Automatic estimation of trunk cross sectional area using deep learning

T. Wang[1], P. Sankari[1], J. Brown[1], A. Paudel[2], L. He[1], M. Karkee[2], A. Thompson[3], C. Grimm[1], J.R. Davidson[1*], S. Todorovic[1]
[1]*Collaborative Robotics & Intelligent Systems Institute, Oregon State University, Corvallis, OR 97331, USA*
[2]*Center for Precision & Automated Agricultural Systems, Washington State University, Prosser WA 99350, USA*
[3]*Department of Horticulture, Oregon State University, Corvallis, OR 97331, USA*
*joseph.davidson@oregonstate.edu

Abstract

This paper presents an automated method for estimating the trunk cross sectional area of fruit trees. An Intel RealSense 435i was used to capture RGB images and point clouds of individual trunks. To segment the trunk in the image from the background, a Masked-attention Mask Transformer model was adopted. The segmentation results were integrated with the 3D point cloud to estimate trunk widths in 3D. The width estimation was evaluated on three diverse datasets collected from a commercial apple orchard using human measurements as ground truth. With a mean absolute error less than 5%, the method is sufficiently accurate to assist orchard operations.

Keywords: Trunk cross sectional area, computer vision, point cloud, deep learning

Introduction

Trunk cross sectional area (TCSA) is a metric that growers and horticulturists use to measure tree productivity in terms of wood mass. There is a positive correlation between the yield of a tree and the TCSA (Lepsis and Blanke, 2006; Kumar *et al.*, 2019), as a result of the positive relationship between TCSA and canopy volume. The industry uses TCSA to calculate the yield efficiency of trees (#fruit weight / TCSA = yield efficiency). TCSA can also be used to determine the number of high-quality fruit a tree can produce without stressing the tree and, therefore, the number of fruit to thin from a tree. TCSA is measured by finding the diameter of a tree at the appropriate height using calipers or a measuring tape and then calculating the area. Unfortunately, despite TCSA being a useful parameter in precision orchard management and a comparatively easy metric to gather on a single tree, when dealing with a large commercial orchard with thousands of trees, collecting the data quickly becomes untenable. An automated technique to measure the TCSA would greatly reduce the manual effort needed to collect per-tree data.
Many previous efforts have been made to detect tree trunks. In many of these cases, the goal was to localize the robot using the trunks as landmarks, so no attempt was made to accurately measure the TCSA of the detected tree. Shalal et al. (2015) developed an algorithm that used data fusion from a camera and a laser scanner to delineate between tree and non-tree objects, such as supports or posts. They used the laser scanner to determine the width of the tree-like objects and the camera to verify (based on the color) whether or not the detected object had parallel edges. No information was given about the precision with which they were able to measure the detected trunks. Another method to detect trunks by Bargoti et al. (2013) used Hough Transforms on LiDAR data to detect

trunk locations. Their aim was to use the trunk location data to build a tree inventory of the orchard, so no attempt was made to measure the trunk width.

Several efforts have used LiDAR to effectively measure tree diameter. Wang et al. (2019) used a terrestrial laser scanner, designed for forest inventory, in a man-made ginkgo forest to measure: the diameter at breast height, total height, and location of the trees. The forest was divided into 10m x 10m square plots for the experiment and a single scan was conducted for each area. The results were then compared to manual measurements; they were able to detect 92.75% of the trunks and had a root mean square error of 1.27cm for the diameter and 0.25m for the height. Bucksch et al. (2014), on the other hand, utilized airborne LiDAR data combined with a novel skeleton measurement methodology to extract the diameter at the breast height of trees in a forestry setting. While LiDAR produces accurate measurements, its cost may make it too expensive for orchard managers.

An image-based approach was proposed by Kan et al. (2008). They utilized a calibration stick placed next to the tree in an image to determine the actual size of pixels in the image at the location of the tree. The diameters of the trunks and branches of the tree were then acquired by processing and analyzing the image. Their method had a mean absolute error of 0.67cm and a mean relative error of 1.9%. Similar to this approach, the proposed approach is image only; however, the use of the depth data eliminates the need for manually placing a calibration pattern in the image.

This paper introduces a method that uses a state-of-the-art deep learning model to locate and estimate the width of tree trunks using just an RGB-D image. The model was trained on data collected during diverse conditions in a commercial orchard, giving it the ability to ignore excess foliage and robustly operate in variable lighting and orchard conditions. Manual measurements were gathered as a ground truth to evaluate the accuracy of the method. Results showed that the method is accurate within 5% of manual measurements.

Methods

Trunk width estimation can be difficult in the noisy outdoor environment of apple orchards. The appearance of trees can also vary greatly throughout the year. Additionally, it is challenging to separate vegetation from trunks in laser scans. In an effort to reduce the effects of sensor noise (and lower the cost), this approach uses a computer vision-based approach. The process begins with an Intel (Santa Clara, CA, USA) RealSense D435i sensor scanning a tree trunk (Fig. 1) and returning an image with additional depth information. A Masked-attention Mask Transformer (Masked2Former) model (Cheng et al., 2022) was trained for trunk segmentation (pixel-level predictions). The segmentation result was then processed, in conjunction with the depth information, to generate width estimates along the thinnest 40% of the trunk visible in the image. The orchard used for this study was a Jazz block trained in a tall spindle architecture with a wire and post trellis system. Data was taken across several years and different lighting conditions.

Figure 1. Utility vehicle with Intel Realsense D435i mounted near the rear at the desired height for measuring the trunk width. The camera continuously records RGB-D images as the vehicle traverses the orchard rows (Prosser, WA, USA).

Trunk segmentation

The latest state of the art model Masked2Former was adopted for instance trunk segmentation. This is a universal architecture capable of addressing panoptic, instance and semantic segmentation tasks. By using the masked attention, multi-scale high-resolution features and calculating mask loss on only a few sampled points, the proposed method has both high performance on evaluation metrics and is computationally efficient. Readers are referred to Cheng et al. (2022) for more details on the algorithm. For transfer learning on the trunk dataset, a pre-training model on COCO (Lin et al., 2014), with 80 "things" and 53 "stuff" categories, was adopted, keeping only one class in the segmentation head. The model was then fine-tuned for 200,000 iterations with a learning rate of 0.00025; this learning rate increased the rate of convergence compared to the 0.0001 learning rate used in the original Mask2Former while still ensuring stable convergence. The data augmentation described in Cheng et al. (2022) was also adopted during training.

Training datasets

The Masked2Former was trained on 450 images with pixel-level annotations from three different data sets (150 images from each dataset):

- Dataset 1 – *February 2020*: sunny, trees do not have foliage, background is mainly dry grass, no fruit on trees.
- Dataset 2 – *July 2021*: sunny, trees have green foliage, background has green vegetation, green fruit on trees.
- Dataset 3 – *November 2020*: sunny, trees have green foliage, background has dry leaves, red fruit on trees.

Labelme (https://github.com/wkentaro/labelme) was used to annotate the datasets, where a polygon was drawn to outline the trunk.

Medial axis and distance transform

After obtaining the mask, the next step was to skeletonize (Zhang and Suen, 1984) the mask, which generates the medial axis and the distance transform along the axis (see Fig. 2 for an example). The medial axis traces the center of the mask along the trunk, while the distance transform gives the minimal distance from a point on the medial axis to the mask boundary (the image radius). This step produces a radius estimate for each pixel along the medial axis (a horizontal "slice" of the image).

Figure 2. *Left*: The mask produced by Masked2Former. *Middle left*: The mask is processed to produce a medial axis (blue) with a distance to the boundary (an example is shown in green) at every point along the axis. *Middle right*: The corresponding depth image, with the green "slice" marked. *Right*: The pixel slices that are below the 40th percentile; the width of the largest of these (shown in white) was chosen as the trunk width.

Physical width estimation

The next step in the pipeline is to convert, for each "slice", the transformed distance from pixel coordinates to a measurement of trunk width in physical coordinates. Preliminary observations showed that measuring the distance between the depth coordinates of the left-right end points of the slice as the estimate was inaccurate due to noise in the depth map. Instead, the algorithm uses the camera's intrinsic parameters to calculate a meters/pixel ratio that is then used to determine the physical width for all transformed distances. Referring to Fig. 3, d is the depth (m) to points along the medial axis, shown in blue, with respect to the origin of the camera O. This depth is used to triangulate the height of the image b - a (m). From simple geometry the height can be found by Eq. 1:

$$b - a = 2 * d * tan\left(\frac{\alpha}{2}\right) \quad (1)$$

where α is the camera's vertical view angle of 42°. The meter per pixel conversion ratio is the image height divided by the camera's pixel height of 480 or 720, depending on the camera model (i.e. $(b - a) / 480 \; or \; 720$). For each horizontal slice, the width of the trunk is calculated by (# of pixels in the slice)*conversion ratio. These per-slice width measurements are combined in the next section to calculate a single trunk width estimate.

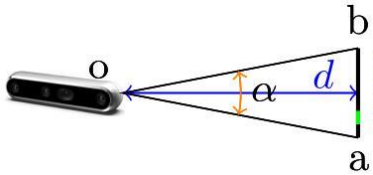

Figure 3. An illustration of the physical width estimation, where O is the position of the camera, α is the vertical view angle, d is the depth and line segment $a - b$ represents the vertical height of the image.

Automatic measurement selection

In field practice, humans take the measurements somewhere between 20 and 30 cm above the tree's graft union. The width varies substantially along the trunk, meaning there is significant human decision making in the current process. Manual inspection of where humans took their measurements showed that if the widths were calculated for every height within the predicted mask and then sorted from smallest to largest, the corresponding human measurements are approximately at the 40th percentile. Rather than choose a specific height in the image, the algorithm follows this practice and returns the

measurement at the 40th percentile. Note that this is the 40th percentile of the measured widths, not the measurement at a specific height.

Test datasets

Three test datasets (Row97, Row98, and Row100) that included separate orchard rows and different periods of the growing season were collected (see Table 1). Human ground truth measurements were obtained 20 cm and 30 cm above the graft union for Row97 and Row98; three separate measurements were completed at 30 cm for Row100. The average width measurements are used as the ground truth width for the final evaluation.

Table 1. Statistics of test datasets.

Dataset	**Row100**	**Row97**	**Row98**
Season	Blossom	Growing	Growing
Dataset size	100	79	75
Image size, pixels	480*640	720*1280	720*1280

Results and Discussion

Trunk segmentation

Table 2 reports the quantitative results from instance segmentation of tree trunks using the standard COCO detection evaluation metrics, i.e. Average Precision (AP) over multiple Intersection over Union (IoU) values (https://cocodataset.org/#detection-eval). The results are impressively competitive considering state of the art AP on mainstream tasks (https://paperswithcode.com/sota/instance-segmentation-on-coco) is only around 55.5%. Figure 4 shows some qualitative segmentation results for the three test datasets. Consistently aligned with the strong APs of the segmenter, it is also apparent by inspection that the trunk segmentation is very accurate in different lighting conditions and at various stages of the growing season, even when there is vegetation and grass present in the image.

Figure 4. The best (top row) and worst (bottom row) segmentation results (masked in white) for each of the three test datasets.

Table 2. Instance segmentation results of test datasets (overall).

Method	**AP**	**AP50**	**AP75**	**APm**	**APl**
Mask2Former	89.1	91.8	88.8	80.8	92.8

Automatic measurement selection

Figure 5 shows the slices that are below the 40th percentile for several images. Since each individual tree has varying shape and width along the height of the trunk, the smallest 40% of transformed distances returned from the medial axis calculations are sometimes not continuous and can be located at different regions of the trunk. The physical widths were calculated for all pixel slices in the red set, and the maximum value of these was selected as the automatic estimate of trunk width.

Figure 5. Sample automatic measurement selection results for the three test datasets. The white line indicates the selected slice.

Automatic width estimation compared to ground truth

Physical width estimates were compared against the human measurements for the three test datasets (Fig. 6). Table 3 shows the mean absolute error (MAE) and error standard deviation (ESD) of the algorithm's predictions. MAE for Row100 used as the true value the mean of three repeated human measurements taken at 30 cm above the graft union. For an average tree width of 6.71 cm from all test datasets, an MAE of 0.305 (Row100) represents 4.6% error in the prediction.

Discussions with growers and horticulturalists revealed that there is some variability in the height selected for manual measurements, depending on the individual collecting data. Therefore, the MAE for Row97 and Row98 incorporated the mean of human width measurements from 20 cm and 30 cm above the graft union as the true value. Table 3 reports the difference between these two measurements as 'Human diff'. The MAE of the automatic predictions for Row97 and Row98 were 0.294 cm and 0.295 cm, respectively, or approximately 4.4% error. The difference between two human measurements was approximately 0.15 cm, or approximately 2.2% error.

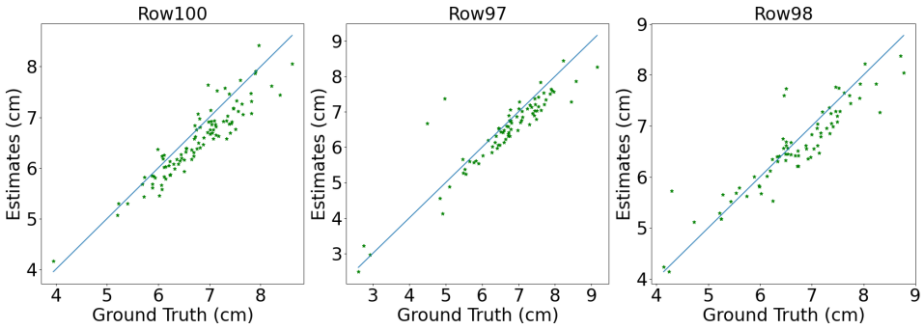

Figure 6. Width estimation and ground truth plots for different test datasets.

In the Row97 and Row98 datasets, there are 2-4 outliers. Upon careful examination, these outliers highlight two main sources of error: incorrect predictions and incorrect depth measurements. The first type of error, incorrect predictions, occurs when the segmenter fails to detect the target tree in the center of the image or when there are significant

occlusions caused by leaves or overlaps between the foreground and background trunks. The second type of error, incorrect depth measurements, occurs when part of the trunk is occluded by leaves and the sensor is unable to accurately capture the depth information of the trunk.

Table 3. Comparison with the baselines on the 3 datasets. 'Predicted' is the automatic estimate and 'Human diff' is the difference between human measurements at 20cm and 30cm above the graft union. The number in the parentheses is the number of trees included in the evaluation. One tree in the Row100 dataset didn't have depth information, and the target tree wasn't detected in two of the Row97 images.

	MAE (cm)	**ESD (cm)**
Row100 (99)		
Predicted	0.305	0.193
Row97 (77)		
Predicted	0.294	0.212
Human diff	0.158	0.152
Row98 (75)		
Predicted	0.295	0.285
Human diff	0.143	0.246

Efficiency

The input to the algorithm is RGB-D data, where RGB information is represented as PNG images and D is NumPy data pre-extracted from the original polygon file format (PLY) file. The algorithm was evaluated on a computing cluster using one Dell AMD EPYC compute node (Model: 2x Dell PowerEdge R7525; Processor 2x 32-core 2.6 GHz AMD; GPUs: 2x Nvidia A40 w/ 48 GB; Memory 256 GB RAM). Table 4 shows the runtimes of the algorithm for several subtasks; total execution time varied from 0.36 to 0.67 sec per image. Optimizing the performance of the framework for deployment on an embedded system in the field is the subject of future work.

Table 4. The runtime (seconds) of the proposed method.

	I/O	**Trunk Segmentation**	**Width Estimation**	**Overall**
Row100	0.013	0.265	0.085	0.363
Row97	0.025	0.484	0.161	0.671
Row98	0.025	0.476	0.160	0.661

Conclusion

This paper described a computer vision-based method for automatically estimating the cross sectional area of apple tree trunks in a commercial orchard. A state-of-the-art Masked2Former model was used to segment tree trunks in RGB images, and the depth to the tree was used to convert the transformed distance of the trunk mask to a physical measurement of the trunk width. Evaluation of the algorithm on multiple datasets showed

that the algorithm's predictions were within 5% of the ground truth human measurements. Likewise, the algorithm performed robustly across images captured in different lighting conditions and at varying stages of fruit production (e.g. flower blossom, green fruitlet period, etc.). The presented framework should be generalizable to other tree types and orchard systems. Future work will integrate this technique as a tool in precision orchard management practices.

Acknowledgements

This research is supported in part by the Washington Tree Fruit Research Commission and the AI Research Institutes Program supported by NSF and USDA-NIFA under the AI Institute: Agricultural AI for Transforming Workforce and Decision Support (AgAID) (award No. 2021-67021-35344).

References

Bargoti, S., Underwood, J.P., Nieto, J.I., and Sukkarieh, S. (2013). A pipeline for trunk localisation using lidar in trellis structured orchards. In: Mejias, L., Corke, P. and Roberts, J. (eds.): Results of the 9th Int'l Conf. on Field & Service Robotics., New York City, USA: Springer Publishing, pp. 455-468.

Bucksch, A., Lindenbergh, R., Zulkarnain, M., Rahman, A. and Menenti, M. (2014). Breast height diameter estimation from high-density airborne lidar data. IEEE Geoscience and Remote Sensing Letters 11(6), 1056-1060.

Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., and Girdhar, R. (2022). Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CF Conf. on Computer Vision & Pattern Recognition (CVPR). IEEE, New York City, U.S.A., pp. 1280-1289.

Kan, J., Li, W., and Sun, R. (2008). Automatic measurement of trunk and branch diameter of standing trees based on computer vision. In: Proc. of the 3rd IEEE Conf. on Industrial Electronics & Applications. IEEE, New York City, U.S.A., pp. 995-998.

Kumar, D., Srivastava, K.K., and Singh, S.R. (2019). Correlation of trunk cross sectional area with fruit yield, quality and leaf nutrient status in plum under the North West Himalayan region of India. Journal of Horticultural Sciences 14(1), 26-32.

Lepsis, J., and Blanke, M.M. (2006). The trunk cross-section area as a basis for fruit yield modelling in intensive apple orchards. In: Braun, P. (ed.): Proc. of the 7th International Symposium on Modelling in Fruit Research and Orchard Management. Acta Hort 707, ISHS, pp. 231-235.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). Microsoft COCO: Common objects in context. In: Proc. of the European Conf. on Computer Vision., Berlin, Germany: Springer, pp. 740-755.

Shalal, N., Low, T., McCarthy, C., and Hancock, N. (2015). Orchard mapping and mobile robot localisation using on-board camera and laser scanner data fusion – Part B: Mapping and localisation. Computers and Electronics in Agriculture 119, 267-278.

Wang, P., Li, R., Bu, G., and Zhao, R. (2019). Automated low-cost terrestrial laser scanner for measuring diameters at breast height and heights of plantation trees. PLoS ONE 14(1).

Zhang, T.Y., and Suen, C.Y. (1984). A fast parallel algorithm for thinning digital patterns. Communications of the ACM 27(3), 236-239.