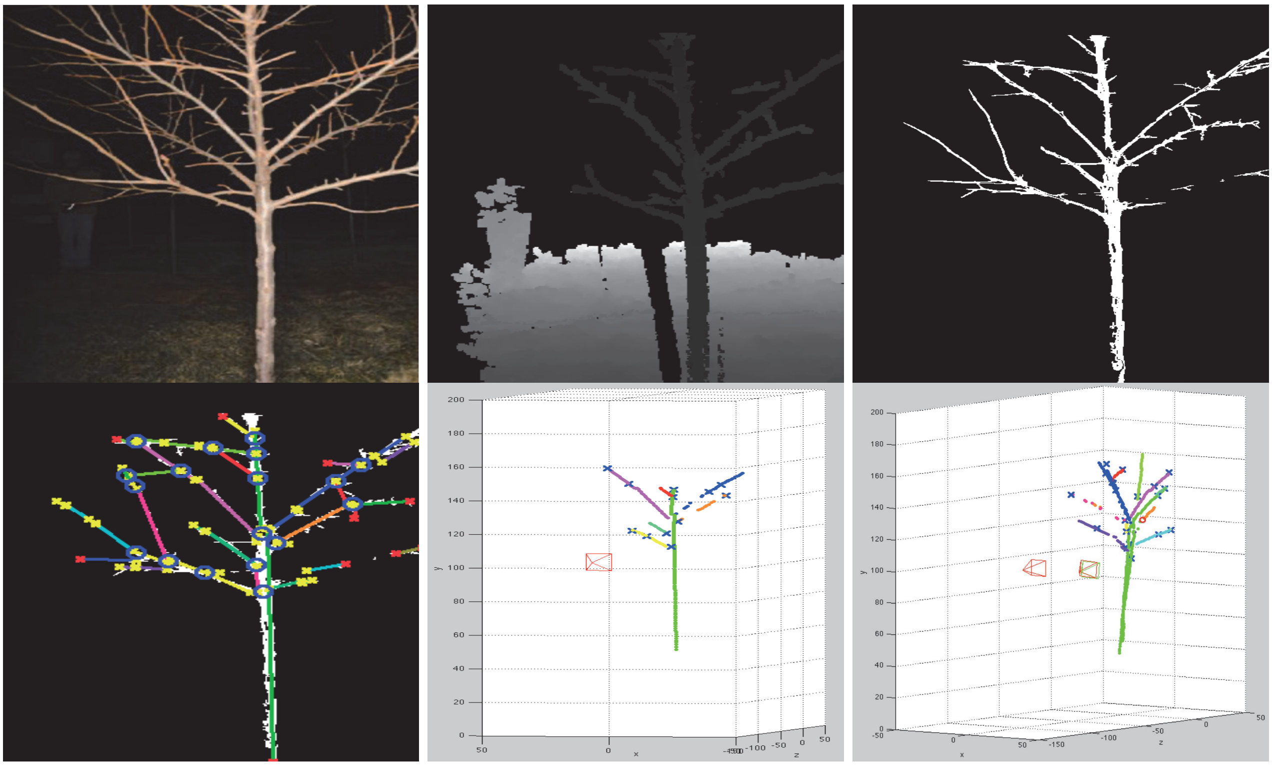

Fig. 11. Wildfire Tupelo. Top left) Hight-definition color image. Top middle) Kinect depth image. Top right) Segmentation result, Bottom left) Tree 2D structure. Bottom middle) 3D structure of the tree from one frame, Bottom right) 3D structure of the tree from different visual positions (Green is GPS location, Red is ICP result) with suggested pruning locations in red circles.