Plant Leaf Meshes from Time-of-Flight RGB-D Sensors

Anonymous 3DV submission

Paper ID ****

Abstract

Research into improving plant growth and yield relies on sensors to measure plant phonotypes, such as photosynthesis, in arrays of growth chambers. The accuracy of these measured phenotypes can be improved if the 3D surface area and orientations of leafs are known. This paper addresses this need for automated 3D plant leaf estimation by presenting a method for using an inexpensive time-of-flight sensor tightly integrated with a color camera to estimate 3D leaf mesh models. 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. Examples are shown on known shapes and real plant data.

1. Introduction

This paper addresses the problem of automatically building 3D mesh models of plant leaves using inexpensive time-of-flight RGB-D sensors. Plant researchers, seeking to understand genetic underpinnings of plant growth [REF] and seeking to develop new varieties [REF], need automated ways to non-invasively measure plant phenotypes including growth, leaf distributions, orientations, photosynthesis and productivity [REF]. An important step in estimating all of these properties is obtaining 3D shape and pose for all the plant leafs. Plants cannot be moved or disturbed in limited-space growth chambers making it impractical to use scanning lasers for shape modeling. Instead our concept is to mount close-range time-of-flight RGB-D sensors in the chambers, acquire color, IR-reflectance and range data, and estimate leaf shape through building 3D mesh models of the leafs.

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. This pixel-depth noise is significantly larger than the surface features we hope to capture, and can be on the order of the leaf size. The goal of this paper is to cut through this noise to obtain high fidelity surface models of leaves and small objects.

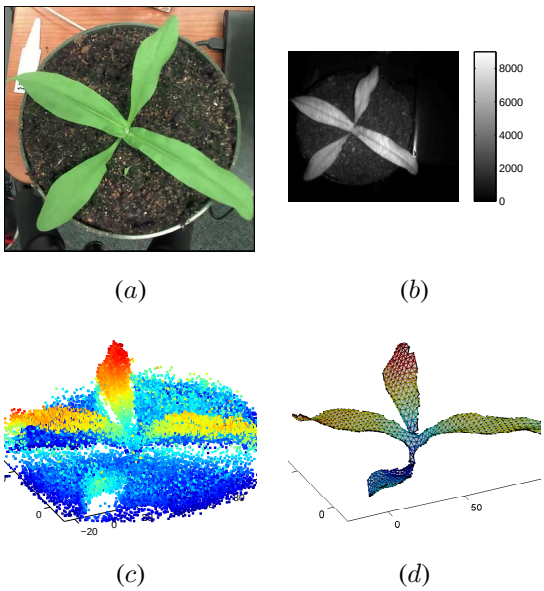

Figure 1: Illustration of sensor data. (a) Portion of color image. (b) IR reflectance image with reflectance values. (c) Portion of a single depth image surrounding plant averaged over 60 frames to reduce noise, and projected into 3D showing significant remaining depth noise. (d) The mesh resulting from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

Mesh fitting to 3D point clouds and depth maps has had a number of objectives. A mesh provides a surface topology and geometry to a point cloud [14, 16]. This enables visualization and shape analysis of the target object. 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 [7, 9]. The problem differs from these in that our depth data are far noisier and our goal is to recover the true surface rather than adhering as closely as possible to the 3D points.

Another goal for mesh fitting is to incrementally build full 3D models of objects. Zippered polygons [15] builds mesh models from range images and combines them discarding noisy points at the boundaries. The method developed by Curless and Levoy [5] populates a weighted voxel occupancy grid from the depth data and recovers the surface by triangulating an isosurface. By raycasting this surface, new camera poses can be aligned and their depths maps contribute to and improve the voxel model. Advantages of this method include that surface topology is automatically determined, additional data can be readily incorporated, holes can be filled when more data are collected and it incorporates directional uncertainty of range data into the models. Recent approaches for environment modeling from RGB-D cameras such as Kinect Fusion [8, 11] build on this voxel modeling and accumulation. For our application the sensor is fixed and there is no option for merging views from different perspectives. Artifacts caused by discretization into voxels are significant particularly with high input noise, and the method is limited in incorporating surface smoothness priors.

The method proposed here is a new mesh generation algorithm that resolves object shape despite large noise in the range images by leveraging multiple sensor modalities. Edge features of 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. The IR reflectance image is used as a Shape from Shading cue to influence facet surface normals. A linear least squares solution to Laplacian smoothing is integrated into the mesh fitting for regularization with surface curvature priors, and modification is made to preserve sharp edges and creases.

2. Related Work

Optimizing meshes for fit point cloud data has been approached through vertex additions and removals [7], although this work assumes the point data are precise and dense

3. Sensor Overview

We explore using a new class of RGB-D sensor such as the Creative Senz3D [12] to build 3D mesh models of plant leafs. The sensor contains a high resolution color camera (1280×720 pixels) adjacent and parallel to a lower resolution depth camera (320×240 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 of the strengths of each sensor to compensate for weaknesses in the other.

3.1. Sensor Issues

In our application the sensor remains fixed in the growth chamber and so there is not the opportunity to combine measurements from different viewpoints. Rather, we seek to maximize the information from the different modalities of the sensors. First leaf outlines can often be more precisely determined in the color camera for three reasons: the color camera has an order of magnitude more pixels than the depth camera, edge pixels in the depth camera are often very noisy due to double reflections, and color is often a useful cue in segmenting leaves. Thus as in [6] and unlike [2, 1] low-level segmentation is performed in the full color image. The color image also provides useful edge cues along leaf creases.

The depth plus reflectance camera can provide leaf pixel cues, albeit at a lower resolution, as well as leaf geometry. Combining these leaf pixel cues with color segments enables automatic selection of the leaf segments. Then the color boundaries and features can be used to constrain a mesh-based surface fitting of the leaves.

Since we are interested in close-range objects, the baseline separating the color from the depth camera must be accounted for when determining pixel correspondences. At depth discontinuities, this baseline can result in pixels from two different surfaces, both visible in the depth camera, being projected onto the same pixel region in the color camera.

Outline:

1. Depth + IR initial leaf detection
2. Augmented color segmentation
3. Color boundary and edges
4. Mesh fitting and filtering

While the sensor produces dense depth measurements over target leaf surfaces, the noise in depth measurements is significantly larger than the features we are seeking to recover as illustrated in Figure 1(c). A key goal in this paper is to overcome this noise to maximize the accuracy of leaf shape estimates. We start in this section by modeling and quantifying the measurement noise.

3.2. Noise Characterization

Since the depth camera returns an IR reflectance in addition to a depth value at each pixel, both it and the color camera are initially calibrated using Zhang's method [17] to obtain intrinsic and extrinsic parameters. Thus each pixel

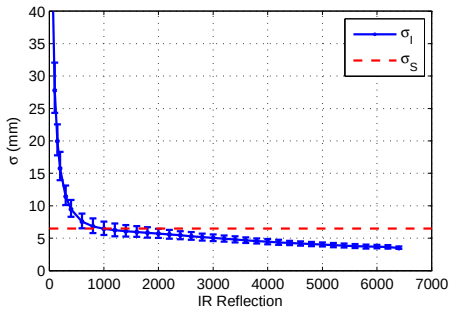

Figure 2: Image-varying noise, σ_I, is predicted well by the IR reflectance in raw units returned by the camera, see Figure 1(*b*). The scene-varying noise, σ_S, is plotted for comparison.

in each camera defines a ray from its camera origin. Depth noise is modeled as a one dimensional random variable, ε, along the ray for each pixel along its ray direction.

The depth noise, ε, is modeled as the sum of an image-varying term, ε_I, and a scene-varying term, ε_S:

$$\varepsilon = \varepsilon_I + \varepsilon_S. \tag{1}$$

The term ε_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 σ_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 σ_I, we found that the best predictor for σ_I was the IR reflectance intensity, as shown in Figure 8. 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 leafs are good IR reflectors.

Averaging depth measurements of a fixed scene will reduce the noise from ε_I, but will not reduce the noise from ε_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 ε_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 ε_I and ε_S, and is given by:

$$\sigma^2 = \frac{\sigma_I^2}{N} + \sigma_S^2, \tag{2}$$

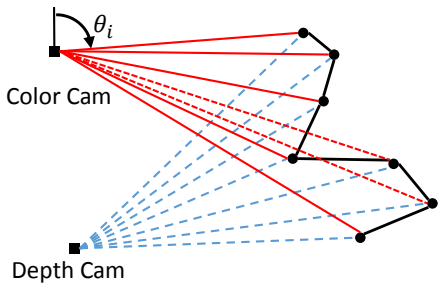

Figure 3: Occlusion modeling

where N is the number of images averaged over. When averaging 5 or more depth images the scene-varying contribution, σ_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, σ_I, and we can filter these points by discarding depth pixels with $\sigma_I > 20$mm.

3.3. Occlusion Modeling

4. Mesh Fitting

We pose mesh fitting to 3D point data as finding the most likely surface that would have generated those points. By incorporating prior surface assumptions, the fitting process estimates a continuous surface from discrete points that can eliminate much of the measurement noise. Methods that fit mesh models to 3D points often minimize the perpendicular distance of points to facets [REF]. 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. Hence the focus of this section is to develop a mesh fitting method that minimizes these pixel depth errors along the pixel rays.

In this paper we define a mesh in a 2D image space and 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 [ref], 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 [ref], 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 features from the high-resolution color camera, and reflectivity information from the IR reflectance image.

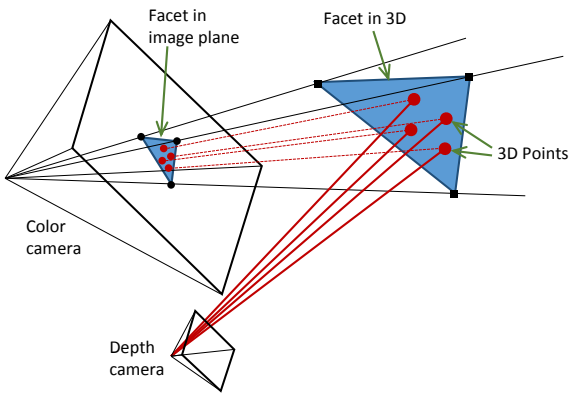

Figure 4: The parallel and adjacent color and depth cameras are shown as pyramids denoting their fields of view, and their size difference illustrates their relative resolutions. Three vertices in a color image define the rays on which the vertices of the corresponding 3D object facet must lie. This facet is fit using the the 3D points projected out from the depth camera.

4.1. Notation

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

4.2. Facet Model

Mesh fitting for an individual facet is illustrated in Figure 4. The 2D vertices and edge connections are determined in an image, in this case the color image although it could be the depth image, as described in section 5. If these vertices lie on a feature of the target leaf, such as its edge, we know that that those 3D features like 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, $\boldsymbol{p}_i$, with the facet, $\mathcal{F}_i$, into which it projects in the color image, as illustrated in Figure 4.

To estimate the facet parameters from depth measurements we will express the depth points, $\boldsymbol{p}_i$, after they have been projected out and transformed into color camera coordinates, as a linear function of the vertices of its facet:

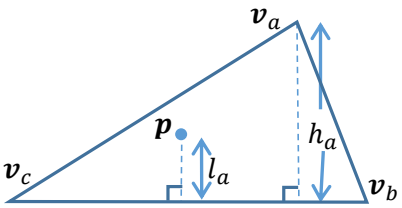

Figure 5: The coordinates of a point on a facet described by Eq. (3) are the weighted linear sum of the three vertex coordinates. The weight, α_a, for vertex $\boldsymbol{v}_a$ is given by $\alpha_a = \frac{l_a}{h_a}$, the ratio of its perpendicular distance l_a to the opposite edge to the vertex perpendicular distance h_a. Analogous expressions describe α_b and α_c.

$$\boldsymbol{p}_i = \sum_{j \in \mathcal{F}_i} \alpha_j \boldsymbol{v}_j. \tag{3}$$

Here $\mathcal{F}$ is the set of three vertex indices belonging to the facet, and α_j is the coefficient of vertex $\boldsymbol{v}_j$ as illustrated in Figure 5, 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 is 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, λ_i, in the color image:

$$\lambda_i = \sum_{j \in \mathcal{F}_i} \alpha_j \lambda_j. \tag{4}$$

4.3. Least Squares Depth

Equation (4) 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 σ_i obtained from Eq. (2). 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 σ_i is a good measure along the camera ray. With this approximation, the weighted least squares cost for between measured and predicted depth is:

$$E_{depth}(\Lambda_{\boldsymbol{v}}) = \sum_{i \in \mathcal{D}} \| \bar{\lambda}_i - \sum_{j \in \mathcal{F}_i} \alpha_j \lambda_j \|^2_{\sigma_i^2}, \tag{5}$$

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

4.4. 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, and in some cases have creases such as along the spine. Since our mesh facets are roughly equal in size, we can use the angle between adjacent facet normals, $\hat{\boldsymbol{n}}_i$ and $\hat{\boldsymbol{n}}_j$:

$$E_{reg} = \sum_{adj(i,j)} \cos(\hat{\boldsymbol{n}}_i \cdot \hat{\boldsymbol{n}}_j)^{-}1. \quad (6)$$

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 opimization 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 6(a) illustrates that the angle between two adjacent 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:

$$\begin{aligned} E_{linreg} &= \textstyle\sum_{adj(i,j)} \|\tan(\theta'_{ai})\|_2 + \|\tan(\theta'_{di})\|_2 \\ &= \textstyle\sum_{adj(i,j)} \|\frac{e_p}{a_i}\|_2 + \|\frac{e_p}{d_i}\|_2. \end{aligned} \quad (7)$$

Again the sum is over all pairs of adjacent facets.

4.5. Algorithm

5. Color-Based Mesh Cues

Since we do not deal with overlapping and highly cluttered background, the segmentation can be done easily by using the color information of the rgb image. The initial segmentation on the leaf is done by using the K-means cluster on the a and b channels of the Lab color space. It is found that the a and b channels contain much better information about the green pigments of the leaf. Using only three clusters on these two channels in the k-means clustering, it is able to give a rough mask of where the leaf regions are. But for an accurate mesh fitting, it is required to find the leaf shape by following the edges of the shape boundaries as closely as possible. It is well known in the literature that superpixels [**?**] can split image into multiple homogenous regions and adhere shape boundaries very closely. We use the simple linear iterative clustering (SLIC) superpixels to form superpixels on the entire image. It is found that though SLIC superpixels form an oversegmentation of the image, they are found to adhere very 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

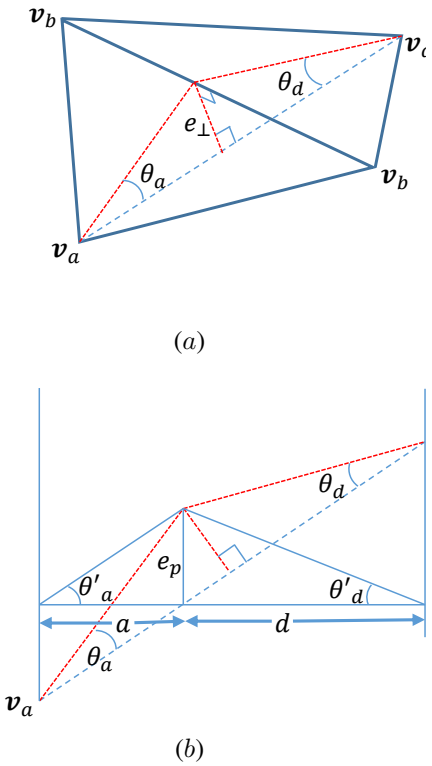

Figure 6: (a) The angle between adjacent facet normals used in Eq. (6) can also be calculated as the sum of the angles subtended by $e_\perp$, namely: $\theta = \theta_a + \theta_d$. ($b$) An alternative is to use θ'_a and θ'_d as these can be calculated using the image projection of the facets, along with the depths of the vertices.

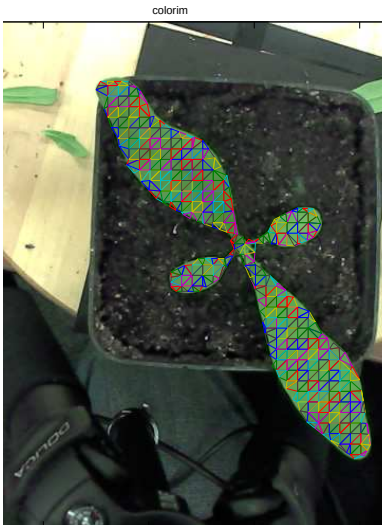

Figure 7: Mesh Distribution on the RGB Plant Imagery

create the segmented plant/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 the merging technique [10] process on the boundary by approximating straight lines with a maximum deviation of one pixels from pixels in the original boundary [**?**]. 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 ℓ 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 structure 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.The slivers of triangular facets that lie outside the polygonal shape are removed by seeking whether the midpoints of the line fall within the polygonal shape or not.The final meshes on a typical rgb plant image looks like in Figure 7:

6. Results

7. Conclusion

References

[1] 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

[2] 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] M. Chelle. Could plant leaves be treated as lambertian surfaces in dense crop canopies to estimate light absorption? *Ecological Modelling*, 198(1):219 – 228, 2006.

[4] C.-Y. Chen and K.-Y. Cheng. A sharpness dependent filter for mesh smoothing. *Computer Aided Geometric Design*, 22(5):376 – 391, 2005. Geometry Processing.

[5] 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 In-*

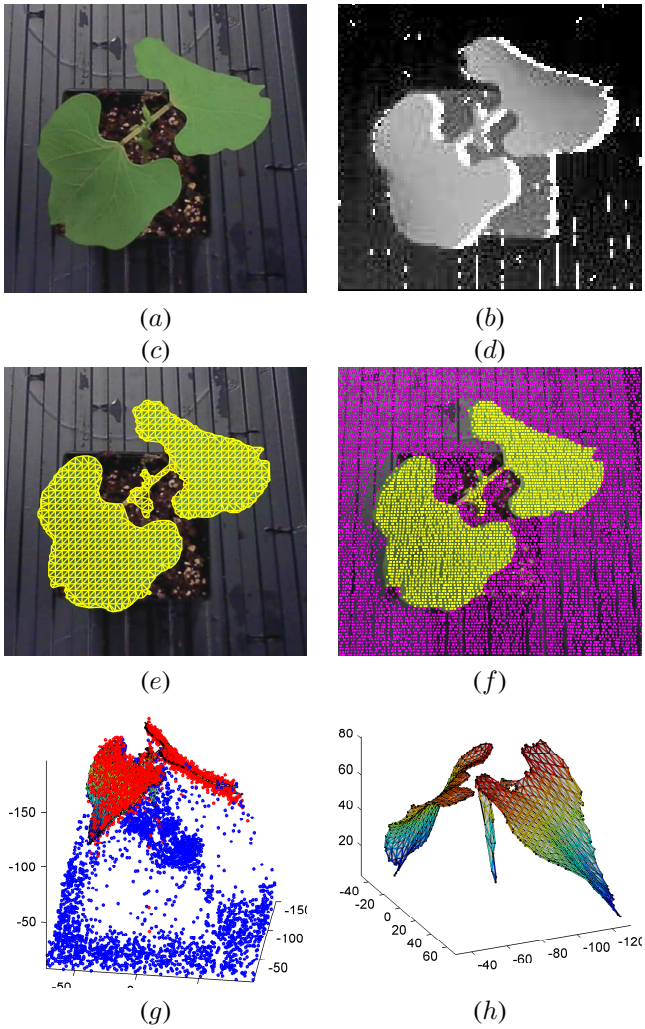

Figure 8: (a) Color image. (b) Depth image. White pixels are those that are automatically masked out due to being not visible in the color image.

teractive Techniques, SIGGRAPH '96, pages 303–312, New York, NY, USA, 1996. ACM. 2

[6] B. Dellen, G. Alenya, S. Foix, and C. Torras. Segmenting color images into surface patches by exploiting sparse depth data. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 591–598, Jan 2011. 2

[7] 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. 2

[8] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proc. of ACM UIST*, pages 559–568, 2011. 2

[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. 2

[10] J.-G. Leu and L. Chen. Polygonal approximation of 2-d shapes through boundary merging. *Pattern Recognition Letters*, 7(4):231 – 238, 1988. 5

[11] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Proc. of IEEE ISMAR*, pages 127–136, 2011. 2

[12] 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

[13] Y. Ohtake, A. Belyaev, and I. Bogaevski. Mesh regularization and adaptive smoothing. *Computer-Aided Design*, 33(11):789 – 800, 2001.

[14] 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

[15] 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. 2

[16] 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

[17] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, Nov 2000. 2