Real-Time Construction of Fruit Tree Model Based on Images

Yongping Li
Ningbo Dahongying University, CHINA

Xingyuan Li
Ningbo Dahongying University, CHINA

Si Tian
Ningbo Dahongying University, CHINA

ABSTRACT
Using the binocular stereo vision system, the branches of citrus trees in natural scene were reconstructed in virtual environment, to help citrus picking robots recognize and evade obstacles in real working scene. During the reconstruction, images were subjected to thinning, pruning, and curve fitting successively. We reduced the computational burden while guaranteeing the model precision. Then, we adopted the principle of modularized modeling and OpenGL for branch reconstruction. It is verified that the method developed in this work provides a route planning criterion and a virtual workplace for the robot's obstacle evading system.
Keywords: thinning, pruning, curve fitting, modularized modeling, real-time modeling

INTRODUCTION

As the main operating component of fruit picking robots, mechanical arm is featured by complex structure and high cost. Mechanical arm is prone to damage in collision with thick branches and trunks. Hence, it is necessary to perform researches for helping robotic pickers evade obstacles and design routes. The general idea is to predict three-dimensional scenes of obstacles (fruits, branches, or trunks), and then plan routes for obstacle evasion (Song, Wang, & Sheng 2016). The first step is to build a three-dimensional database of obstacles according to their locations and to map real scenes onto the virtual world, thus creating a virtual picking environment for fruit picking robots. Then route planning is conducted in the virtual scene. Finally, simulation result is transmitted to joint motors and end effectors to achieve collision-free picking.

As the branches and trunks of fruit trees are major obstacles in the picking environment, the accurate construction of branch and trunk model is critical for virtual-scene construction. To accurately model a tree, Kempthorne, Turner, Belward, Mccue, & Young, et

Correspondence: Yongping Li, *Department of Information Engineering, Ningbo Dahongying University, Ningbo City, Zhejiang Province, China,315175.Tel: +886-135-6659-6110.*
✉ liyoongping@qq.com

State of the literature

- The general idea is to predict three-dimensional scenes of obstacles and then plan routes for obstacle avoidance.
- After image acquisition, there are a lot of redundant data, and these data directly reduce the subsequent computing speed.
- The model generation rate hardly meet the requirements of real-time obstacle avoidance for harvesting robot.

Contribution of this paper to the literature

- Using the binocular stereo vision system, the natural scene was photographed to reconstruct the branches of citrus trees in the virtual world. This provides a route planning criterion and a virtual workplace for the robot's obstacle evading system
- Using the method of multi-line fitting, the redundant data is reduced, and the necessary data is entered into the modeling stage to improve the overall operation speed.
- Using the data required by the modeling calculation, the visual effect and obstacle avoidance effect of the model are synthetically investigated, and the real-time virtual scene for the robot action is provided for the operation.
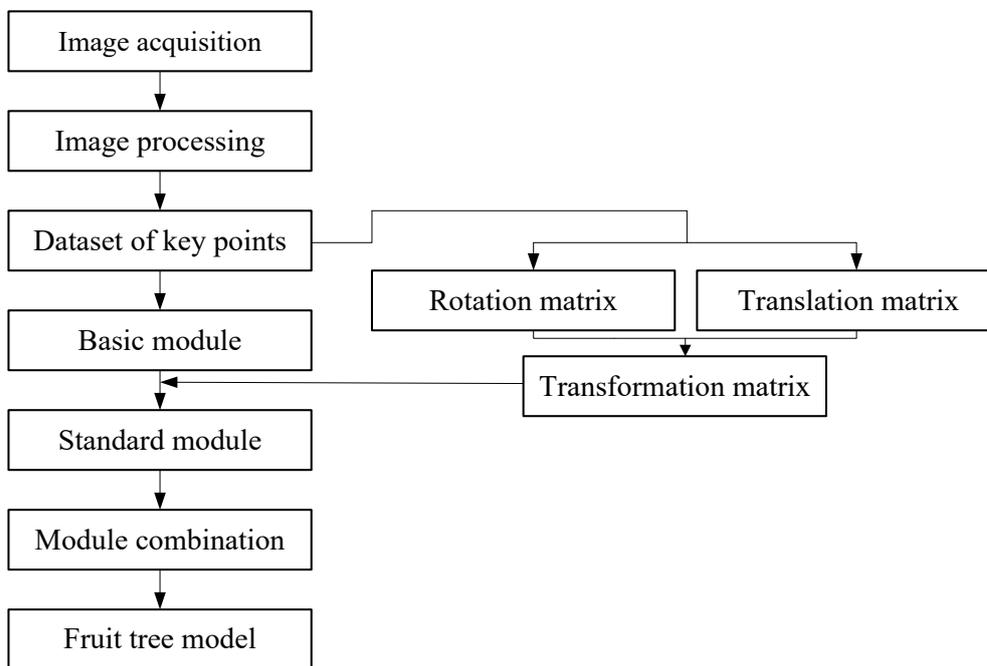
al. (2015), Fernandes, Armada-Da-Silva, Pool-Goudaazward, Moniz-Pereira, & Veloso (2016), and Wang, Sibeck, Merka, Boardsen, Karimabadi, & Sipes, et al. (2013) extracted and analyzed tree framework and form by a combined application of image and graphic processing methods. Fu, Li, Du, Han & Zhang, et al. (2015) adopted curve fitting to control the collected tree data and developed a tree model with controllable precision. However, these studies paid less attention to accurate real-time modeling. Thus, their methods cannot be used to create a virtual environment for fruit picking robots. Based on the binocular stereo vision system, we tried to accurately reconstruct the virtual branches and trunks of citrus trees in real time, providing a real-time scene for route planning for fruit picking robots. To generate a dataset of key points(Gonzálezjosé, Escapa, Neves, Cúneo, & Pucciarelli, 2008), the information of key points was extracted by curve fitting after segmentation, binarization, thinning, and short shoot removal for the collected images. Subsequently, basic modules, rotation matrix, and translation matrix were generated according to the dataset. Then, using the transformation matrix calculated based on rotation and translation matrices, the basic modules were transformed spatially to generate standard modules. Finally, a virtual fruit model was constructed by the combination of multiple modules (see **Figure 1**).

MATERIALS AND METHODS

Data acquisition and information extraction for key modeling points

Acquisition tools and computer platform

The accuracy of data acquisition is vital for precise modeling. In this study, data acquisition was based on the binocular stereo vision system. Binocular camera Bumblebee2 (Point Grey), which adopts IEEE-1394 card to control the camera or transmit videos, was used

```
┌─────────────────────┐
│  Image acquisition  │
└─────────────────────┘
          │
          ▼
┌─────────────────────┐
│  Image processing   │
└─────────────────────┘
          │
          ▼
┌─────────────────────┐          ┌──────────────────┐   ┌────────────────────┐
│ Dataset of key points├─────────┤ Rotation matrix  │   │ Translation matrix │
└─────────────────────┘          └──────────────────┘   └────────────────────┘
          │                                    │
          ▼                          ┌────────────────────────┐
┌─────────────────────┐             │  Transformation matrix  │
│    Basic module     │◄────────────└────────────────────────┘
└─────────────────────┘
          │
          ▼
┌─────────────────────┐
│   Standard module   │
└─────────────────────┘
          │
          ▼
┌─────────────────────┐
│ Module combination  │
└─────────────────────┘
          │
          ▼
┌─────────────────────┐
│  Fruit tree model   │
└─────────────────────┘
```

Figure 1. Flow chart of real-time construction of virtual fruit tree model

as data gatherer. Bumblebee2 can directly gather RGB color values in current field of view and coordinates of the corresponding spatial location *(x, y, z)*, and transmit data to computer (CPU: Intel(R) Core(TM) i3-4160, 3.60 GHz; Memory: 2.00 GB) via the IEEE-1394 card.

Information extraction for key points of modularized modeling

A fruit tree consists of branches, trunk, leaves, and fruits, which vary greatly in morphology. As branches and trunks are hard wooden structures, they are considered as obstacles of the mechanical arm in route planning for robots (Kim & Jeong, 2014). Leaves and shoots are soft and flexible, so they are not considered as obstacles. Fruits are picking objects of the mechanical arm, namely, targets in route planning (Urrutia, Lara, Villalba, Christie, Quesne & Cuq, 2011). Therefore, branches and trunks are main components of the virtual scene to be constructed.

In modularized modeling, cylinder and conical frustum can be taken as modules because most of branches and trunk have such shapes (Yang, Shen, Yuan, & Gao, 2015). The most concise information describing these modules is radii of the circles at both ends and spatial locations of centers of circles. Take the branch in **Figure 2** as an example. As long as the spatial locations of O_1, O_2, O_3, and O_4 as well as radii of the circles at both ends are got, the branch can be described fully. Essentially, the information extraction for key points of modularized modeling is to acquire the information of key points such as O_1, O_2, O_3, and O_4 from the image of real scene.

4037
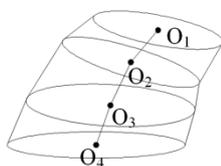
Figure 2. Tree morphology analysis

Figure 3. Branch information extraction from the real scene

Coordinate and radius acquisition for points on the center line of branch

All key points are on the center line of branch image. The radius of the corresponding circle of a key point refers to its distance to the edge. Hence, we need to extract the center line of branch from the image at first. Then the coordinates of key points and the corresponding radii can be solved. Citrus trees were the modeling objects in the present work. The information concerning branches, trunks, leaves, fruits, and background was gathered. To extract the information of key points, branches and trunks needed to be segmented from the background to generate a separate image. **Figure 3** shows the process of gradual segmentation (Agrawal, Barnwell & Raskar, 2010; Letsch & Kjer, 2011; Mccraig, Osinski, Cloutis, Flemming, Izawa, & Reddy, et al, 2017; Yang, Shen, Yuan, & Gao, 2015;).

To obtain the information concerning the center line of branch or trunk, the separate image should be binarized firstly (**Figure 4a**), followed by distance transformation and thinning (Tan, Jamdagni, He, Nanda, Liu, & Hu, 2015). **Figure 4b** shows the branches thinned by Yokoi's eight-connectivity-number algorithm (Morrison, Huckvale, Corish, Dorn, Kontschieder & Hara, et al., (2016)).
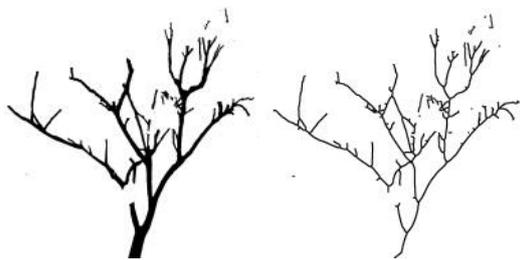
(a) Binary image   (b) Graph thinning

**Figure 4.** Branch images after binarization (a) and thinning (b)

Data optimization

After distance transformation and thinning, the coordinates of any pixel on the center line and the corresponding radius (set as $\{(x, y, z), R\}$) can be solved according to known spatial locations (Štefan & Strnad, 2015). But there were some redundant points. Hence, data optimization should be performed to remove redundant points (Mossel, Roch & Sly, 2011), so as to avoid their impact on real-time modeling.

The branches of fruit trees occupy a small space at intersections and bifurcations, which are insignificant for route planning. So the nodes where branches are connected can be removed (Zhang, Yuan, Gao, Wei & Diushen, et al., 2015). That is, the pixel (node) surrounded by over two pixels in the thinned image was regarded as redundant point and removed. Branches were no longer connected after redundant points were eliminated.

In the meantime, there were a number of short branches at the periphery, which had little influence on the overall structure of a tree (List & Mashayekhi, 2016). Hence, they were also pruned as a bunch of redundant points (Schmitt, Shahzad, & Zhu, 2015). The specific operations were as follows. According to a preset threshold (the ratio of branch length to branch radius), whether the disconnected branches were too short were judged. The branch should be pruned if its ratio of length to radius was smaller than the threshold (Escamez, Sirois, Lahtinen, Stenvall, Badel & Tixador, et al.). **Figure 5a** shows the result of pruning.

Next, curve fitting was performed to further remove redundant points. **Figure 6** is the schematic of curve fitting. Suppose AC is the center line of a branch. The curve AC could be described with segments AB and BC, as the distances of all points on curves AC and BD to these segments would not exceed the preset threshold. Except for points A, B, and C, all the points on curve AC were eliminated as redundant points after fitting (Feng, Kim, Yu, Peng & Hart, 2010; Zeng, Zhao, Zheng & Liu, 2013; Menon, Ma, Hewitt, Science, Styner & Gerig, et al, 2011).

Figure 5b shows the image subjected to curve fitting and thinning successively. After curve fitting, branch data were stored as $\{(x', y', z'), R', (x'', y'', z''), R''\}$, wherein $\{(x', y', z'), R'\}$ and $\{(x'', y'', z''), R''\}$ represent the coordinates and radius of points on two center lines,
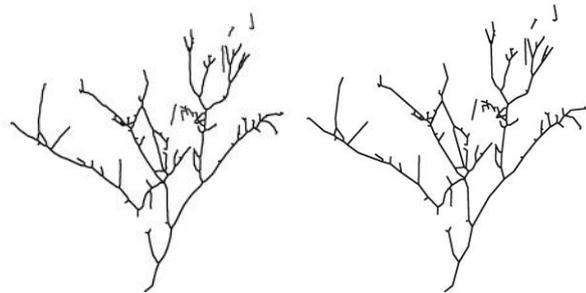
(a) Short branch culling image   (b) Curve fitting image

Figure 5. Branch images after pruning (a) and curve fitting (b)
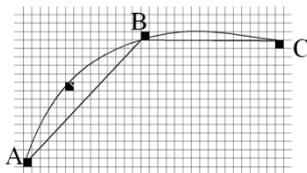
Figure 6. Schematic of curve fitting

respectively. These data were critical for modeling (Lindsay, Maxwell, Rosenberg & Tucker, 2007).

Modeling for branch modules

By extracting the information of key points, we have simplified branch data into the datasets essential for conical frustum (Méndez, Rosell-Polo, Sanz, Escolà & Catalán, 2014) construction. Modularized modeling would be performed based on these datasets (Wright, Guan, Tseng, Cook, Wei & Chang, 2015) in the following section.

Solving of transformation matrix

The fundamental element to describe a model is the vertices of the model. For the branch model in **Figure 2**, its vertices situated on the edges (Côté, Widlowski, Fournier & Verstraete, 2009)) of circles O_1, O_2, O_3, and O_4. As branches are shaped like a cylinder or conical frustum, polygon approximation was adopted for modeling such shape (Meseguer, Lobo, Ree, Beerling & Sanmartín, 2015). In the cylinder modeling by prism approximation, the prism would be closer to a real cylinder if it had more sides. The number of sides of a prism is called transverse precision. In the present work, the transverse precision of models was set as a variable to satisfy varying precision demands (Tang, Dong & Buckles, 2013). Meanwhile, the points of equal division were taken as vertices to guarantee the smoothness of each model.

The following can be known according to the rule of spatial transformation: the cylinder or conical frustum (for which the radii of circles at two ends were R' and R'', and the centers located at $P'(x', y', z')$ and $P''(x'', y'', z'')$ was obtained after two spatial rotations (**Figure**

4040
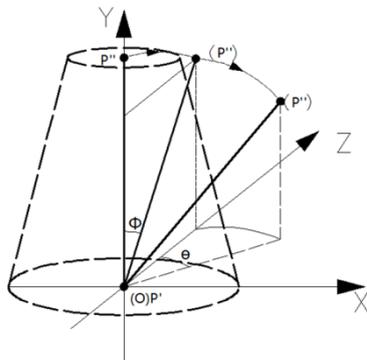
Figure 7. Schematic of spatial transformation of conical frustum

7) and one spatial translation were carried out on the cylinder or conical frustum (Jew, Hendrich & Zhang, 2010) at height *D*, for which the circle of initial end (its center was exactly the origin) was in the plane *XOZ*. Therein, the height *D* is calculated as follows.

$$D = \sqrt{(x' - x'')^2 + (y' - y'')^2 + (z' - z'')^2} \tag{1}$$

According to radii *R'* and *R''*, it can be known that the center of circle located at the origin (Shi, Liang & Liu, 2011; Liang, Gong, Li & Ibrahim, 2014). The matrix *P1* containing the homogeneous coordinates of points equally dividing the circle in plane *XOZ* into *n* equal parts (center of circle: (*0, D, 0*), and its parallel matrix *P2* are described as follows (Raveh & Zaide, 2006).

$$P1 = \begin{bmatrix} R' & 0 & 0 & 1 \\ \cdots & \cdots & \cdots & 1 \\ R'\cos(\frac{2k\pi}{n}) & 0 & R'\sin(\frac{2k\pi}{n}) & 1 \\ \cdots & \cdots & \cdots & 1 \\ R'\cos(\frac{2\pi(n-1)}{n}) & 0 & R'\sin(\frac{2\pi(n-1)}{n}) & 1 \end{bmatrix} \text{ (where } k \in \{0,1,2,\dots,n-1\})$$

$$P2 = \begin{bmatrix} R'' & D & 0 & 1 \\ \cdots & \cdots & \cdots & 1 \\ R''\cos(\frac{2k\pi}{n}) & D & R''\sin(\frac{2k\pi}{n}) & 1 \\ \cdots & \cdots & \cdots & 1 \\ R''\cos(\frac{2\pi(n-1)}{n}) & D & R''\sin(\frac{2\pi(n-1)}{n}) & 1 \end{bmatrix} \text{ (where } k \in \{0,1,2,\dots,n-1\}) \tag{2}$$

First rotation: Rotating by $\phi(0 \leq \phi \leq 180)$ around the *X* axis. There is the following result according to **Figure 7**.

$$\cos\Phi = \frac{y'' - y'}{D}, \quad \sin\Phi = \sqrt{1 - \cos^2\Phi} \tag{3}$$

Second rotation: Rotating by $\theta(0 < \theta < 360)$ around the Y axis. There is the following result according to **Figure 7**.

$$cos\,\theta = \frac{z''-z'}{\sqrt{(x''-x')^2+(z''-z')^2}} \quad sin\,\theta = \frac{x''-x'}{\sqrt{(z''-z')^2+(x''-x')^2}} \tag{4}$$

Translation: The center of the circle at initial end was shifted to (x', y', z').

Matrix *Y* was obtained after the three transformations above.

$$Y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\Phi & -sin\Phi & 0 \\ 0 & sin\Phi & cos\Phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\theta & 0 & sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -sin\theta & 0 & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} cos\theta & 0 & sin\theta & 0 \\ sin\theta\,sin\Phi & cos\Phi & -cos\theta\,sin\Phi & 0 \\ -sin\theta\,cos\Phi & sin\Phi & cos\theta\,cos\Phi & 0 \\ x' & y' & z' & 1 \end{bmatrix} \tag{5}$$

Matrix *Y* was the synthetic transformation matrix of vertex location data.

Solving of vertices of a model

After obtaining the synthetic transformation matrix, calculation was conducted (Sati, Rossignac, Seidel, Wyvill & Musuvathy, 2016) according to the coordinates *{(x', y', z'), R'}* of *P'* and its corresponding radius *R'*, as well as the coordinates *{(x'', y'', z''), R''}* of *P''* and its corresponding radius *R''* (Jacob, Colby, Kabilan, Einstein & Carson, 2013; Kilgard & Bolz, 2012; Mateo, Zaro, Nchez Navarro, Garc & AGil, et al., 2014; Pulli, Baksheev, Kornyakov & Eruhimov, 2012). Thus, the vertex coordinates corresponding to P' can be described as

$$P' = P1 \times Y \tag{6}$$

The coordinates of 12 vertices corresponding to P'' can be described as

$$P'' = P2 \times Y \tag{7}$$

After determining each vertex of a conical frustum, they were linked by the sequence in standard model to form a module eventually.

DISCUSSION, CONCLUSION AND SUGGESTIONS

Modules needed to be assembled after all modules were built (Wang & Pai, 2012; Kim, 2013; Henderson, Gerber, Hilinski, Falcucci, Ojima, & Salvatore, et al., 2015). Before the assembly of modules, the module constructing velocity (Skalski, Townsend & Gilbert, 2015) should be assessed, to achieve a balance between modeling velocity and model precision. Then a suitable model precision was selected to satisfy the demand of real-time modeling and visualization (Jungblut, Vlachos, Schuldt, Zahn, Deller & Wittum, 2012).

The precision of modules is adjustable (Vankipuram, Kahol, Mclaren & Panchanathan, 2010). Hence, in overall branch modeling, we can determine the optimal precision under current hardware conditions (Foley & Hanrahan, 2011; Janoos, Mosaliganti, Xu, Machiraju, Huang & Wong, 2009) by measuring the relationship between module constructing velocity and precision (Bo, Bartoň, Plakhotnik, & Pottmann, 2016). In this article, 194 datasets of key

Table 1. Relationship between model precision and modelling time

Model precision (Number of sides)	6	8	10	12	24	50
Modeling time (ms)	258	341	451	516	1121	2371
Ratio of modeling time to model precision	43	42.6	45.1	43	46.7	47.4

Figure 8. Display of final modeling result from several angles

points were obtained after image processing. The modeling time (Decost, Jain, Rollett, & Holm, 2016) at varying model precision was measured and listed in **Table 1.**

Table 1 shows that modeling time increased linearly with model precision. The model created at the precision of 12 had satisfied human visual demand, and the corresponding modeling time was short enough to achieve the timely obstacle evasion of robots. Hence, we chose 12 as the model precision. **Figure 8** shows the final model, which cost 516 ms.

Using the binocular stereo vision system, the natural scene was photographed to reconstruct the branches of citrus trees in the virtual world. Branch reconstruction was achieved via two steps. Firstly, branch obstacles were segmented from the image and data were optimized by removing redundant points, followed by information extraction on key points. Secondly, modules were constructed according to the information of key points and then assembled to generate a complete model.

Due to the complexity of natural scene, the acquisition of valuable information from pictures requires the assistance of image processing software. The final modeling result heavily depends on data processing. Therefore, the optimization of image processing methods and exploration of more targeted processing methods will be the emphases of future study.

ACKNOWLEDGEMENTS

REFERENCES

Agrawal, A., Sun, Y., Barnwell, J., & Raskar, R. (2010). Vision-guided robot system for picking objects by casting shadows. *International Journal of Robotics Research, 29(2-3), 155-173.* doi:10.1177/0278364909353955

Bo, P., Bartoň, M., Plakhotnik, D., & Pottmann, H. (2016). Towards efficient 5-axis flank cnc machining of free-form surfaces via fitting envelopes of surfaces of revolution. *Computer-Aided Design, 79, 1-11.* doi:10.1016/j.cad.2016.04.004

Côté, J. F., Widlowski, J. L., Fournier, R. A., & Verstraete, M. M. (2009). The structural and radiative consistency of three-dimensional tree reconstructions from terrestrial lidar. *Remote Sensing of Environment, 113(5), 1067-1081.* doi:10.1016/j.rse.2009.01.017

Decost, B. L., Jain, H., Rollett, A. D., & Holm, E. A. (2016). Computer vision and machine learning for autonomous characterization of am powder feedstocks. *JOM, 69(3), 1-10.* doi:10.1007/s11837-016-2226-1

Escamez, G., Sirois, F., Lahtinen, V., Stenvall, A., Badel, A., & Tixador, P., et al. (2016). 3-d numerical modeling of ac losses in multifilamentary mgb2 wires. *IEEE Transactions on Applied Superconductivity, 26(3), 1-7.* doi:10.1109/TASC.2016.2533024

Feng, W. W., Kim, B. U., Yu, Y., Peng, L., & Hart, J. (2010). Feature-preserving triangular geometry images for level-of-detail representation of static and skinned meshes. *Acm Transactions on Graphics, 29(2), 11.* doi:10.1145/1731047.1731049

Fernandes, R., Armada-Da-Silva, P., Pool-Goudaazward, A., Moniz-Pereira, V., & Veloso, A. P. (2016). Three dimensional multi-segmental trunk kinematics and kinetics during gait: test-retest reliability and minimal detectable change. *Gait & Posture, 46, 18-25.* doi:10.1016/j.gaitpost.2016.02.007

Foley, T., & Hanrahan, P. (2011). Spark: modular, composable shaders for graphics hardware. *Acm Transactions on Graphics, 30(4), 107.* doi:10.1145/2010324.1965002

Fu, X., Xu, G. J., Li, Z. J., Du, C. L., Han, Z., & Zhang, T., et al. (2015). Three-dimensional reconstruction modeling of the spatial displacement, extent and rotational orientation of undisplaced femoral neck fractures. *Medicine, 94(39), e1393.* doi:10.1097/MD.0000000000001393

Gonzálezjosé, R., Escapa, I., Neves, W. A., Cúneo, R., & Pucciarelli, H. M. (2008). Cladistic analysis of continuous modularized traits provides phylogenetic signals in homo evolution. *Nature, 453(7196), 775-778.* doi:10.1038/nature06891

Henderson, B. B., Gerber, P. J., Hilinski, T. E., Falcucci, A., Ojima, D. S., & Salvatore, M., et al. (2015). Greenhouse gas mitigation potential of the world's grazing lands: modeling soil carbon and nitrogen fluxes of mitigation practices. *Agriculture Ecosystems & Environment, 207, 91-100.* doi:10.1016/j.agee.2015.03.029

Jacob, R. E., Colby, S. M., Kabilan, S., Einstein, D. R., & Carson, J. P. (2013). In situ casting and imaging of the rat airway tree for accurate 3d reconstruction. *Experimental Lung Research, 39(6), 249-57.* doi:10.3109/01902148.2013.801535

Janoos, F., Mosaliganti, K., Xu, X., Machiraju, R., Huang, K., & Wong, S. T. (2009). Robust 3d reconstruction and identification of dendritic spines from optical microscopy imaging. *Medical Image Analysis, 13(1), 167-179.* doi:10.1016/j.media.2008.06.019

Jew, D. K., Hendrich, N., & Zhang, J. (2010). Multi sensor fusion of camera and 3d laser range finder for object recognition. *IEEE, 236 - 241.* doi:10.1109/MFI.2010.5604459

Jungblut, D., Vlachos, A., Schuldt, G., Zahn, N., Deller, T., & Wittum, G. (2012). Spinelab: tool for three-dimensional reconstruction of neuronal cell morphology. *Journal of Biomedical Optics, 17(7), 076007.* doi:10.1117/1.JBO.17.7.076007

Kempthorne, D. M., Turner, I. W., Belward, J. A., Mccue, S. W., Barry, M., & Young, J., et al. (2015). Surface reconstruction of wheat leaf morphology from three-dimensional scanned data. *Functional Plant Biology, 42(5), 444-451.* doi:10.1071/FP14058

Kilgard, M. J., & Bolz, J. (2012). Gpu-accelerated path rendering. *Acm Transactions on Graphics, 31(6), 1-10.* doi:10.1145/2366145.2366191

Kim, J., & Jeong, I. K. (2014). Single image-based 3d tree and growth models reconstruction. *Etri Journal, 36(3), 450-459. doi:10.4218/etrij.14.0113.0069*

Kim, M. (2013). Gpu isosurface raycasting of fcc datasets. *Graphical Models, 75(2), 90-101.* doi:10.1016/j.gmod.2012.11.001

Letsch, H. O., & Kjer, K. M. (2011). Potential pitfalls of modelling ribosomal rna data in phylogenetic tree reconstruction: evidence from case studies in the metazoa. *BMC Evolutionary Biology, 11(1), 146.* doi:10.1186/1471-2148-11-146

Liang, J., Gong, J., Li, W., & Ibrahim, A. N. (2014). Visualizing 3d atmospheric data with spherical volume texture on virtual globes.*Computers & Geosciences, 68, 81-91.* doi:10.1016/j.cageo.2014.03.015

Lindsay, K. A., Maxwell, D. J., Rosenberg, J. R., & Tucker, G. (2007). A new approach to reconstruction models of dendritic branching patterns. *Mathematical Biosciences, 205(2), 271.* doi:10.1016/j.mbs.2006.08.005

List, G. F., & Mashayekhi, M. (2016). A modular colored stochastic petri net for modeling and analysis of signalized intersections. IEEE Transactions on Intelligent *Transportation Systems, 17(3), 701-713.* doi:10.1109/TITS.2015.2483324

Mateo, L., Zaro, J., Nchez Navarro, J., ngel, Garc, & A Gil, A., et al. (2014). 3d-geological structures with digital elevation models using gpu programming. *Computers & Geosciences, 70(C), 138-146.* doi:10.1016/j.cageo.2014.05.014

Mccraig, M. A., Osinski, G. R., Cloutis, E. A., Flemming, R. L., Izawa, M. R. M., & Reddy, V., et al. (2017). Fitting the curve in excel®; systematic curve fitting of laboratory and remotely sensed planetary spectra. *Computers & Geosciences, 100, 103-114.* doi:10.1016/j.cageo.2016.11.018

Méndez, V., Rosell-Polo, J. R., Sanz, R., Escolà, A., & Catalán, H. (2014). Deciduous tree reconstruction algorithm based on cylinder fitting from mobile terrestrial laser scanned point clouds. *Biosystems Engineering, 124(4), 78-88.* doi:10.1016/j.biosystemseng.2014.06.001

Menon, J., Ma, Y. L. S., Hewitt, W. T., Science, C., Styner, M., & Gerig, G., et al. (2011). Direct trimming of nurbs surfaces on the gpu. *Acm Transactions on Graphics, 28(3), 1.* doi:10.1145/1531326.1531353

Meseguer, A. S., Lobo, J. M., Ree, R., Beerling, D. J., & Sanmartín, I. (2015). Integrating fossils, phylogenies, and niche models into biogeography to reveal ancient evolutionary history: the case of hypericum (hypericaceae). *Systematic Biology, 64(2), 215-32.* doi:10.1093/sysbio/syu088

Morrison, C., Huckvale, K., Corish, B., Dorn, J., Kontschieder, P., & Hara, K., et al. (2016). Assessing multiple sclerosis with kinect: designing computer vision systems for real-world use. *Human-Computer Interaction, 31(3-4), 191-226.* doi:10.1080/07370024.2015.1093421

Mossel, E., Roch, S., & Sly, A. (2011). Robust estimation of latent tree graphical models: inferring hidden states with inexact parameters. *IEEE Transactions on Information Theory, 59(7), 4357-4373.* doi:10.1109/TIT.2013.2251927

Pulli, K., Baksheev, A., Kornyakov, K., & Eruhimov, V. (2012). Real-time computer vision with opencv. *Communications of the Acm, 55(6), 61-69.* doi:10.1145/2184319.2184337

Raveh, D. E., & Zaide, A. (2006). Numerical simulation and reduced-order modeling of airfoil gust response. *Aiaa Journal, 44(8), 1826-1834.* doi:10.2514/1.16995

Sati, M., Rossignac, J., Seidel, R., Wyvill, B., & Musuvathy, S. (2016). Average curve of n smooth planar curves. *Computer-Aided Design, 70(C), 46-55.* doi:10.1016/j.cad.2015.06.017

Schmitt, M., Shahzad, M., & Zhu, X. X. (2015). Reconstruction of individual trees from multi-aspect

Shi, B. Q., Liang, J., & Liu, Q. (2011). Adaptive simplification of point cloud using k k mathcontainer loading mathjax -means clustering.*Computer-Aided Design, 43(8), 910-922.* doi:10.1016/j.cad.2011.04.001

Skalski, J. R., Townsend, R. L., & Gilbert, B. A. (2015). Calibrating statistical population reconstruction models using catch-effort and index data. *Journal of Wildlife Management, 71(4), 1309–1316.* doi:10.2193/2005-707

Song, B., Wang, Z., & Sheng, L. (2016). A new genetic algorithm approach to smooth path planning for mobile robots. *Assembly Automation, 36(2), 138-145.* doi:10.1108/AA-11-2015-094

Štefan Kohek, & Strnad, D. (2015). Interactive synthesis of self-organizing tree models on the gpu. *Computing, 97(2), 145-169.* doi:10.1007/s00607-014-0424-7

Tan, Z., Jamdagni, A., He, X., Nanda, P., Liu, R. P., & Hu, J. (2015). Detection of denial-of-service attacks based on computer vision techniques. *IEEE Transactions on Computers, 64(9), 2519-2533.* doi:10.1109/TC.2014.2375218

Tang, S., Dong, P., & Buckles, B. (2013). Three-dimensional surface reconstruction of tree canopy from lidar point clouds using a region-based level set method. *International Journal of Remote Sensing, 34(4), 1373-1385.* doi:10.1080/01431161.2012.720046

Tomosar data. *Remote Sensing of Environment, 165(2015), 175-185.* doi:10.1016/j.rse.2015.05.012

Urrutia, R. B., Lara, A., Villalba, R., Christie, D. A., Quesne, C. L., & Cuq, A. (2011). Multicentury tree ring reconstruction of annual streamflow for the maule river watershed in south central chile. *Water Resources Research, 47(6), 179-187.* doi:10.1029/2010WR009562

Vankipuram, M., Kahol, K., Mclaren, A., & Panchanathan, S. (2010). A virtual reality simulator for orthopedic basic skills: a design and validation study. *Journal of Biomedical Informatics, 43(43), 661-668.* doi:10.1016/j.jbi.2010.05.016

Wang, Y., Sibeck, D. G., Merka, J., Boardsen, S. A., Karimabadi, H., & Sipes, T. B., et al. (2013). A new three-dimensional magnetopause model with a support vector regression machine and a large database of multiple spacecraft observations. *Journal of Geophysical Research-Space Physics, 118(5), 2173-2184.* doi:10.1002/jgra.50226

Wang, B., & Pai, D. K. (2012). Adaptive image-based intersection volume. *Acm Transactions on Graphics, 31(4), 1-9.* doi:10.1145/2185520.2185593

Wright, W. E., Guan, B. T., Tseng, Y. H., Cook, E. R., Wei, K. Y., & Chang, S. T. (2015). Reconstruction of the springtime East Asian subtropical jet and western pacific pattern from a millennial-length

taiwanese tree-ring chronology. *Climate Dynamics, 44(5-6), 1645-1659.* doi:10.1007/s00382-014-2402-3

Yang, Z., Shen, L. Y., Yuan, C. M., & Gao, X. S. (2015). Curve fitting and optimal interpolation for cnc machining under confined error using quadratic b-splines. *Computer-Aided Design, 66(C), 62-72.* doi:10.1016/j.cad.2015.04.010

Zeng, M., Zhao, F., Zheng, J., & Liu, X. (2013). Octree-based fusion for realtime 3d reconstruction. *Graphical Models, 75(3), 126-136.* doi:10.1016/j.gmod.2012.09.002

Zhang, T., Zhang, R., Yuan, Y., Gao, Y., Wei, W., & Diushen, M., et al. (2015). Reconstructed precipitation on a centennial timescale from tree rings in the western Tien Shan Mountains, central asia. *Quaternary International, 358, 58-67.* doi:10.1016/j.quaint.2014.10.054

http://www.ejmste.com