

# Plant Leaf Surface Mesh Fitting Using RGB-D Sensors

Anonymous 3DV submission

Paper ID \*\*\*\*

## Abstract

Plant phenotyping is an emerging science that assess plant growth and development from the interactions between genetic variations and environments. Automatic plant phenotyping relies on sensors to measure plant leaf shape and pose in limited growth space. We propose a new method to significantly improve the accuracy of plant phenotyping by detecting and modeling the leaf 3D surface with inexpensive close-range time-of-flight RGB-D sensors. While these sensors provide dense depth, the noise in the depth is large compared to the surface features, making high-fidelity surface estimation challenging. Our method seeks to maximize the resolution and accuracy of the estimated surface by combining features from multiple modalities of the sensor including dense depth, near infra-red reflectance, and color. The result is an automated surface mesh that captures leaf boundary and shape information and filters out much of the noise. Experimental results on known shapes and real plant data show that our method is clearly better than the existing solutions.

## 1. Introduction

In order to meet the increasing needs for food and materials, researchers aim to produce sustainable plants with high yield and high resource-use efficiency by testing approaches ranging from molecular biology to field management [26]. In this process high-throughput plant phenotyping is the major bottleneck [6]. New automated methods to non-invasively measure plant phenotypes including growth, plant structure, leaf layout and orientations, photosynthesis and productivity, are highly expected [12, 18, 26].

An important step in estimating all of the plant phenotypic properties is to obtain 3D shape and pose for all the plant leaves [18], which has to be measured in a limited growth space with minimum disturbance, making it impractical to use scanning lasers for shape modeling. As an alternative, we propose to mount close-range time-of-flight RGB-D sensors in the growth chambers, acquire color, IR-reflectance and range data, and estimate leaf shape through

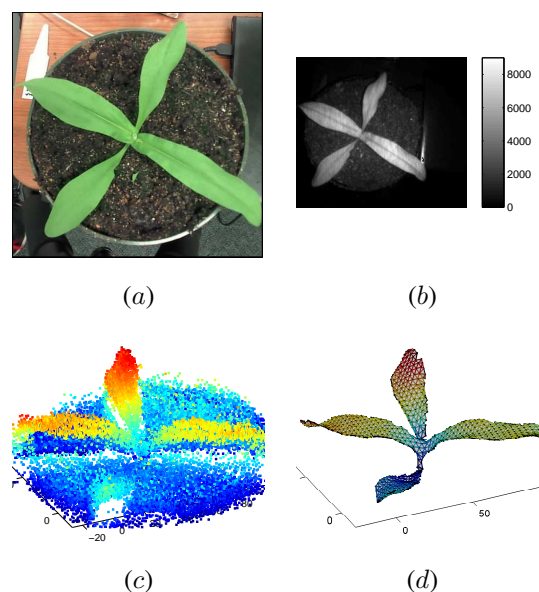


Figure 1: Illustration of sensor data. (a) Portion of color image. (b) IR reflectance image with reflectance values. (c) Portion of a single depth image surrounding plant averaged over 60 frames to reduce noise, and projected into 3D showing significant remaining depth noise. (d) The mesh resulting from the algorithm in this paper using data from the color, depth and IR reflectance images. Units of 3D plots are mm.

building leaf 3D mesh models (see Fig. 1). Mesh fitting to 3D point clouds and depth maps enables visualization and leaf shape analysis by providing a surface topology and geometry to a point cloud [22, 28]. A mesh can efficiently compress dense scanned data and mesh fitting methods have been developed that adhere to the underlying shape and preserve geometric features such as creases [8, 9]. However, while time-of-flight RGB-D sensors are small, inexpensive and provide dense 3D surface sampling of objects, they pose a challenge to surface modeling of small objects due to large depth noise. Specifically, in our leaf 3D mesh modeling ap-

plication, the pixel-depth noise has a standard deviation on the order of the surface feature sizes (see Fig. 3). In this paper, we present a new solution to automatically build precise 3D mesh models of plant leaves using time-of-flight RGB-D sensors. We achieve this through formulating an optimization that minimizes depth noise and adds a curvature-based regularization.

There have been differing goals in previous work on automated 3D leaf modeling. One goal has been to generate realistic foliage for graphics [5]. Here known leaf shapes are fit to 3D structure from motion data producing realistic-looking foliage, but not necessarily a close approximation to the actual plant leaf shapes and poses. In [21] an interactive system enables semi-automatic estimation of complex arrangements of leaves. The leaf shapes are all very similar and are fit by warping a template. In [3, 2] plant leaves are automatically segmented and fit using color and time-of-flight data enabling robotic arm inspection. This work is most similar to ours and they use a similar sensor, but the goals are quite different. They automatically segment and estimate poses of densely spaced leaves using simple geometric models (planes and parabolas) to fit the 3D data resulting in leaf poses and rough shape approximations. Our goal, on the other hand, is not segmenting overlapping leaves, but rather to estimate complex 3D leaf surfaces as precisely as possible from the noisy data. For this our flexible mesh model fitting approach gives much more accurate results.

We propose a new mesh generation algorithm that has the following advantages:

- It resolves complex 3D leaf shapes despite large noise in the range images by leveraging multiple sensor modalities.
- Boundary edge features in the high resolution color image are used to help select and constrain vertices and mesh edges.
- The errors in depth are modeled along camera rays and minimized during fitting to facets.
- A regularization term modeling curvature is incorporated and a linear approximation enables a fast, least-squares shape fit.

## 2. Related Work

This work detects and segments plant leaves in color images before fitting a 3D mesh. These detection and segmentation tasks can be challenging depending on the clutter, lighting, shadows, overlapping leaves, leaf textures etc., and there have been numerous research efforts in this area to address these challenges.

One class of approaches to leaf segmentation is through combining many low-level features. Contour detection is

well studied although only more recently have they been used for leaf shape and boundary segmentation. Simple cues like color, texture and local brightness and its combination are extensively used in contour and image boundary detections [16, 25]. To enhance segmentation and boundary regions, active contours and several improved variants of it [17] are also found to be quite popular in computer vision community. LeafSnap [11], a popular mobile application tool that identifies leaf species from photographs of leaves, rely on the more simplistic method of segmenting leaf by estimating the foreground and background color distribution by using Expectation Maximization in the saturation value space of the HSV colorspace. Research has also gained particular attention on segmentation of overlapping leaves for leaf detection. Pape et al. [20] tackles the problem of segmenting overlapping leaves partly by thresholding on built 3D histograms cubes of Lab color space both for training and testing data, and the quality of segmentation depends on the homogeneity of training data.

Another class of segmentation methods uses prior models. Manh et al. [15] use deformable templates in weed leaf segmentation which consists of fitting a parametric models to leaf outlines in image, by minimizing energy term related to internal constraints of the model and salient features of the image, such as color of plant. In related work, Toshev et al. [23] use both geometric properties of object boundary edges and coherent saliency cues distinct from the background. The method overcomes clutter in realistic scenes by handling segmentation using object specific knowledge like similarity in shape (top-down approach) and region growing principles (bottom up approach) from cues like color texture and normalized perimeter of objects.

A third approach includes Wei Ma et. al [14] who proposes modeling leaves using voxels. It aims to deal with the highly cluttered background by extracting the more visible apex features (position, middle axis and neighborhood) of the leaves instead of intelligent segmentation, from the volumetric data recovered from the images. Volumetric data provide pose and position of 3D leaves, and the 3D leaf shapes can be extracted from the optimized voxels.

For this paper we restrict ourselves to simplified conditions including non-overlapping leaves, low background clutter, and single plant types. This is to focus on the task of accurate leaf boundary and surface estimation. Hence our approach is not to constrain leaf boundaries with prior parametric shape models, but instead rely primarily on color cues for segmentation.

## 3. Sensor Overview

We explore using a new class of RGB-D sensor such as the Creative Senz3D [19] to build 3D mesh models of plant leaves. The sensor contains a high resolution color camera ( $1280 \times 720$  pixels) adjacent and parallel to a lower reso-

lution depth camera ( $320 \times 240$  pixels). A flash IR emitter adjacent to these cameras illuminates the scene and depth sensor measures the time-of-travel for the reflected light as well as its reflectance over its pixel grid. These modalities are illustrated in Figure 1. Our approach is to leverage the strengths of each modality to compensate for weaknesses in the other.

In order to gain the most from the sensors, we model effects due to sensor alignment and sensor noise. Pose alignment between the color and depth cameras as well as intrinsic parameters are calibrated using Zhang’s method [29]. This is possible for the depth camera since it also returns an IR reflectance at each pixel, and the checker board can be detected in this. In the rest of this section the occlusion effects of displaced sensors are modeled and the depth noise is modeled.

### 3.1. Occlusion Modeling

The color camera is offset to the side of the depth camera, and so at depth discontinuities, such as leaf boundaries, some pixels visible by the depth camera may be occluded in the color camera. During mesh fitting these occluded pixels may project into the leaf mesh and result in large depth distortions. In [3] an interpolated z buffer is used to detect these occluded pixels. Here we present a visibility analysis method that is closer to the stereo occlusion modeling [4]. We do not require rectification as there is no pixel matching, but we use the known horizontal offset of the color camera from the depth camera.

A row of pixels in the depth camera are projected out into world coordinates, as illustrated in Fig. 2. Then using just the planar position of the color camera (and not its orientation), the angle to the color sensor,  $\theta(i)$ , can be calculated for each depth pixel  $i$ . Starting from the left-most pixel we incrementally step through all the pixels and set  $\theta_{\max}(i) = \max(\theta(i), \theta_{\max}(i-1))$ . This keeps track of the largest angle seen so far in the row traversal. Occluded pixels are then simply:

$$\theta(i) < \theta_{\max}(i). \quad (1)$$

This assumes occluding objects are wide enough not to be seen around. Results of this are shown in Fig. 4(b). The only filtering done on this is to remove single occluded pixels with no neighbors as these occlusion labels are likely to be due to noise.

### 3.2. Noise Characterization

Depth measurements are performed along pixel rays, and hence depth noise is modeled as a one dimensional random variable,  $\varepsilon$ , along the ray for each pixel along its ray direction. We observe significant noise correlation between sequential measurements of a given pixel, and so do not assume sequential measurements are independent. On the

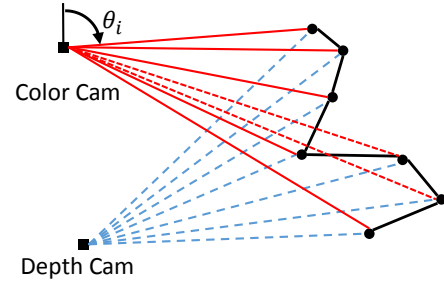


Figure 2: Occlusion modeling

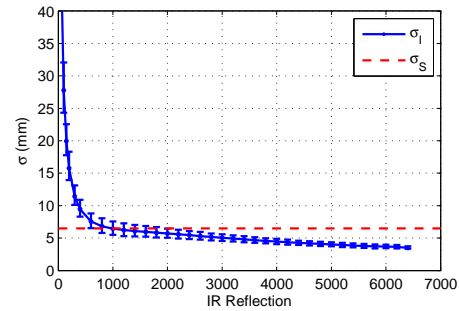


Figure 3: Image-varying noise,  $\sigma_I$ , is predicted well by the IR reflectance in raw units returned by the camera, see Figure 1(b). The scene-varying noise,  $\sigma_S$ , is plotted for comparison.

other hand averaging multiple measurements reduces the noise. To capture partial correlation we propose the following simple noise model.

The depth noise,  $\varepsilon$ , is modeled as the sum of an image-varying term,  $\varepsilon_I$ , and a scene-varying term,  $\varepsilon_S$ :

$$\varepsilon = \varepsilon_I + \varepsilon_S. \quad (2)$$

The term  $\varepsilon_I$  models the random change in depth for camera pixels of subsequent images of a static scene from a static camera. To quantify this term we measured the standard deviation  $\sigma_I$  in depth of each pixel for a batch of 300 images of a fixed scene containing a flat matte surface. We repeated this at different poses and depths, and with different surface albedos. While target depth, inclination, albedo, and pixel position are all correlated with  $\sigma_I$ , we found that the best predictor for  $\sigma_I$  was the IR reflectance intensity, as shown in Figure 3. For typical scenes the single measurement noise in depth is roughly 5mm, although for low reflectivity objects or objects at long range this noise can increase significantly. Fortunately plant leaves are good IR reflectors.

Averaging depth measurements of a fixed scene will reduce the noise from  $\varepsilon_I$ , but will not reduce the noise from

$\varepsilon_S$ . This latter scene-varying term is constant for a static scene, but changes when the scene changes. To characterize this noise we first eliminated (approximately) the image-varying noise contribution by averaging over a large number of images (300). Then assuming  $\varepsilon_S$  is independent and identically distributed between pixels, we measured the variance of the pixel depth errors between a known flat surface and the estimated surface. In our experiments we obtained  $\sigma_S = 6.5mm$ , and found that it was insensitive to changes in depth.

The total pixel noise can be estimated assuming independence of  $\varepsilon_I$  and  $\varepsilon_S$ , and is given by:

$$\sigma^2 = \frac{\sigma_I^2}{N} + \sigma_S^2, \quad (3)$$

where  $N$  is the number of images averaged over. When averaging 5 or more depth images the scene-varying contribution,  $\sigma_S^2$ , will dominate. There are additional sources of noise not modeled by this. These include object specularities, and mixed-depth pixels on object edges. These tend to produce very large image-varying noise,  $\sigma_I$ , and we can filter these points by discarding depth pixels with  $\sigma_I > 20mm$ .

#### 4. Color Image-Based Mesh Creation

For simplicity, and since the focus of this work is on accurate individual leaf surface modeling, we assume leaves are not overlapping. This allows us to rely primarily on the color image for leaf segmentation. Initial segmentation is done using the K-means clustering on the a and b channels of the Lab color space. We found that the a and b channels contain good discriminatory information on the leaf pigments [20]. Using three clusters on these two channels in the k-means clustering was sufficient to obtain a rough mask of the leaf regions. To refine the segmentation accuracy we then use simple linear iterative clustering (SLIC) superpixels [1] to split image into multiple homogenous regions. In most cases the boundaries of these superpixels adhere closely to the leaf boundaries. To detect whether each superpixel belongs to the leaf pigments or not, the initial batch of superpixels are selected by computing the centroid of each of them and checking whether it falls within the initial mask developed by K-means cluster. Any non leaf superpixels that are chosen in the first selection are then filtered out by thresholding on the ratio of 'a' and 'b' channels of the Lab color space. The selected superpixels are then merged to create the segmented leaf segment.

We build line segment based on boundary pixels of the segmented plant/leaf boundaries. Since plant leaves are clumped together in a single structure, it is possible to get closed boundary on the entire segmented image of the plant. We then perform a polygonal approximation motivated by

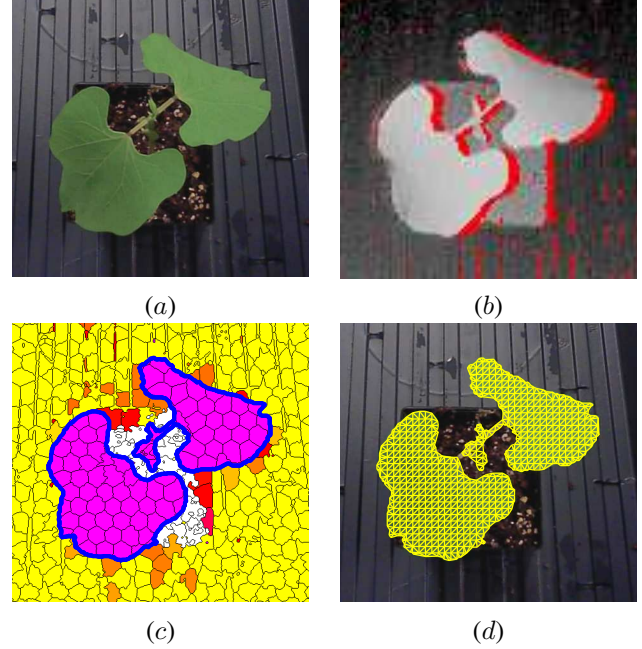


Figure 4: Initial algorithm steps. (a) Color image portion on plant. (b) Depth image, where red pixels are those that are masked out using the method in Fig. 2 as they are not visible in the color image. (c) SLIC superpixels on LAB color space. Each of the superpixels are colored with the mean L, a and b values of that particular superpixel. The leaf boundaries obtained by integrating the superpixels having the leaf pigments are shown in green dots. (d) A mesh is defined in the color image with evenly spaced vertices within and along the leaf boundaries.

the merging technique [13] process on the boundary by approximating straight lines with a maximum deviation of one pixels from pixels in the original boundary [10]. The two end points of the approximated lines are saved as vertices that join an edge of the polygon.

Having obtained the polygonal approximation of the plant leaf boundaries, we then sample points with a uniform spacing of  $\ell$  pixels on each side of the polygon. Uniform grid of points with a spacing of  $r$  are also created in the entire image, and the grid points which fall within or on the mask built by integrating the leaf superpixels are only selected. Points falling within  $\ell \leq r$  are removed from the set of selected grid points. Both the selected grid points and the sampled points on the boundary are then used to create a Delaunay triangulation on the plant mask. Facets that lie outside the polygonal shape are removed by determining whether their midpoints of the line fall within the polygonal shape or not. The final meshes on a typical color plant image looks like that in Fig. 4 (d).



## 5. Mesh Fitting

In mesh fitting we seek the most likely surface that generated the measured depth points. This is achieved by minimizing a noise model and by incorporating prior surface assumptions through regularization. Methods that fit mesh models to 3D points often minimize the perpendicular distance of points to facets [22]. This makes sense when point-cloud noise is equal in all directions or else the point noise is small compared to the facets. For our data the measurement noise is large and is not equal in all directions, but rather is along the depth camera rays. Our method to minimize this noise is described below.

We initialize a planar mesh in the color image as described in section 4, and then project it into 3D. This is more limiting than full 3D meshes as it models only the surface portions visible from the sensor, but it also provides a number of advantages. Compared to methods that fit prior surface models to depth maps [24, 8, 22], need to search of the space of poses, scales and distortions of the model with the chance of finding local minima. Compared to voxel-based models with implicit surfaces [7], our method can better incorporate pixels uncertainties and surface priors, as well as having fewer discretization artifacts. In addition our method can naturally incorporate detailed boundary features from the high-resolution color camera.

### 5.1. Notation

A vertex,  $\mathbf{v}_j$ , is a vector in 3D. In a given camera coordinate system, it projects onto a pixel on the unit focal-length image-plane  $\tilde{\mathbf{v}}_j = (u, v, 1)^\top$ , where the “~” indicates a homogeneous vector, and  $u$  and  $v$  are the coordinates in this plane. Now  $\tilde{\mathbf{v}}_j$  defines a ray from the camera origin, and the original vertex is obtained by scaling the image-plane vertex by its depth,  $\lambda_j$ , along the ray, namely:  $\mathbf{v}_j = \lambda_j \tilde{\mathbf{v}}_j$ .

### 5.2. Facet Model

Mesh fitting for an individual facet is illustrated in Figure 5. The initial 2D vertices and edge connections are determined in the color image. If these vertices lie on a feature of the target leaf, such as its edge, we know that those 3D features lie somewhere along the rays emanating from camera origin through those vertices. Hence a triangular facet approximation to the object surface will have vertices on these three rays.

The next step is to associate depth measurements with the facet. Pixels in the depth camera are projected along their rays out into 3D, and then they are projected into the color image. We associate the depth pixel,  $\mathbf{p}_i$ , with the facet,  $\mathcal{F}_i$ , into which it projects in the color image, as illustrated in Figure 5.

To estimate the facet parameters from depth measurements we will express the depth points,  $\mathbf{p}_i$ , after they have

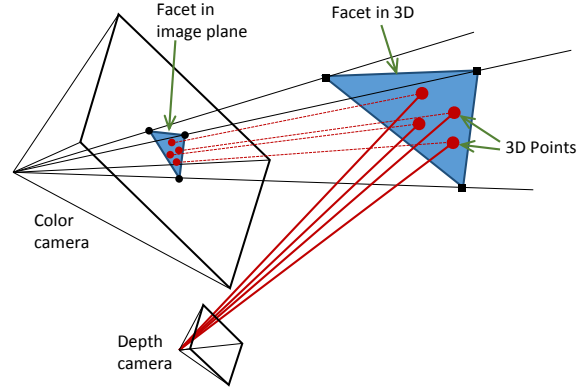


Figure 5: The parallel and adjacent color and depth cameras are shown as pyramids denoting their fields of view, and their size difference illustrates their relative resolutions. Three vertices in a color image define the rays on which the vertices of the corresponding 3D object facet must lie. This facet is fit using the 3D points projected out from the depth camera.

been projected out and transformed into color camera coordinates, as a linear function of the vertices of its facet:

$$\mathbf{p}_i = \sum_{j \in \mathcal{F}_i} \alpha_j \mathbf{v}_j. \quad (4)$$

Here  $\mathcal{F}$  is the set of three vertex indices belonging to the facet, and  $\alpha_j$  is the coefficient of vertex  $\mathbf{v}_j$  as illustrated in Figure 6, and  $\sum_{j \in \mathcal{F}} \alpha_j = 1$ . This linear sum is valid if we make a local orthographic approximation for the projection of a facet. It will be a good approximation as long as the facet size is small compared to its depth from the camera, which is true for most applications. Substituting in depth-scaled homogeneous vectors, and taking the third row, we obtain an equation for the point depth,  $\lambda_i$ , in the color image:

$$\lambda_i = \sum_{j \in \mathcal{F}_i} \alpha_j \lambda_j. \quad (5)$$

### 5.3. Least Squares Depth

Equation (5) gives the modeled depth of a point in terms of its facet vertices. The depth camera will provide measured depths for each pixel, indicated as  $\bar{\lambda}_i$ , along with a standard deviation estimate  $\sigma_i$  obtained from Eq. (3). Now this noise is along the depth ray, rather than the color camera ray, but since the depth and color cameras are close together compared to the distance to target, these rays are close to parallel and we assume  $\sigma_i$  is a good measure along the camera ray. With this approximation, the weighted least squares

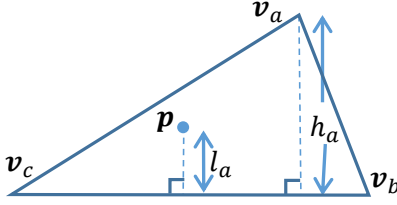


Figure 6: The coordinates of a point on a facet described by Eq. (4) are the weighted linear sum of the three vertex coordinates. The weight,  $\alpha_a$ , for vertex  $v_a$  is given by  $\alpha_a = \frac{l_a}{h_a}$ , the ratio of its perpendicular distance  $l_a$  to the opposite edge to the vertex perpendicular distance  $h_a$ . Analogous expressions describe  $\alpha_b$  and  $\alpha_c$ .

cost between measured and predicted depth is:

$$E_{depth}(\lambda_v) = \sum_{i \in \mathcal{D}} \|\bar{\lambda}_i - \sum_{j \in \mathcal{F}_i} \alpha_j \lambda_j\|_{\sigma_i^2}^2, \quad (6)$$

where  $\mathcal{D}$  is the set of depth pixels that project onto the target. The L2 norm is weighted with the inverse variance of each point. Minimizing this for  $\lambda_v$ , the set of all vertex depths, is a straightforward linear calculation. It estimates the mesh and it correctly minimizes the measurement error.

#### 5.4. Regularization

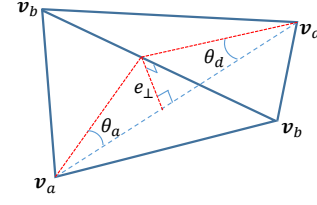
Prior models on surface properties can be incorporated into the mesh via regularization and in so doing reduce the impact of noise. In our application leaf surfaces are generally smoothly curved with few sharp edges. We propose a regularization that approximates. Since our mesh facets are roughly equal in size, we can use the angle,  $\theta$ , between adjacent facet normals,  $\hat{n}_i$  and  $\hat{n}_j$ :

$$E_{reg} = \sum_{adj(i,j)} \cos(\hat{n}_i \cdot \hat{n}_j)^{-1}. \quad (7)$$

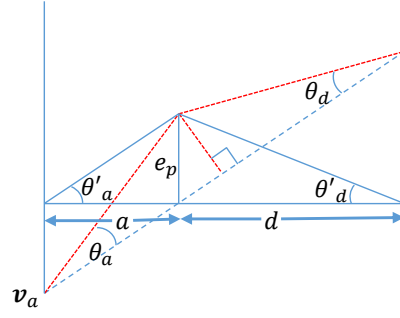
Here the sum is over all pairs of normals whose facets share an edge and hence are adjacent. Adding this nonlinear regularization term to the optimization can significantly improve surface estimation with strong noise.

It is also useful to have a linear regularization function. The reason is that initial depth fitting on the mesh is a fast linear least squares estimate, but if some vertices are insufficiently constrained the matrix inversion suffers from loss of full rank. This tends to happen when facets are small enough that a number of them have no depth pixels projecting into them. Adding a linear regularization term between pairs of facets will ensure that the coefficient matrix maintains full rank during least squares.

We created an alternative linear regularization that uses an approximation to the angle between adjacent facets. Figure 7(a) illustrates that the angle between two adjacent



(a)



(b)

Figure 7: (a) The angle between adjacent facet normals used in Eq. (7) can also be calculated as the sum of the angles subtended by  $e_{\perp}$ , namely:  $\theta = \theta_a + \theta_d$ . (b) An alternative is to use  $\theta'_a$  and  $\theta'_d$  as these can be calculated using the image projection of the facets, along with the depths of the vertices.

facets is given by  $\theta = \theta_a + \theta_d$ , the angles subtended by  $e_{\perp}$ , the perpendicular distance between lines through opposite vertices of the (non-planar) quadrilateral formed by the two facets. Instead we use the projection of the quadrilateral in the image, find the intersection of the lines between opposite vertices, and calculate the depth distance along this intersection ray  $e_p = \frac{a\lambda_a + d\lambda_d}{a+d} - \frac{b\lambda_b + c\lambda_c}{b+c}$ , where  $a$ ,  $b$ ,  $c$  and  $d$  are image distances along the edges joining the intersection point to the respective vertices. Finally we use the tangent of the angles in the regularization to obtain:

$$\begin{aligned} E_{linreg} &= \sum_{adj(i,j)} \|\tan(\theta'_{ai})\|_2 + \|\tan(\theta'_{di})\|_2 \\ &= \sum_{adj(i,j)} \|\frac{e_p}{a_i}\|_2 + \|\frac{e_p}{d_i}\|_2. \end{aligned} \quad (8)$$

Again the sum is over all pairs of adjacent facets.

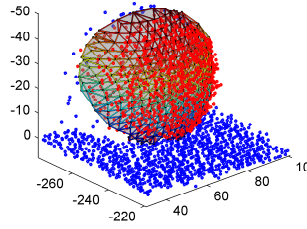
#### 5.5. Algorithm

### 6. Results

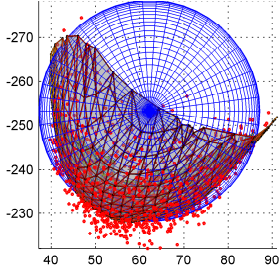
Mesh reconstructions on a number of plants are presented. In all 3D plots the units are millimeters. Figure 1 shows image and 3D data, along with the estimated mesh of a potted plant. Detailed shapes are recovered. The esti-



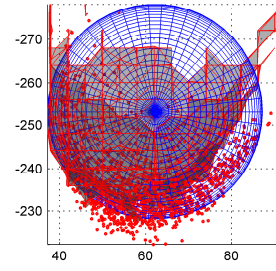
(a)



(b)



(c)



(d)

Figure 8: (a) Color image of a 50mm diameter sphere. (b) Estimated 3D mesh. (c) Known sphere shape with center fit to data points shows how well the mesh captures the real surface. (d) The isosurface is much more jagged and does not fit as well.

mated mesh is compared to a laser scan of the same plant in Fig. 9. This shows overall good accuracy of the shape, with the main error being an extension of the leaf tip area. It appears the lower leaf shifted in position between the laser scan and the camera image, and this may be a result of the turntable required by the laser.

Another way to quantitatively evaluate the surface is to estimate a known 3D object. In Fig. 8 a 50mm diameter sphere is imaged and the aligned to the points on the surface. The 3D point offsets from the sphere changes with the slope indicating that there are some inherent depth biases in the sensor. Nevertheless the mesh smoothes the data noise and approximates the sphere surface well. The standard deviation of the raw points fit to the sphere is 2.3mm and the mesh vertices is 1.3mm. This does not account for absolute depth error, only shape error, as the sphere center was aligned.

Surface estimates for the bean plant from Fig. 4 are shown in Fig. 10 and from a soybean plant in Fig. 11. The 3D points have been averaged over 30 frames to reduce noise, but still are noisy. Despite this mesh is able capture the leaf features well. A comparison is made to the Marching Cube algorithm from an isosurface [7] in Fig. 4(c). We used a modified implementation from [27] along with the

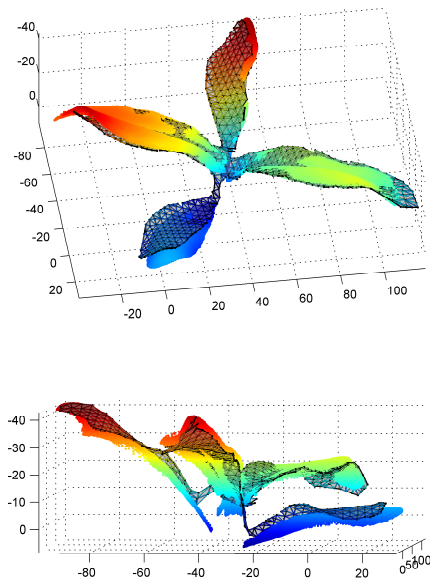


Figure 9: The 3D reconstruction from Fig. 1 is aligned to a laser scan of the plant image. The laser scan data are shown as smoothly-colored surfaces plotted in the same axes as the mesh. The two views show good overall agreement including feature curves. Errors are seen near the ends of the left and right leaves. The lower leaf has likely shifted between the laser scan and the depth data collection.

Matlab isosurface command. This shows the voxelization generating large artifacts, and using smaller voxels simply leads to surface holes. Fig. 10(d) compares the mesh surface with a parabolic fit [3]. The latter gives rough shape but not a detailed model as we are able to obtain with our mesh fitting.

## 7. Conclusion

## References

- [1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, 2012. 4
- [2] G. Alenya, B. Dellen, S. Foix, and C. Torras. Robotized plant probing: Leaf segmentation utilizing time-of-flight data. *Robotics Automation Magazine, IEEE*, 20(3):50–59, Sept 2013. 2, 8
- [3] G. Alenya, B. Dellen, and C. Torras. 3d modelling of leaves from color and tof data for robotized plant measuring. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3408–3414, May 2011. 2, 3, 7, 8

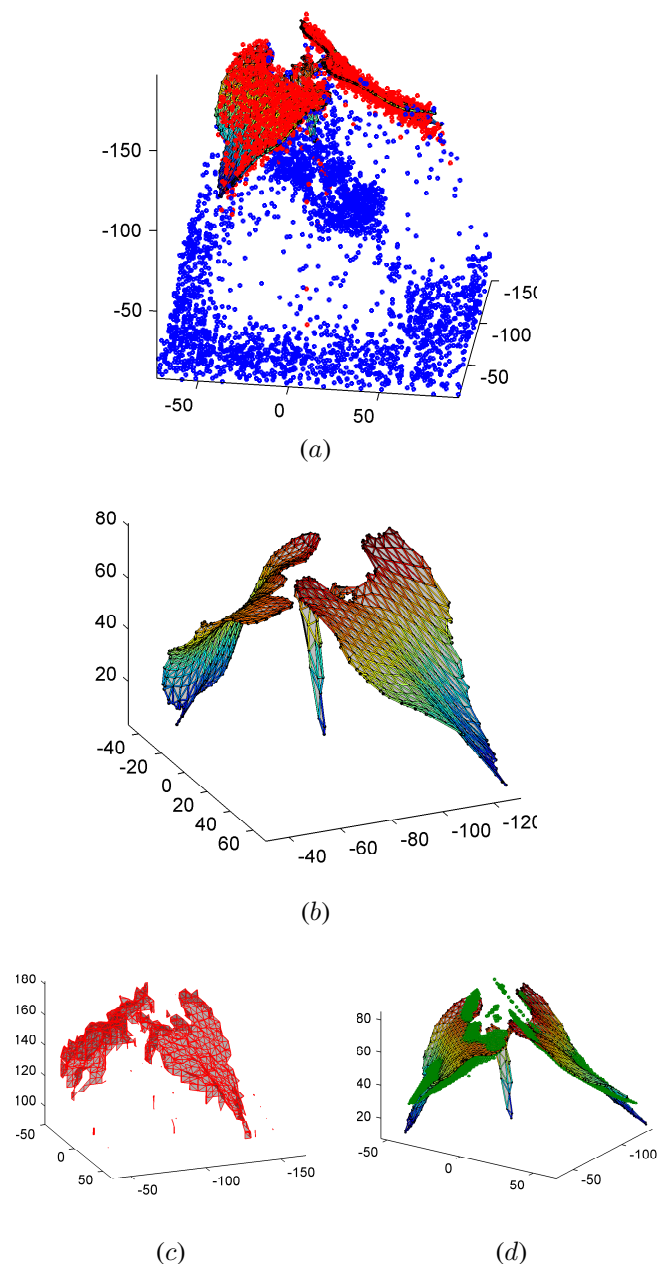


Figure 10: Result of fitting the mesh to the bean data from Fig. 4. (g) The 3D points are red for those on the mesh and blue for the remaining. (h) A more detailed view of the mesh revealing detailed curves of the leaves. (c) A comparison with an isosurface created using marching cubes [7]. (d) A comparison with the use of parabolic fits to leaf surfaces (dark green points) as used in [3, 2]. Clearly parabolic models cannot capture complex bean leaf structures or the leaves in Fig. 9.

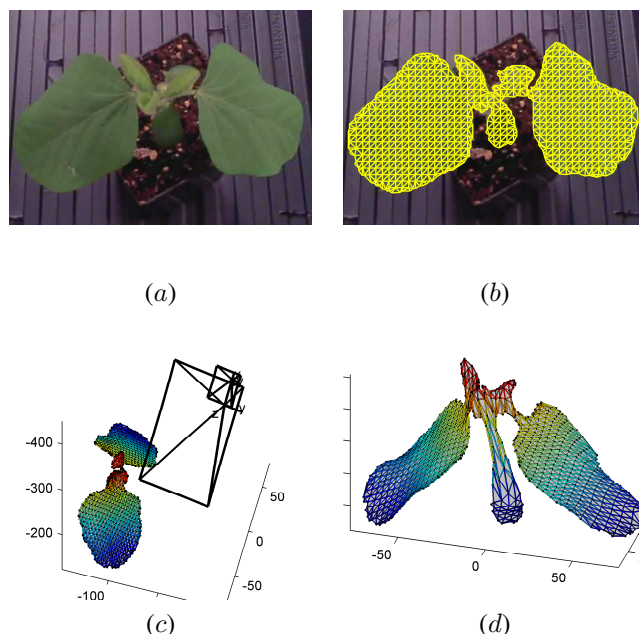


Figure 11: (a) Color image of a soybean plant. (b) Mesh on the color image. (c) 3D mesh along with camera poses. (d) Close-up of 3D mesh.

- [4] P. Belhumeur. A bayesian approach to binocular stereopsis. *International Journal of Computer Vision*, 19(3):237–260, 1996. 3
- [5] D. Bradley, D. Nowrouzezahrai, and P. Beardsley. Image-based reconstruction and synthesis of dense foliage. *ACM Trans. Graph.*, 32(4):74:1–74:10, July 2013. 2
- [6] J. N. Cobb, G. DeClerck, A. Greenberg, R. Clark, and S. McCouch. Next-generation phenotyping: requirements and strategies for enhancing our understanding of genotype–phenotype relationships and its relevance to crop improvement. *Theoretical and Applied Genetics*, 126(4):867–887, 2013. 1
- [7] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’96*, pages 303–312, New York, NY, USA, 1996. ACM. 5, 7, 8
- [8] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’94*, pages 295–302, New York, NY, USA, 1994. ACM. 1, 5
- [9] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’98*, pages 105–114, New York, NY, USA, 1998. ACM. 1



- [10] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia. Available from: <http://www.csse.uwa.edu.au/~pk/research/matlabfns/>. 4
- [11] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *Computer Vision-ECCV 2012*, pages 502–516. Springer, 2012. 2
- [12] D. Leister, C. Varotto, P. Pesaresi, A. Niwergall, and F. Salamini. Large-scale evaluation of plant growth in *Arabidopsis thaliana* by non-invasive image analysis. *Plant Physiology and Biochemistry*, 37(9):671–678, 1999. 1
- [13] J.-G. Leu and L. Chen. Polygonal approximation of 2-d shapes through boundary merging. *Pattern Recognition Letters*, 7(4):231 – 238, 1988. 4
- [14] W. Ma, H. Zha, J. Liu, X. Zhang, and B. Xiang. Image-based plant modeling by knowing leaves from their apexes. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, Dec 2008. 2
- [15] A.-G. Manh, G. Rabatel, L. Assemet, and M.-J. Aldon. Automation and emerging technologies: Weed leaf image segmentation by deformable templates. *Journal of Agricultural Engineering Research*, 80(2):139 – 146, 2001. 2
- [16] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(5):530–549, 2004. 2
- [17] A. K. Mishra, P. W. Fieguth, and D. A. Clausi. Decoupled active contour (dac) for boundary detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(2):310–324, 2011. 2
- [18] M. Müller-Linow, F. Pinto-Espinosa, H. Scharr, and U. Rascher. The leaf angle distribution of natural plant populations: assessing the canopy with a novel software tool. *Plant methods*, 11(1):11, 2015. 1
- [19] V. Nguyen, M. Chew, and S. Demidenko. Vietnamese sign language reader using intel creative senz3d. In *IEEE International Conference on Automation, Robotics and Applications (ICARA)*, pages 77–82, 2015. 2
- [20] J.-M. Pape and C. Klukas. 3-d histogram-based segmentation and leaf detection for rosette plants. In L. Agapito, M. M. Bronstein, and C. Rother, editors, *Computer Vision - ECCV 2014 Workshops*, volume 8928 of *Lecture Notes in Computer Science*, pages 61–74. Springer International Publishing, 2015. 2, 4
- [21] L. Quan, P. Tan, G. Zeng, L. Yuan, J. Wang, and S. B. Kang. Image-based plant modeling. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 599–604, New York, NY, USA, 2006. ACM. 2
- [22] J. Sienz, I. Szarvasy, E. Hinton, and M. Andrade. Computational modelling of 3d objects by using fitting techniques and subsequent mesh generation. *Computers & Structures*, 78(13):397 – 413, 2000. 1, 5
- [23] A. Toshev, B. Taskar, and K. Daniilidis. Shape-based object detection via boundary structure segmentation. *International journal of computer vision*, 99(2):123–146, 2012. 2
- [24] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, pages 311–318, New York, NY, USA, 1994. ACM. 5
- [25] N. Valliammal and S. Geethalakshmi. Leaf image segmentation based on the combination of wavelet transform and k means clustering. *International Journal of Advanced Research in Artificial Intelligence*, 1(3):37–43, 2012. 2
- [26] A. Walter, F. Liebisch, and A. Hund. Plant phenotyping: from bean weighing to image analysis. *Plant methods*, 11(1):14, 2015. 1
- [27] J. Xiao. Princeton vision toolkit, 2013. Available from: <http://vision.princeton.edu/code.html>. 7
- [28] I.-C. Yeh, C.-H. Lin, O. Sorkine, and T.-Y. Lee. Template-based 3d model fitting using dual-domain relaxation. *Visualization and Computer Graphics, IEEE Transactions on*, 17(8):1178–1190, Aug 2011. 1
- [29] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, Nov 2000. 3