Three-dimensional dense reconstruction of plant or tree canopy based on stereo vision

Zhijiang Ni, Thomas F. Burks*

(*Department of Agricultural and Biological Engineering, University of Florida, Gainesville, Florida, USA*)

Abstract: Three-dimensional (3D) reconstruction of the plant or tree canopy is an important step to measure canopy geometry, volume, and leaf cover density for applications in precision agriculture, robotic harvesting, or plant phenotype. In this research, binocular stereo vision was used to recover the 3D information of the canopy. A revised camera calibration method was provided to calibrate the cameras in the world coordinate system. Only two images were used to realize a dense reconstruction. These two images were firstly rectified to make sure the corresponding feature points in the left and right images were on the same horizontal line. An efficient large-scale stereo matching (ELAS) algorithm was used to find the disparity map. The plant or tree canopy was finally reconstructed based on these calibrated camera matrices and the disparity map through a triangulation method. In this research, a series of laboratory experiments were conducted to validate the 3D reconstruction and verify the dimensional accuracy of the reconstructed croton plant's large leaves. The measurements of the reconstruction were compared to manual measurement showing that this reconstruction was metric reconstruction. In addition, two reconstructions were completed based on a side view of the croton plant, and a top view of the croton plant. This was followed by a series of field experiments under uncontrolled lighting conditions in a citrus grove to demonstrate the algorithm's robustness in ambient conditions. Two reconstructions were completed based on a top view of a branch of a mature citrus tree, and a side view of a small citrus tree. All four reconstructions gave a good 3D visualization of the objects. The outcomes of this research have demonstrated the potential for metric reconstruction of 3D tree canopy using a traditional stereo vision camera pair, which provides a low-cost alternative that can operate on normal daylight conditions as compared to other approaches using LIDAR or Kinect RGB-D cameras, or TOF cameras.

Keywords: 3D reconstruction, canopy reconstruction, computer vision, precision agriculture, stereo vision, metric reconstruction

Citation: Ni, Z., and T. F. Burks. 2018. Three-dimensional dense reconstruction of plant or tree canopy based on stereo. Agricultural Engineering International: CIGR Journal, 20(2): 248–260.

1 Introduction

Canopy width and height, and total surface area are important measurements for precision agriculture. Destructive and non-destructive methods are available to do this type of measurement. Usually, destructive method involved manually removing each leaf and manually doing measurements. This job could be time consuming and tedious. Ultrasonic sensor, laser, Bumblebee stereo camera, Microsoft Kinect RGB-D camera, and time-of-flight camera could be used to do 3D reconstruction of the target (Schumann et al., 2005; Song et al., 2011; Tumbo et al., 2002; Wang and Zhang, 2013; Wei and Salyan, 2004; Xie et al., 2010). Then the results of 3D reconstruction could be used in non-destructive methods. However, these sensors are expensive and have difficulty in daylight condition. Use two regular webcams as binocular stereo cameras to do 3D reconstruction is a more-cost effective way. Scharstein and Szeliski (2002) summarized and evaluated different dense two-frame stereo correspondence algorithms. Different optimization methods were compared, such as winner-take-all, graph cuts, dynamic programming, and cooperative algorithm. All of these methods worked only at pixel level. To obtain sub-pixel level accuracy and make the disparity

Received date: 2017-05-15 **Accepted date:** 2018-03-29

*** Corresponding author: Thomas F. Burks,** Professor, Department of Agricultural and Biological Engineering, University of Florida, 225 Frazier Rogers Hall, Gainesville, FL 32611-0570, USA, Email: tburks@ufl.edu. Tel: 352-392-1864 ext.225.

map look continuous, Birchfield and Tomasi (1998) calculated disparity by pixel-to-pixel stereo.

To realize a real-time computation, Geiger et al. (2010) provided a method called Efficient Large-Scale Stereo (ELAS). This could be realized on a single CPU, without GPU (Graphics Processing Unit) technology. Andersen et al. (2005) investigated the use of stereo vision on 3D analysis of plants and estimation of geometric properties. Simulated annealing method was used to find the dense matches. Plant height and leaf area of 10 young wheat plants under laboratory conditions were tested.

Song et al. (2007) created a surface model of a plant from images taken by stereo cameras. The images of the plants were taken from top view. Three different optimization methods for stereo matching were investigated: 1) Pixel-to-Pixel method, 2) Graph cut method, and 3) Multi-resolution method. The height of the plant calculated by these three methods was compared to manually measured height. Then the surface of the plant was modeled by using Self-Organizing Map (Kohonen et al., 1996). Song et al. (2011) provided a non-destructive leaf area measurement by using both stereo and ToF cameras. Two images from stereo cameras and one image from ToF camera were provided. For the experiment, one leaf was reconstructed and measured its area after segmented manually. And only four leaves of used 44 plants in the foreground were used to calculate the leaf area for testing.

The objectives of this research were: 1). Use stereo vision to make a 3D metric reconstruction of the object, 2) Confirm that the reconstruction shows the real size of the object.

In this paper, a revised 3D reconstruction method using stereo vision was introduced. Camera matrices for left and right cameras were calibrated in the world coordinate system. An image rectification method based on projective transformation was introduced. And a reliable and equally spread feature matching method was used to detect and match the key points used in image rectification. Then a fast disparity calculation method called ELAS was used to find the dense matches between rectified left and right images. Finally, the scene was reconstructed through triangulation method. In experimental results and discussion section, four tests in the lab and two tests in the field were performed.

2 Materials and methods

In this paper, two Microsoft LiftCam Studio web cameras were used as the stereo camera. They were assembled inside a wooden box, as shown in Figure 1.

a. Whole view

b. Inside view

Figure 1 Setup of stereo camera

2.1 Stereo camera calibration

Camera matrix is composed of intrinsic parameters and extrinsic parameters. The purpose of camera calibration is to pre-determine camera matrix. The most used calibration method was provided by Zhang (2000), a checkerboard was used as a pattern. At least five images of the checkerboard would be taken at different directions. The checkerboard was assumed to be located on z plane. Different checkerboard has its own z plane, so the 3D coordinates of the corners were known as well as 2D images. The intrinsic and extrinsic parameters would be solved by minimizing the errors between actual 3D points and estimated 3D points. Bouguet (2008) provided a Matlab toolbox to realize this calibration for single camera or stereo camera. Through this method, camera intrinsic and extrinsic parameters were obtained. However, the extrinsic parameters were not related to the same world coordinate system. A 3D world coordinate

system was built to solve this problem, which consisted of a 2D x-z coordinate system plotted on a A0 paper (Figure 2, this figure was extended from the original scale in -z direction), and a vertical checkerboard (Figure 3) provided the y coordinates. Each line in this 2D coordinate system was at 50mm spacing. The line marked with numbers from 1 to 15 was rotated -10^0 around O_{orig} from the middle line, and the line marked with numbers from 31 to 45 was rotated 10^0 around O_{orig} from the middle line. The size of the squares in the checkerboard is 50 mm × 50 mm.

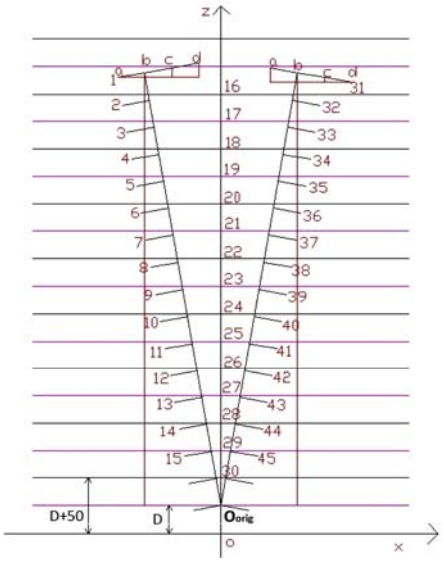


Figure 2 2D x-z coordinate system

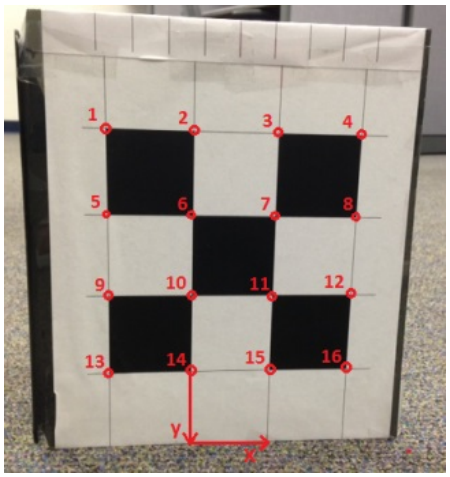


Figure 3 Vertical checkboard providing y coordinates

Then the checkerboard was placed at different locations along the left, middle and right lines. The actual 3D coordinates (X_{act}) for each corner at different locations could be solved. All 45 left images and 45 right images were taken by the stereo camera. The whole system is shown in Figure 4, which includes stereo cameras, checkerboard, and 2D x-z coordinate system. The 2D coordinates for corner could be obtained from these 2D images.

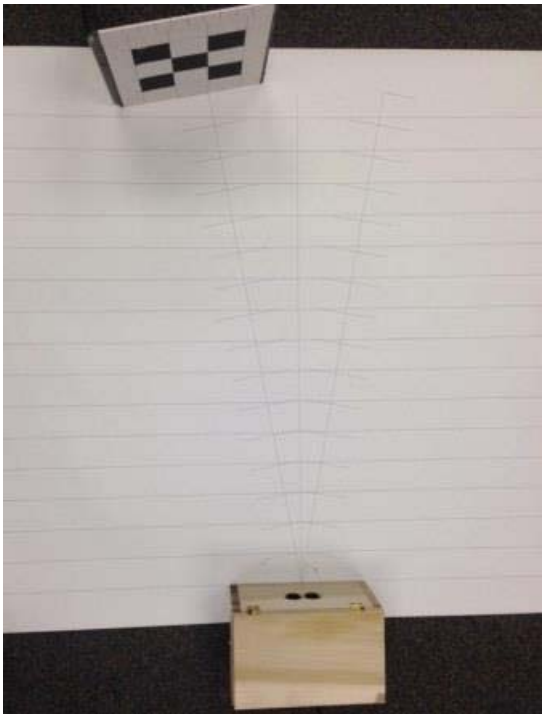


Figure 4 The setup of camera calibration system

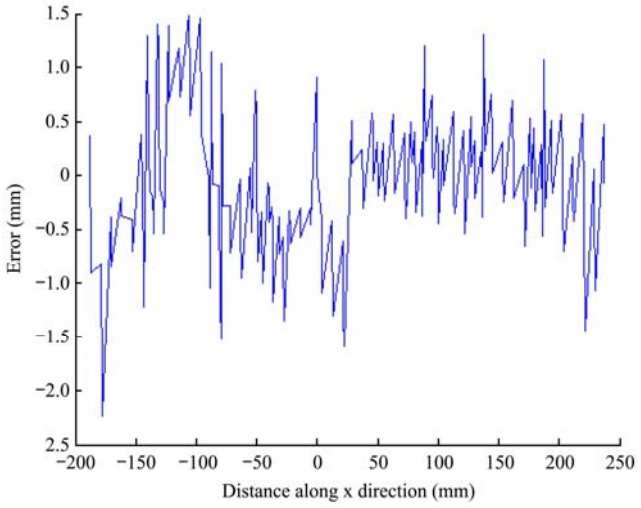
A method called Gold Standard algorithm was used to solve camera matrix from 2D and 3D correspondences (Hartley and Zisserman, 2003). The left camera matrix and right camera matrix were:

$$P = \begin{bmatrix} 654.5363 & 7.5381 & 317.7192 & -1.2018e+04 \\ -7.7905 & 630.1475 & 261.8939 & 4.6008e+04 \\ 0.0470 & -0.0374 & 0.9982 & 14.6055 \end{bmatrix},$$

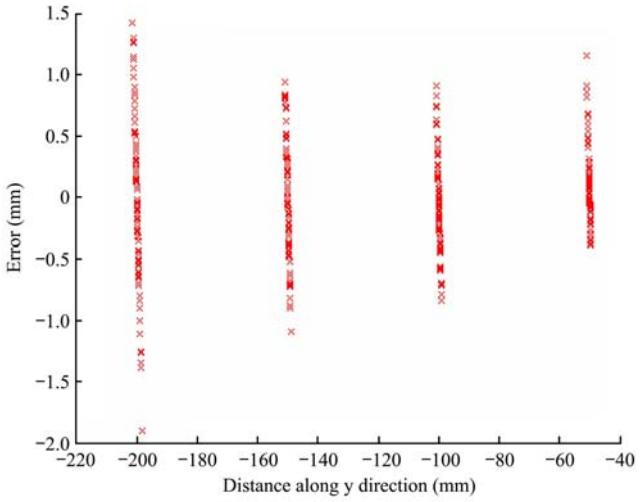
$$P_R = \begin{bmatrix} 662.0958 & 3.7496 & 301.5776 & -3.2281e+04 \\ 3.4730 & 635.9826 & 257.2801 & 4.4712e+04 \\ 0.0628 & -0.0279 & 0.9976 & 13.2096 \end{bmatrix}.$$

The estimated 3D points (X_{est}) are calculated from the camera matrix and 2D projections through triangulation method (Harris and Stephens, 1998). The reconstruction error was calculated from X_{act} - X_{est}. The errors in X, Y, and Z directions are within 3 mm, 2 mm and 9 mm when the distance is less than 800 mm in Figure 5. The statistical analysis of errors is shown in Table 1. The mean error in x, y and z direction is 0.42 mm, 0.36 mm and 2.78 mm.

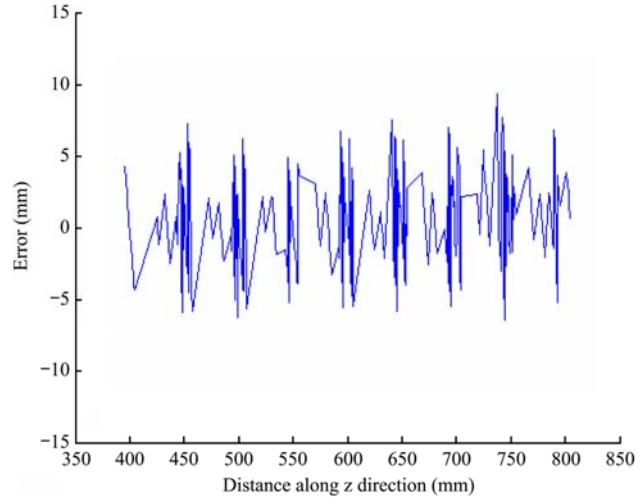
Camera centers ($c = -R^T \cdot t$) for both left and right images were obtained from camera matrix. The left and right cameras are shown in the world coordinate system as Figure 6. The results of 3D reconstruction will be based on this system.



a. Error in x direction



b. Error in y direction



c. Error in z direction

Figure 5 Errors in x, y and z direction

Table 1 Statistical analysis of errors between estimated and actual corners

Axis	Mean absolute error (mm)	Standard Deviation (mm)
X	0.42	0.35
Y	0.36	0.31
Z	2.78	1.74

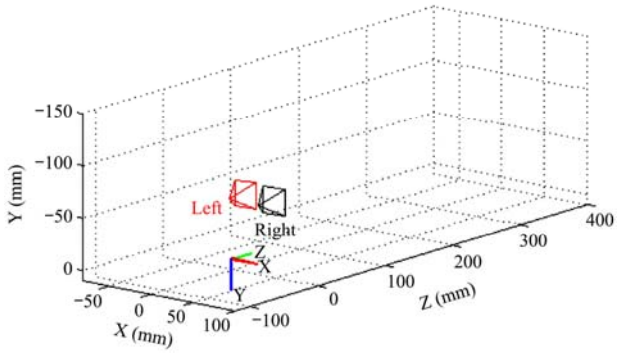


Figure 6 World coordinate system with left and right cameras

2.2 Feature points detection and matching

Feature points detection and matching methods include corner detector, blob detector, Laplacian-of-Gaussian (LoG) detector, etc (Harris and Stephens, 1998; Lindeberg, 1993, 1998). Based on these researches, Harris Affine detector and Hessian Affine detector were introduced (Mikolajczyk and Schmid, 2002, 2004). Experiment results showed that Hessian Affine detector outperformed Harris Affine detector (Mikolajczyk et al., 2005).

Based on LoG detector (Lindeberg, 1998), Lowe (2004) presented a method called Scale Invariant Feature Transform (SIFT) to detect and match image features. Difference-of-Gaussian (DoG) was used to approximate the scale-normalized LoG. The determinant and trace of Hessian matrix of DoG were used to detect features. To speed up SIFT, SURF (Speeded Up Robust Features) was introduced (Bay, 2006; Bay et al., 2008). Experiment results showed that SIFT outperformed SURF in the match ratio and the total number of correct matches, but SURF produced more correct matches per time interval, meaning SURF was faster in computation (Bauer et al., 2007).

Based on the original SIFT was scale invariant (Lowe, 2004), affine invariant SIFT (ASIFT) was introduced (Morel and Yu, 2009). ASIFT was invariant in the change of rotation, translation, zoom, and two angles defining the camera axis orientation. ASIFT can provide many more matches while took more time for computation than SIFT.

Sometimes, the sparse feature points were not enough to provide good matches, especially when only salient features were detected and matched. For those images without salient features, the detected features would be

only shown in some regions of the images, not evenly distributed over the images. The feature points generated by using quasi-dense matching method (qdmatch) can handle this problem (Lhuillier and Quan, 2005). This method used SIFT as seeds, the features were re-sampled after seeds propagated to generate quasi-dense matches. For two images, the image was subdivided into small regions, and an affine transformation will be fitted between corresponding sub-regions, the center point in the left sub-region and the affine transformed point in the right image will be matched.

To determine which method is the best among these six methods, Harris affine, Hessian affine, SIFT, ASIFT, SURF, and qdmatch, comparisons were made among the time to detect features, the detected number of feature points, the time to do the matching, the number of matches between two images, and the distribution of feature points over the whole image (Mikolajczyk and Schmid, 2002, 2004; Lowe, 2004; Morel and Yu, 2009; Bay, 2006; Lhuillier and Quan, 2005). The left image was divided into 8×8 square grids. The grid will be tagged as used if one or more feature points were in a grid. The feature points' distribution was defined as: the number of used grids divided by the total number of grids.

Three sets of images were compared, the first set of images was taken from a potted aloe plant, which only occupied a small portion of the whole image (Figure 7), the second set of images is taken from a potted banyan tree, which has small leaves (Figure 8), and the last set of images is taken from a potted croton plant, which has big leaves and salient features (Figure 9).

The testing code for Harris affine and Hessian affine was given by Robotics Research Group (Robotics Research Group, 2007). The code of SIFT was provided by Lowe (2005). The ASIFT code was given by Yu and Morel (2011) . The SURF code was provided by Bay (2006). The quasi-dense matching code was provided by Xu (2004), and a revised version using GPU in this paper. The test was performed on Ubuntu 12.04 on Lenovo IdeaPad 500 with Intel Core i7-3630QM CPU and GeForce GT 650M GPU. The comparisons results of aloe plant, banyan tree and croton plant were shown in Table 2, Table 3 and Table 4 indicated that qdmatch has the best distribution, which followed by ASIFT. The computing time of qdmatch outperformed the one of ASIFT.

a. Left image b. Right image

Figure 7 Aloe plant

a. Left image b. Right image

Figure 8 Banyan tree

a. Left image b. Right image

Figure 9 Croton plant

Table 2 Comparisons of different features of 'aloe'

Methods	t1 (s)	num1	t2 (s)	num2	t (s)	propagation (s)	resample (s)	num	Distribution
Harris affine	0.27	279	0.27	349	0.03	/	/	31	0.42%
Hessian affine	0.17	344	0.16	298	0.03	/	/	36	0.58%
SIFT	1.25	3284	1.28	3384	0.23	/	/	609	9.54%
ASIFT	15.04	20281	16.70	25704	1.56	/	/	2377	18.98%
SURF	0.23	401	0.24	447	0.09	/	/	100	2.00%
qdmatch[1]	0.15	5705	0.07	4823	0.06	2.76	3.40	1624	31.08%

Note: t1 is the time to get feature points in the left image; t2 is time to get feature points in the right image; num1 is the number of feature points in the left image; num2 is the number of feature points in the right image; t is time to match the feature points between left and right images; propagation and resample time are only used in qdmatch method; num is the matched feature points between left and right images; distribution is used to measure how feature points are distributed among the image.

[1] This qdmatch used SiftGPU to do sparse matching, and then used GPU to do parallel computation on resampling.

Table 3 Comparisons of different features of 'banyan'

Methods	t1 (s)	num1	t2 (s)	num2	t (s)	propagation (s)	resample (s)	num	distribution
Harris affine	0.36	1137	0.35	1028	0.08	/	/	19	0.31%
Hessian affine	0.28	1367	0.28	1379	0.09	/	/	34	0.54%
SIFT	1.17	2307	1.34	3467	0.17	/	/	148	2.35%
ASIFT	18.38	3118	18.35	31358	2.41	/	/	663	6.73%
SURF	0.39	808	0.38	752	0.16	/	/	64	1.25%
qdmatch[1]	0.10	3772	0.08	4812	0.06	3.32	3.82	1423	27.25%

Note: t1 is the time to get feature points in the left image; t2 is time to get feature points in the right image; num1 is the number of feature points in the left image; num2 is the number of feature points in the right image; t is time to match the feature points between left and right images; propagation and resample time are only used in qdmatch method; num is the matched feature points between left and right images; distribution is used to measure how feature points are distributed among the image. [1] This qdmatch used SiftGPU to do sparse matching, and then used GPU to do parallel computation on resampling

Table 4 Comparisons of different features of 'croton'

Methods	t1 (s)	num1	t2 (s)	num2	t (s)	propagation (s)	resample (s)	num	distribution
Harris affine	0.36	989	0.36	1006	0.06	/	/	110	1.73%
Hessian affine	0.29	1131	0.32	1306	0.07	/	/	155	2.63%
SIFT	1.12	2092	1.11	2012	0.13	/	/	451	6.88%
ASIFT	16.63	26960	17.03	27972	2.06	/	/	4386	22.88%
SURF	0.49	1039	0.51	1055	0.22	/	/	326	6.27%
qdmatch[1]	0.11	2418	0.05	2500	0.02	2.03	3.30	1546	28.08%

Note: t1 is the time to get feature points in the left image; t2 is time to get feature points in the right image; num1 is the number of feature points in the left image; num2 is the number of feature points in the right image; t is time to match the feature points between left and right images; propagation and resample time are only used in qdmatch method; num is the matched feature points between left and right images; distribution is used to measure how feature points are distributed among the image. [1] This qdmatch used SiftGPU to do sparse matching, and then used GPU to do parallel computation on resampling.

2.3 Image rectification

The two-view epipolar geometry is shown in Figure 10 (Hartley and Zisserman, 2003). Here $\vec{X}$ is the 3D point expressed in homogenous form: $\vec{X}=(X, Y, Z, 1)^T$, and $\vec{x}$ is the 2D point expressed in homogenous form: $\vec{x}=(x, y, 1)^T$. $\vec{C}_L$ and $\vec{C}_R$ are camera centers of left and right cameras. The line defined by $\vec{C}_L$ and $\vec{C}_R$ is called baseline. $\vec{x}_L$ and $\vec{x}_R$ are the projections of the 3D point $\vec{X}$ on the left and right images, respectively. The plane defined by $\vec{C}_L$, $\vec{C}_R$, and $\vec{X}$ is called epipolar plane. $\vec{e}_L$ and $\vec{e}_R$ are called epipole, where $\vec{e}_L$ is the projection of the optical center of the right camera in the left image, and $\vec{e}_R$ is the projection of the optical center of the left camera in the right image. Given a projection $\vec{x}_L$ in the left image, the corresponding point $\vec{x}_R$ in the right image will be constrained on an epipolar line, which is the intersection of epipolar plane and the right image plane. The similar definition works for the epipolar line of $\vec{x}_R$ in the left image plane.

When both the epipoles are at infinity, the epipolar lines will be parallel to the baseline in both left and right images. The procedure of changing epipolar lines to parallel positions is called image rectification, so the searching of correspondence is limited on a horizontal line, which is 1-D search.

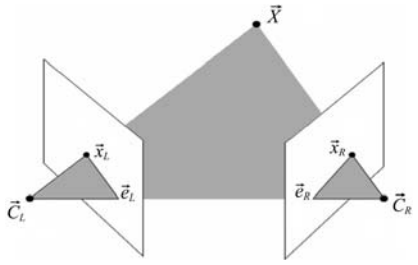


Figure 10 Two-view epipolar geometry (Harris and Stephens, 1998)

A compact algorithm (Fusiello, 2000) was introduced to do image rectification. The stereo cameras should be pre-calibrated. The camera matrices before rectification were called old cameras. The camera matrices after rectification were called new cameras. The camera centers and intrinsic parameters were the same. Only the rotation was changed, which was expressed as $R=\begin{bmatrix}\vec{r}_1^T\\ \vec{r}_2^T\\ \vec{r}_3^T\end{bmatrix}$.

Another algorithm called projective rectification didn't require pre-computed camera matrices (Hartley,

1999). Fundamental matrix (F) for two-view geometry was required. F could be solved through RANSAC algorithm (Hartley and Zisserman, 2003). The feature points used in RANSAC algorithm were from quasi-dense matching algorithm (Lhuillier and Quan, 2005).

The target of projective rectification was to find a projective transformation of an image which transferred the epipole to a point at infinity $(1, 0, 0)^T$. The procedure was summarized as following:

1. Compute the epipoles $\vec{e}_L$ and $\vec{e}_R$ for two images;

2. Compute the projective transformation t_R which maps the epipole $\vec{e}_R$ to the point at infinity;

3. Find the corresponding projective transformation t_L which minimizes the least squares distance $\sum_i d(t_L\vec{x}_{L,i}, t_R\vec{x}_{R,i})$.

The results are shown in Figure 11. Figure 11a) shows the original left image with three feature points on it and Figure 11b) shows the original right image with three unparallel epiplolar lines corresponding to the three feature points in left image. Figure 11c) shows the rectified left image with transformed three feature points. Figure 11d) shows the rectified right image with three corresponding epipolar lines, which are parallel after image rectification.

a. Original left image b. Original right image

c. Rectified left image d. Rectified right image

Figure 11 The results of image rectification

2.4 Disparity calculation

Once images are rectified, the search for correspondences in the other image will be constrained on a horizontal line. For a point $(x,y)^T$ in the left image, the corresponding matched point in the right image is $(x–d,y)^T$. The difference (d) between the x coordinate of a point in the left image and the matched point in the right image is called disparity, which is illustrated in Figure 12. For a feature point in the left image, the matched feature point in the right image will be located in the same horizontal line (the blue line). The matched feature point is the intersection of the blue and green line in the right image of Figure 12.

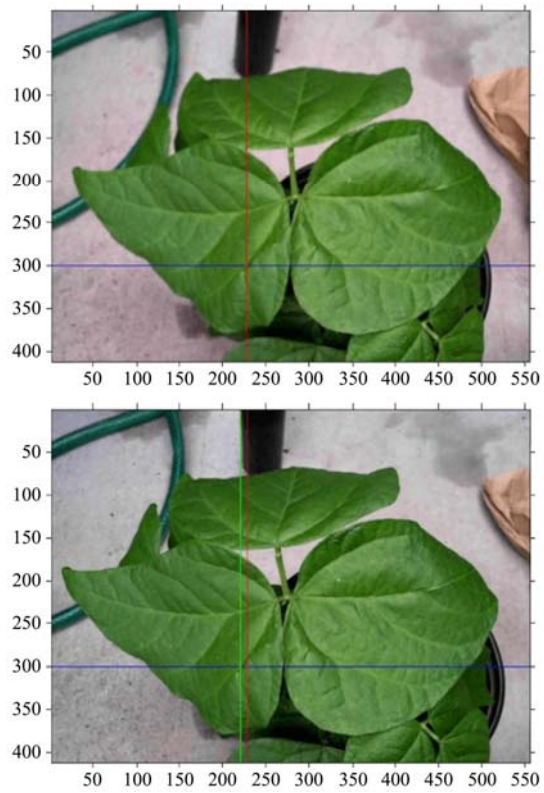


Figure 12 Disparity for rectified images

The Efficient Large-Scale Stereo matching algorithm is used to calculate the disparity for each pixel, which is also called dense matching (Geiger et al., 2010). A value which is less than 0 means non- matching in this algorithm, and it doesn't require pre-knowledge of the largest disparity, therefore, the resulting disparity is at sub-pixel level. The most important property is its fast computation, which can be used for real-time matching.

2.5 Matches in the original images

After image rectification and disparity calculation, the matches between left and right images were calculated. These matches are now expressed in the coordinate system of rectified images. The matches should be mapped back to the coordinate system of the original

images in order to do 3D reconstruction. Inverse projective transformation will be applied both on the left and right rectified matches.

2.6 Image segmentation

In order to make a clear reconstruction of the target, the image was segmented to separate the object from the background. An interactive foreground extraction algorithm (GrabCut) using Iterated Graph Cuts (Rother et al., 2004) was provided. The matlab code (Irena and Aviad) was also available.

By using this method, the user only needed to draw a rough region to mark the foreground, then the foreground would be separated from the background.

2.7 3D reconstruction

For a pinhole camera, the 2D projection $\vec{x}$ and 3D point $\vec{X}$ are related through camera matrix as $\vec{x} = P\vec{X}$. For the left image, we have $\vec{x}_L = P_{left}\vec{X}$. And for right image, we have $\vec{x}_R = P_{right}\vec{X}$. From Figure 10, we can see that the left camera center ($\vec{C}_L$) and the 2D projection ($\vec{x}_L$) form a line, and the right camera center ($\vec{C}_R$) and the 2D projection ($\vec{x}_R$) form another line. These two lines would intersect at a common point ($\vec{X}$), if there wasn't noise or error. This 3D point can be solved through triangulation method (Hartley and Zisserman, 2003).

2.8 Algorithm

The algorithm for stereo dense reconstruction is summarized in Figure 13.

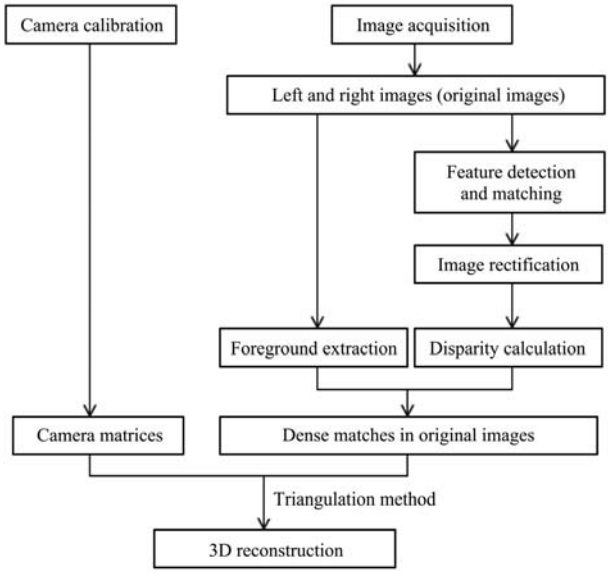


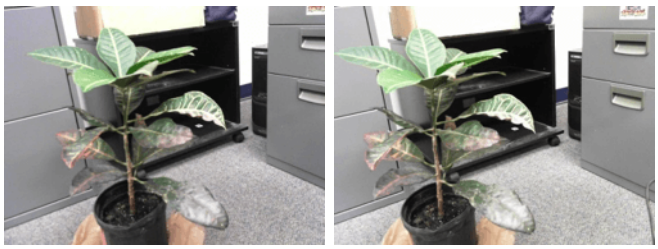
Figure 13 The algorithm for stereo dense 3D reconstruction

3 Results and discussion

Test 1 (Laboratory Test on Potted Plant): In this test, a plant called croton is used to build the 3D reconstruction. Images of two separate leaves, a side view, and a top view are taken for reconstruction trials. The original stereo-view images are shown in Figure 14, where the images in the left column are taken from the left camera, and the images in the right column are taken from the right camera. The 1st leaf is shown in row 1, the 2nd leaf in row 2, the side view in row 3, and the last row is the top view.

a. Left and right images of leaf1

b. Left and right images of leaf2



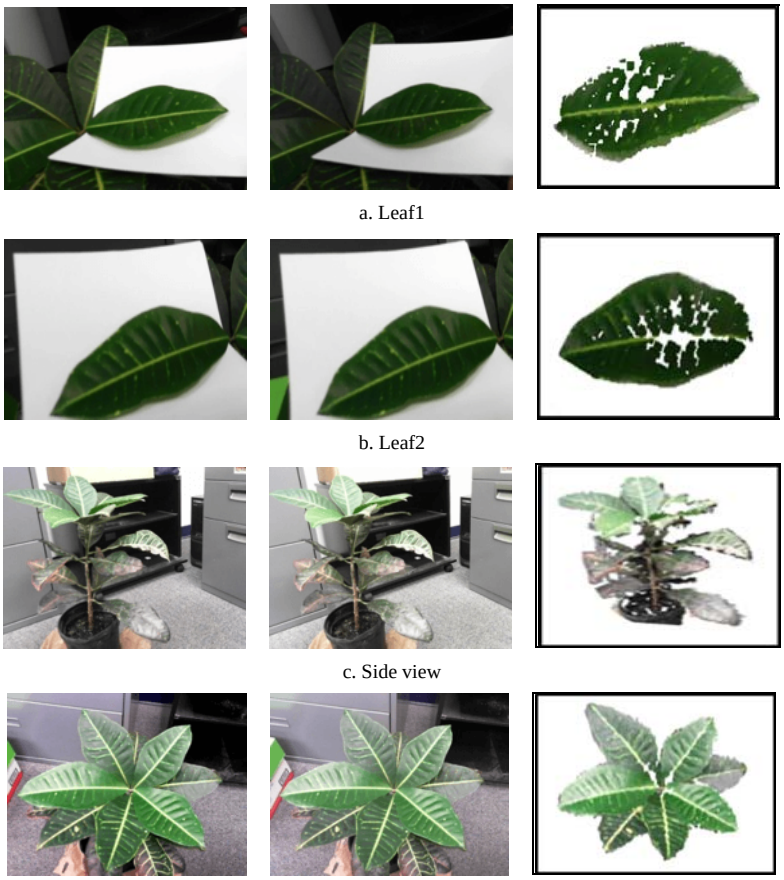
c. Left and right images of side view

d. Left and right images of top view

Figure 14 The original left and right images for laboratory test

Once the images are taken, image reconstructions are generated using the steps described in the algorithm section (Figure 13). The results of these steps are shown in Figure 15, where the rectified left and right images are shown in the first and the second columns, and the

reconstructions are shown in the last column of Figure 15. There are some holes in the leaves because of some mismatches, but overall it has a reproduction of original 3D plant.



a. Leaf1

b. Leaf2

c. Side view

d. Top view

Figure 15 Rectified left image, rectified right image, and reconstructed 3D result for leaf1, leaf2, side view, and top view

Test2 (Field Test on Citrus Tree): In this experiment, image scenes are taken from citrus trees in the field. During this stage of the development process, it found that the web-cameras has a limited field of view and that it will be best to limit the target scene. The first pair of images are taken from a branch segment of a mature citrus tree. The original left and right camera images are shown in Figure 16, and the well reconstructed of leaves are shown in Figure 17.

a. Original left image b. Original right image

Figure 16 A branch of a mature citrus tree

Then a stereo pair of images from the side view of a small citrus tree is taken. The original images are shown in Figure 18, then these two images are rectified and the scene is reconstructed. The 3D well reconstructed result of the small citrus tree is shown in Figure 19.

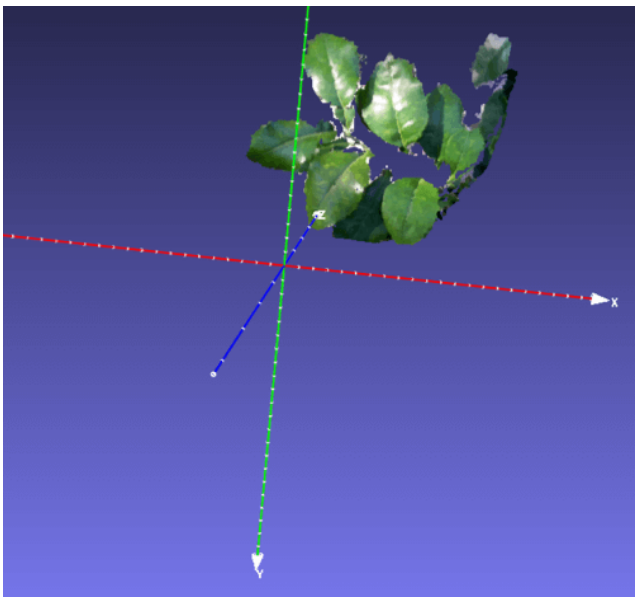


Figure 17 3D reconstructed citrus branch

To measure the estimated width and length of croton leaf from the 3D reconstructed result, Experiments were conducted on two marked croton leaves (Figure 20).

a. Original left image b. Original right image

Figure 18 A small citrus tree in the field

Figure 19 3D reconstructed small citrus tree

a. Left image of leaf1 b. Right image of leaf1

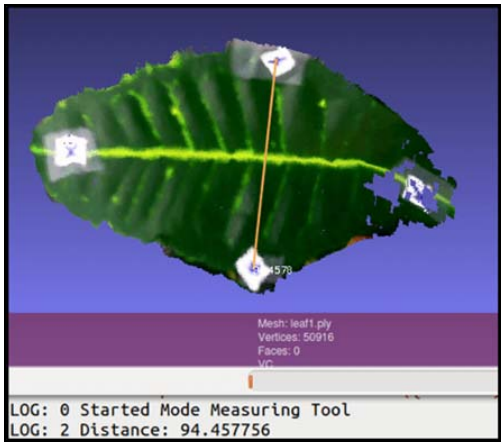
c. Left image of leaf2 d. Right image of leaf2

Figure 20 Left and right images of two experimental leaves

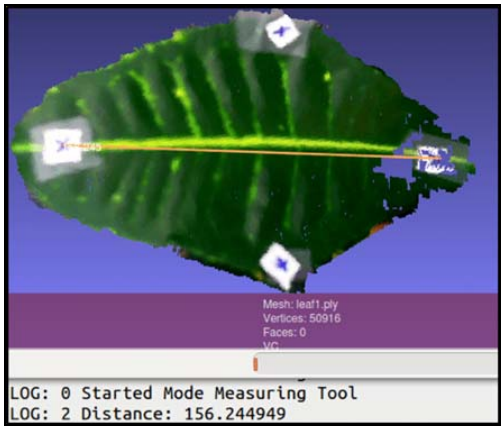
After 3D reconstruction, the width and length of these two leaves can be measured in Meshlab (a free source software, available at http://meshlab.sourceforge.net/). The width and length of leaf1 is 94.46 mm and 156.24 mm, the width and length of leaf2 is 87.04 mm and 144.00 mm (Figure 21). The actual width and length are measured using a plastic ruler, and are shown in Figure 22. The actual width and length of leaf1 is 97.0 mm and 160.0 mm, the actual width and length of leaf2 is 86.0 mm and 145.0 mm, which indicated that results of the 3D reconstruction and the hand measurement are very similar. When comparing these results with other reconstruction methods, this approach is identified as a metric reconstruction method (Kohonen et al., 1996; Schumann and Zaman, 2005).

The results from two experiments indicated that the objects were well reconstructed using only two images. One of the strengths of a good feature matching algorithm is that, a good basis for dense matching. The employed algorithm called quasi-dense matching (Lhuillier and Quan, 2005), which provides an evenly distributed set of feature points, comparing favorably with SIFT, a popular and robust sparse feature matching algorithm (Lowe, 2004).

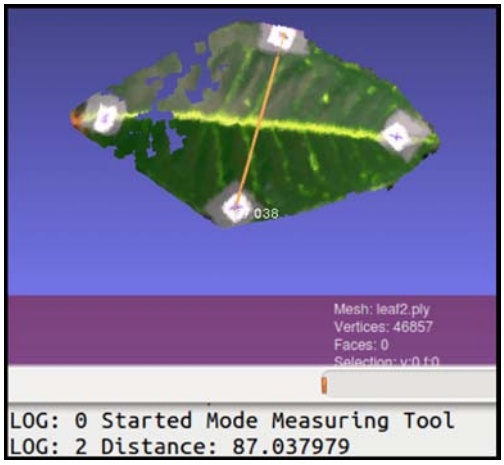
The limitations of this approach are that stereo vision is insufficient, of itself, to insure a complete 3D reconstruction of the plant. Leaf occlusions and smooth features may cause inaccurate disparity, which limits feature point selection, it indicated that some partial leaves were reconstructed due to occlusion in the third row (side view of croton) of Figure 15. In addition, in the last row (top view of croton) of Figure 15, one of the rightmost leaves wasn't accurately reconstructed, may because there are an insufficient number of strong features found during disparity calculation to insure an accurate 3D reconstruction.



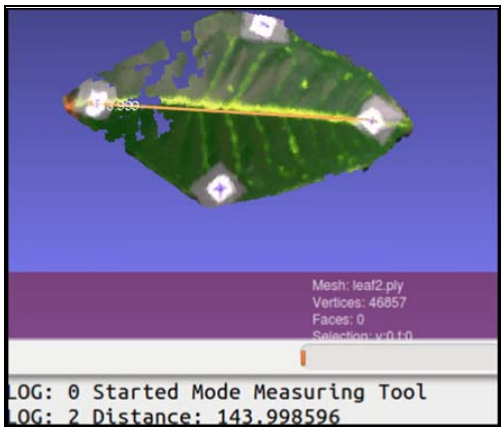
a. Measured width of leaf1



b. Measured length of leaf1



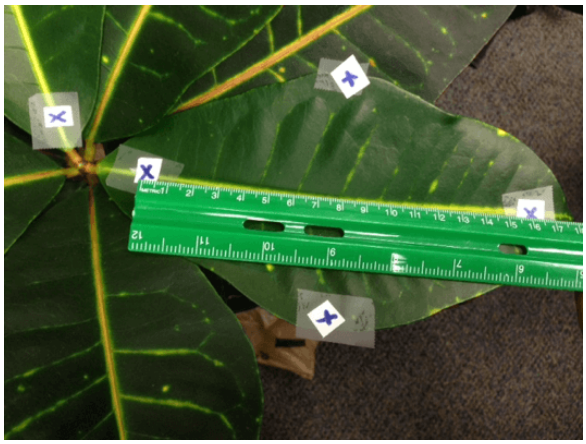
c. Measured width of leaf2



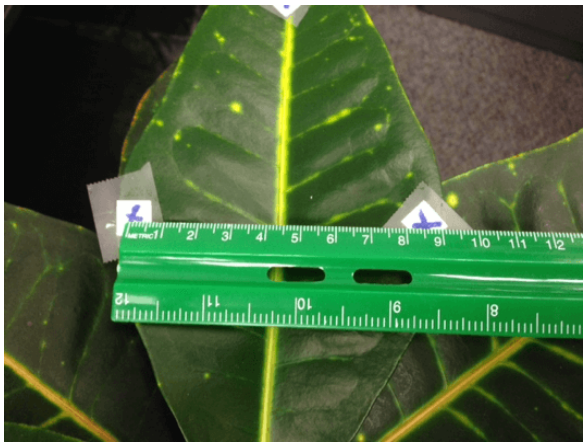
d. Measured length of leaf2

Figure 21 Measured width and length of two leaves in Meshlab

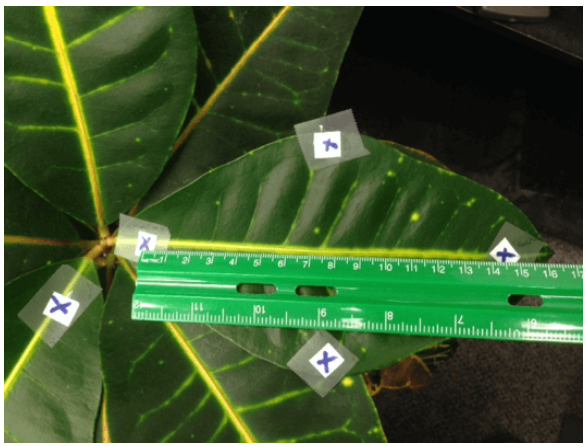
a. The width of leaf1 is 97.0 mm



b. The length of leaf1 is 160.0 mm



c. The width of leaf2 is 86.0 mm



d. tThe length of leaf2 is 145.0 mm

Figure 22 The actual width and length of leaf1 and leaf2

4 Summary and conclusions

In this research, a series of laboratory experiments were conducted to validate the 3D reconstruction and verify the dimensional accuracy of the reconstructed croton plant's large leaves. The comparison of measurements from the reconstruction and manual measurement showing that this reconstruction was metric reconstruction. In addition, two reconstructions were completed based on a side view of the croton plant and a top view of the croton plant, this was followed by a series of field experiments under uncontrolled lighting conditions in a citrus grove to demonstrate the algorithm's robustness in ambient conditions. Then two reconstructions were completed based on a top view of a branch from a mature citrus tree and a side view of a small citrus tree. The results show that all four reconstructions gave a good 3D visualization of the objects.

In this research, a 3D reconstruction method based on two-view dense stereo vision was introduced, and a camera calibration method based on the world coordinate system was utilized, which is applying the individual camera matrix to 3D reconstruction, the results could be projected onto and then measured in the same coordinate system. An image rectification method employing projective transformation was used with a quasi-dense matching algorithm to find evenly distributed feature points. A fast disparity calculation method was used to find dense matched features so that the canopy could be reconstructed through a triangulation algorithm. The final reconstructed scene was a metric reconstruction, which represented the actual dimension of the object. Three-dimensional reconstruction based on stereo vision is a kind of passive reconstruction. In some regions of the target image, insufficient features were found using stereo matching. This may have been due to leaf occlusion, highly saturated reflection, as well as other reasons.

3D reconstruction of plant or tree canopy is an important step to measure canopy geometry, volume, and leaf cover density for applications in precision agriculture, robotic harvesting, or plant phenotype. The outcomes of this research have demonstrated the potential for metric reconstruction of 3D tree canopy using a traditional stereo vision camera pair, which provides a lower cost alternative that can operate during normal daylight conditions as compared to other approaches using LIDAR, Kinect RGB-D cameras, or TOF cameras. This approach can be adopted for most user applications, and has the potential to generate full view reconstruction using a multi-perspective viewing approach.

In future work, images of multiple views could be taken and then merged together to form a full 360 degree reconstruction, this approach might minimize the negative influence of occlusions.

Acknowledgements

The authors would like to acknowledge the contribution to this research provided by sponsors University of Florida, Citrus Initiative and the USDA NIFA AFRI National Robotics Initiative #2013-67021-21074.

References

Andersen, H. J., L. Reng, and K. Kirk. 2005. Geometric plant properties by relaxed stereo vision using simulated annealing. *Computers and Electronics in Agriculture*, 49(2): 219–232.

Bauer, J., N. Sünderhauf, and P. Protzel. 2007. Comparing several implementations of two recently published feature detectors. In *Proc of the International Conference on Intelligent and Autonomous Systems*, 143–148. Tolouse, France. 3-5 September 2007.

Bay, H. 2006. Surf: Speeded up Robust Features. Available at: http://www.vision.ee.ethz.ch/~surf/download_ac.html. Accessed 05 July 2011.

Bay, H., A. Ess, T. Tuytelaars, and L. V. Gool. 2008. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3): 346–359.

Bay, H., T. Tuytelaars, and L. V. Gool. 2006. Surf: Speeded up robust features. In *the ninth European Conference on Computer Vision*, 404–417. Springer-Verlag, Berlin, Heidelberg. 07-13 May 2006.

Birchfield, S., and C. Tomasi. 1998. Depth discontinuities by pixel-to-pixel stereo. In *Proceedings of the 6th International Conference on Computer Vision*, 1073–1080. Washington, DC, USA. 04-07 January 1998.

Bouguet, J. Y. 2008. Camera Calibration ToolBox for Matlab. Available at: http://www.vision.caltech.edu/bouguetj/calib_doc/. Accessed 27 October 2011.

Fusiello, A., E. Trucco, and A. Verri. 2000. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*,

12(1): 16–22.

Geiger, A., M. Roser, and R. Urtasun. 2010. Efficient large-scale stereo matching. In *Proceedings of the 10th Asian conference on Computer vision*, 25–38. Queenstown, New Zealand, 8-12 November 2010.

Harris, C., and M. Stephens. 1988. A combined corner and edge detector. In *Proceedings of the Fourth Alvey Vision Conference*, 147–151. Manchester, 31 August-2 September 1988.

Hartley, R., and A. Zisserman. 2003. *Multiple View Geometry in Computer Vision*. United Kingdom: Cambridge University Press.

Hartley, R. I. 1999. Theory and practice of projective rectification. *International Journal of Computer Vision*, 35(2): 115–127.

Irena, D., and S. Aviad. Implementing the "GrabCut" Segmentation Technique. Available at: http://www1.idc.ac.il/toky/CompPhoto-09/Projects/Stud_projects/IrenaAviad/Web/. Accessed 02 March 2013.

Kohonen, T., and J. Hynninen, J. Kangas, and J. Laaksonen. 1996. Som pak: The self-organizing map program package. Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science.

Lhuillier, M., and L. Quan. 2005. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(3): 418.

Lindeberg, T. 1993. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, 11(3): 283–318.

Lindeberg, T. 1998. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2): 79–116.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2): 91–110.

Lowe, D.G. 2005. Demo Software: SIFT Keypoint Detector. Available at: http://www.cs.ubc.ca/~lowe/keypoints/. Accessed 03 June 2011.

Mikolajczyk, K., and Schmid, C. 2002. An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer*, 128–142. London, UK. 28-31 May 2002.

Mikolajczyk, K., and Schmid, C. 2004. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1): 63–86.

Mikolajczyk, K., T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. 2005. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2): 43–72.

Morel, J. M., and G.Yu. 2009. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2): 438–469.

Robotics Research Group, 2007. Affine Covariant Features. Available at: http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html. Accessed 06 August 2012.

Rother, C., V. Kolmogorov, and A. Blake. 2004. "GrabCut": interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, 309–314. Los Angeles, California, 08-12 August.

Scharstein, D., and R. Szeliski. 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3): 7–42.

Schumann, A. W., and Q. U. Zaman. 2005. Software development for real-time ultrasonic mapping of tree canopy size. *Computers and Electronics in Agriculture*, 47(1): 25–40.

Snavely, N., S. M. Seitz, and R. Szeliski. 2006. Photo tourism: exploring photo collections in 3D. In *ACM transactions on graphics (TOG)*, 835-846. Boston, Massachusetts, 30 July – 03 August.

Song, Y., C. A.Glasbey, and G. Polder. 2011. Non-destructive automatic leaf area measurements by combining stereo and time-of-flight images. *IET Computer Vision*, 28(5): 391–403.

Song, Y., R. Wilson, R. Edmondson, and N. Parsons.2007. Surface modelling of plants from stereo images. In *Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling. IEEE Computer Society*, 312–319. Montreal, QC, Canada. 21-23 Aug 2007.

Tumbo, S. D., M. Salyani, J. D. Whitney, T. A. Wheaton, and W. M. Miller. 2002. Investigation of laser and ultrasonic ranging sensors for measurements of citrus canopy volume. *Applied Engineering in Agriculture*, 18(3): 367–372.

Wang, Q., and Q. Zhang. 2013. Three-dimensional reconstruction of a dormant tree using rgb-d cameras. ASABE Paper No.131593521. St. Joseph, Michigan: American Society of Agricultural and Biological Engineers.

Wei, J., and M. Salyani. 2004. Development of a laser scanner for measuring tree canopy characteristics: phase 1. prototype development. *Transactions of the ASAE*, 47(6): 2101–2107.

Xie, Z., H. Wang, R. Liu, W. Xiang, and M. Jiang. 2010. 3D terrain reconstruction for patrol robot using point grey research stereo vision cameras, In *Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on*, 47–51. Sanya, China. 23-24 Oct 2010.

Xu, Z. 2004. Quasi-Dense Matching. Available at: http://www.openpr.org.cn/index.php/Code-of-Individual-Algorithms/6-Quasi-Dense-Matching/View-details.html. Accessed 10 June 2012.

Yu, G., and J. M. Morel. 2011. ASIFT: An Algorithm for Fully Affine Invariant Comparison. Available at: http://www.ipol.im/pub/art/2011/my-asift/. Accessed on 20 October 2012.

Zhang, Z. 2000. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11): 1330–1334.