Surface Reconstruction of Plant Shoots from Multiple Views

Michael P. Pound[1], Andrew P. French[1,2], Erik H. Murchie[1] and Tony P. Pridmore[1,2]

[1]Centre for Plant Integrative Biology, University of Nottingham
[2]School of Computer Science, University of Nottingham, Nottingham NG8 1BB, UK

Abstract. Increased adoption of the systems approach to biological research has focused attention on the use of quantitative models of biological objects. This includes a need for realistic 3D representations of plant shoots for quantification and modelling. We present a fully automatic approach to image-based 3D plant reconstruction. The reconstructed plants are represented as a series of small planar sections that together model the more complex architecture of the leaf surfaces. The boundary of each leaf patch is refined using the level set method, optimising the model based on image information, curvature constraints and the position of neighbouring surfaces. The reconstruction process makes few assumptions about the nature of the plant material being reconstructed, and as such is applicable to a wide variety of plant species and topologies, and can be extended to canopy-scale imaging. We demonstrate the effectiveness of our approach on datasets of wheat and rice plants, as well as a novel virtual dataset that allows us to compute distance measures of reconstruction accuracy.

Keywords. plant phenotyping, multi-view reconstruction, 3D, level sets

1 Introduction

In recent years there has been a surge in interest in the construction of geometrically accurate models of plants. Increased adoption of the systems approach to biological research has focussed attention on the use of quantitative models of biological objects and processes to both make and test hypotheses.

Existing plant modelling approaches can be broadly classified as either rule-based, or image-based [1]. Rule-based approaches generate model plants based on rules or grammars with specified structure. These rules, and hence the form and parameters of the models produced, are often derived from measurements of real plants [2,3]. The resulting virtual plants can model different phenotypes, plant response to various growing conditions and stresses, and when based on real-world data will be reasonably accurate. However, the data acquisition process is often extremely time consuming, and is usually tailored to a particular species. In many cases only a small set of varieties can be described, due to the manual measurements required to parameterise the model.

Image-based approaches attempt to directly model a given object by extracting the necessary information from one or more images of that object. These approaches are particularly attractive as a means of plant modelling, where in addition to supporting systems biology, they provide a route to plant phenotyping [4,5].

Plants, however, provide a particularly challenging subject, with large amounts of self-occlusion, and depending on plant species, leaves that lack the texture necessary to perform robust feature matching, either to separate leaves from one another, or locate specific leaves across multiple views. To overcome this, where image-based modelling approaches are successful, they have often involved user-interaction to guide the process [1].

Top-down image-based approaches attempt to simplify the model construction problem by instead solving a model refinement problem. An existing model is adjusted to fit the image data, so that the new plant representation is consistent with what is observed. [1,6] take this approach, first obtaining an ideal leaf model from a single leaf, and then fitting it to all other leaves in the scene. By adapting an existing model, topological inconsistency (such as the self-intersection of leaf surfaces) is avoided, but this comes at the expense of generality. [7] guides the segmentation of laser range data using planar or curved-quadratic surface models, however this approach extents only to the refinement of point cloud data, without reconstructing leaf surfaces.

Bottom-up methods begin with one or more images, and reconstruct a plant model based only on the observed pixel data. Two broad approaches exist, both requiring a set of images captured from different, but known, viewpoints. Silhouette-based methods [8,9] segment each image independently to identify the boundary of the object of interest. These regions are combined to determine the maximum possible object size that is consistent with the images presented to the algorithm, the *photo hull* [10]. In many cases, where the number of input images is high, the resulting model will be a good approximation to the true plant structure. However, as the scene becomes increasingly complex, for example with the addition of more leaves in an older plant, the discrepancy between true object and model will increase. This problem becomes more pronounced when extending these techniques to very complex scenes such as plant canopies, where its effectiveness is limited.

Other approaches include correspondence-based methods that identify features of interest independently in each of a set of images, and then match those features between views. If the image features associated with a particular plant feature (e.g. the tip of a leaf) can be identified in multiple images taken from different viewpoints, knowledge of the cameras' positions and orientations allow its 3D location to be computed. The work in [11] extracts the centre lines of wheat plants from two orthogonal viewpoints, improving reliability where single images would fail. This work does not, however, complete the 3D structure of each plant, preserving only the centre-line of each leaf after skeletonisation.

Image-based modelling algorithms are widely applicable and require only easily accessible and affordable cameras. Their generality can, however, become a hindrance, as the challenging nature of plant topology may require additional assumptions to be made as the reconstruction proceeds. The representations they produce may also be unsuitable for direct use in some situations. The volumetric data structures produced by silhouette-based methods, for example, are static: the size and position of the voxels are defined early in the process and are difficult to change. While measurements of e.g.

height and volume are easily made from volumetric descriptions, estimating motion e.g. of leaves moving in the breeze is extremely difficult. Similarly, point clouds can be used to calculate density and distributions of plant material, but cannot immediately be used in modelling applications where a surface-based representation is required.

This paper describes a fully automatic, bottom-up approach to image-based 3D plant reconstruction that is applicable to a wide variety of plant species and topologies. The method is accurate, providing a true representation of the original plant, and produces data in a form that can support both trait measurement and modelling techniques such as forward ray tracing [12].

An initial 3D point cloud is first described by a set of planar patches, each representing a small section of plant material, usually a segment of leaf. Where the quality of the input point cloud is high, the initial surface estimate will provide a good model of the plant. Image noise and the complexity of the plant will, however, typically lead to missing areas of leaf material, and poorly defined leaf boundaries. We therefore extend existing approaches by refining the initial surface estimate into a more accurate plant model. Each surface patch is re-sized and re-shaped based on the available image information, and positional information obtained from neighbouring surfaces. The resulting surface patches are then re-triangulated to produce a smooth and geometrically accurate model of the plant.

The reconstruction process makes few assumptions about the nature of the plant material being reconstructed; by representing each leaf as a series of small planar sections, the complete leaf surface itself can take any reasonable shape. The generality of our technique allows it to be scaled to scenes involving multiple plants, and even plant canopies. However, the focus of this paper is on the accurate reconstruction of single plants of varying species.

2 Plant Reconstruction

2.1 Input Point Cloud

The reconstruction algorithm described in this paper uses an initial point cloud estimate as a basis for the growth of plant surfaces in three dimensions. Numerous software- and hardware-based techniques exist to obtain point clouds of objects. We have chosen to make use of a software-based technique, patch-based multi-view stereo (PMVS) [13]. This approach reconstructs dense point clouds from any calibrated image set, and is not restricted to plant data. However, by including robust visibility constraints, it is well suited to plant material that contains large amounts of occlusion. Let $\{X_i\}_{i=1}^{n}$ be the set of all points in an input cloud of size n. We identify the co-ordinate system used by the point cloud, and the resulting reconstruction, as “world” co-ordinates. An individual point $p \in X$ in world co-ordinates is represented as a 3D vector $\mathbf{w}$.

A requirement of both PMVS and our reconstruction approach is that the intrinsic and extrinsic camera parameters be known. We use the VisualSFM [14] system to perform automatic camera calibration. Any number of arbitrary camera positions may be calibrated using VisualSFM, and calibration is performed quickly. However, as it is based on SIFT features [15], the approach is not suitable for images with insufficient texture and feature information. This is particularly problematic within plant datasets, where leaves may have few suitable feature points. In our datasets, the surrounding scene provides an adequate feature set for correspondence, and in our virtual dataset, we calibrate separately using a highly-textured model.

We capture N_{cam} images of the scene from N_{cam} locations to obtain a set of images $\{I_j\}_{j=1}^{N_{cam}}$. Associated with each camera location is a perspective projection matrix, based on a standard pinhole camera model [16], derived from the calibration information output by VisualSFM. For a given world point, there is a perspective projection function, $\mathcal{V}_j$, that maps onto a point in a specific camera coordinate frame, given by the 2D vector $\mathbf{v}$. This gives a set of functions $\{\mathcal{V}_j(\mathbf{w}): \mathbb{R}^3 \rightarrow \mathbb{R}^2\}_{j=1}^{N_{cam}}$. where j is the index of the input image and associated camera geometry. Once in camera co-ordinates, pixel information for a given location is represented by $I_j(\mathbf{v})$.

PMVS makes no assumptions about the nature of the objects being reconstructed. It is likely that additional points are contained in X that comprise background or other non-plant material. Many such points will be removed by our level set approach, however for computational efficiency many can be removed before reconstruction begins.

The point cloud is pre-filtered to remove obvious errors; those that differ greatly from the expected colour of the plant, or those that appear below the expected location of the plant. Two filters are applied, first a clipping plane positioned at the base of the plant is used to remove the majority of background points on the floor, container etc. Second, colour filtering is achieved by examining the projected pixel values for every point, and removing those that do not appear green in colour. These filters are meant only as a conservative first pass, a more sensitive colour-based metric is used within the speed function during application of the level set method. The final filtered point cloud $X' \subseteq X$ is used for the remainder of the reconstruction process.

2.2 Point Cloud Segmentation

The point cloud representation produced by PMVS contains no explicit surface description. Methods for the reconstruction of a surface mesh from a point cloud exist [17,18] Most, however, construct a single surface describing the entire point cloud. Plants contain complex surface geometry that encourages the separation of leaves. We also wish to approach the more general problem of plant reconstruction, without assuming the connectivity or nature of the plant leaves is known. Instead, we model plant material as a series of small planar patches. Patch size is restricted to avoid fitting surfaces between nearby leaves, and to accurately model the curved nature of each leaf surface. The filtered point cloud is first segmented into small clusters of points using a radially bounded nearest neighbour strategy [19]. Points are grouped with their nearest neighbours, as defined by a pre-set distance, and the method is extended to limit the potential size of each cluster. More formally, from the filtered cloud we obtain a set of clusters $\{C_k\}_{k=1}^{N_{clus}}$ in which each cluster contains at least one point and all clusters are disjoint, so $|C_k| > 0, \forall k$ and $C_k \cap C_l, \forall k \neq l$.

This distance used for the nearest neighbour approach is dependent on the size and resolution of the model being captured. As PMVS and laser scanning devices usually output points with a consistent density, the distance parameter can be set once and then remain unchanged between experiments using the same image capture technique. Reducing this number will increase the number of planar sections fitted to the data, increasing accuracy at the cost of decreased algorithmic efficiency.

Our surface fitting approach begins with an approximation of the surface that will then be refined. A least-squares orthogonal regression plane is fitted to each cluster using singular value decomposition. This best fit plane minimises the orthogonal distance to each point, providing each cluster with a centre point $\mathbf{c}$, a normal vector $\mathbf{n}$, and an orthogonal vector $\mathbf{x}$ indicating the rotation about the normal. The vector $\mathbf{x}$ is aligned along the major-principle axis of the point within the cluster. We then define a set of orthographic projection functions that project individual world points into each cluster plane, $\{\mathcal{C}_k(\mathbf{w}) \colon \mathbb{R}^3 \to \mathbb{R}^2\}_{k=1}^{N_{clus}}$, where $\mathcal{C}_k$ represents the projection into plane k (i.e. the plane associated with cluster C_k). We say that points projected onto any plane now occupy planar co-ordinates. Any such point, denoted by the 2D vector $\mathbf{p}$, can be projected back into world co-ordinates by the set of functions $\{\mathcal{W}_k(\mathbf{p}) \colon \mathbb{R}^2 \to \mathbb{R}^3\}_{k=1}^{N_{clus}}$.

The orthogonal projection in $\mathcal{C}_k$ has the effect of flattening the points in each cluster to lie on their best fit plane, reducing any noise in individual points, and reducing the surface fitting algorithm to a 2D problem. Point and mesh surfaces generated on a cluster plane will have an associated world position that can be output as a final 3D model.

2.3 Surface Estimation

An initial surface estimate is constructed by calculating the α-shape of the set of 2D points in planar co-ordinates. An α-shape is a generalisation of the convex-hull for a set of points, and is closely related to the commonly used Delaunay triangulation. For the incomplete leaf surfaces that exist within the input cloud, the Delaunay triangulation and convex hull represent an over-simplification of the complex boundary topology of the clusters. For a point set S, Edelsbrunner [20] defines the concept of a generalized disk of radius 1/α, with an edge between two points in S being included in the alpha shape if both points like on the boundary of the generalised disk, and that disk contains the entire point set. The set of α-shapes represent a triangulation of each surface at varying levels of detail. In this work, a negative value of α is used, with larger negative values removing larger edges or faces. The α value can be tuned for a given data set, to preserve the shape of the boundary of each reconstructed point set.

2.4 Boundary Optimisation

The α-shapes computed over each cluster form an initial estimate of the location and shape of the plant surface. The challenging nature of plant datasets in multi-view reconstruction means that in many instances the initial point cloud estimate will be inaccurate or incomplete. The initial surface boundaries based on these points will require further optimisation to adequately reflect the true shape of each leaf surface. Missing leaf surfaces should be reconstructed, and overlapping shapes should be optimised to meet at a single boundary. Many methods, such as active contours [21], parameterise the boundary of shape before attempting this optimisation. However, such approaches are ill-suited to the complex boundary conditions produced by α-shapes. For any value of $\alpha < 0$, the surface may contain holes or disjoint sections, and as such many surfaces will change topology during any boundary optimisation process.

Tracking of such complex boundaries can be achieved using the level set method [22,23]. The method defines a 3D function φ that intersects the cluster plane. φ is represented as a signed distance function, initialised such that negative values lie within our α-shape boundary, and positive values occur outside. Thus, the boundary itself is defined as the set of all points in φ that intersect the cluster plane, given as:

$$\Gamma = \{(x, y) | \varphi(x, y) = 0\} \ . \tag{1}$$

A speed function determines the rate of change of φ. It may be based on both global and local parameters, and will act to grow or shrink the boundary Γ as necessary to fit the underlying data. The change in φ, based on a speed function v, is defined as

$$\frac{\delta\varphi}{\delta t} = -v \cdot |\Delta\varphi| \ , \tag{2}$$

where Δφ is the gradient of the level set function at a given point, which we calculate through Godunov’s upwinding scheme. The speed function is defined as

$$v = v_{curve} + v_{image} + v_{inter} \ , \tag{3}$$

where v_{curve} is a measure of the local curvature, calculated using a central finite difference approximation

$$v_{curve} = \omega \cdot \frac{\varphi_{xx}\varphi_x^2 - 2\varphi_y\varphi_x\varphi_{xy} + \varphi_{yy}\varphi_x^2}{\left(\varphi_x^2 + \varphi_y^2\right)^{\frac{3}{2}}} \ . \tag{4}$$

The curvature term encourages the boundary of the level set to remain smooth. The weighting ω is required to prevent curvature from dictating the movement of the front, in cases where the boundary is already sufficiently smooth.

The image term, v_{image}, references colour information in the input images to ascertain whether the projection of the planar surface lies over regions with a high likelihood of containing leaf material. To achieve this, the function φ is discretized and uses the planar co-ordinate system, each planar point **p** maps to a position on φ, and any point on φ will have an associated planar position. By performing consecutive projections, we are able to examine the relevant location in any image of a cluster plane position. Such a projection is given as $(\mathcal{V}_j \circ \mathcal{W}_k)(\mathbf{p})\colon \mathbb{R}^2 \rightarrow \mathbb{R}^2$, where k is the cluster index, and j is the camera index. Not every image will provide a helpful view of every cluster, they may be out of the camera’s field of view, or seen at an oblique angle. One reference view is chosen from which to obtain colour information, as follows. We choose a reference image $I_R \in I$ that represents a calculated “best view” of a planar surface. Selection of the reference view begins by projecting each cluster into each camera view. Only the interiors (triangular faces) of each α-shape are projected using a scan-line rasterisation algorithm. Attached to each projected position is a z depth, calculated as the third component output from the function $\mathcal{C}_j(\mathbf{w})$ when using homogenous co-ordinates.

This z depth represents the distance that the projected point lies from the camera's image plane, and can be used to sort clusters that project onto the same location. Projections with the lowest z value are seen in front of, so occlude, those with higher z values.

The projection locations and z depths for all clusters are analysed using a series of z-buffer data structures, one z-buffer associated with each input image. We define the z-buffers as a set $\{Z_j\}_{j=0}^{N_{cam}}$, where each buffer contains pixel locations in camera co-ordinates that map directly to the corresponding image. For each image location, any cluster that can be seen in (i.e. projects onto) that point is recorded in the z-buffer. A given position $Z_j(\mathbf{v})$ contains a depth sorted list of all clusters that project into that camera co-ordinate, i.e. $Z_j(\mathbf{v}) = \ (C_0, \dots, C_n)$.

It is desirable to select camera views that contain as little interference between clusters as possible. For a given z-buffer j, and a given cluster i, we can calculate the following measure:

$$\mathcal{V}_j^{clear}(i) = |\{\mathbf{v} \mid i \ \in Z_j(\mathbf{v}) \ \wedge \ |Z_j(\mathbf{v})| = 1\,\}| \ . \tag{5}$$

The clear pixel count represents a measure of the number of pixels each cluster projects into for a given image. This value reflects both the proximity of the cluster to the camera plane, and the angle of incidence between the camera view and the cluster plane. The clear pixel counts for all projections of a given cluster i are normalised to the range [0,1]. This measure does not include pixel positions shared by other clusters, to avoid heavily occluded views affecting the normalised value. The amount of occlusion for each cluster i, in a given z-buffer j are calculated as:

$$V_j^{occluded}(i) = \frac{\left|\{\mathbf{v} \mid i \in Z_j(\mathbf{v})\backslash\{Z_j(\mathbf{v})_{(1)}\} \wedge \ |Z_j(\mathbf{v})| > 1\,\}\right|}{\left|\{\mathbf{v} \mid i \in Z_j(\mathbf{v})\}\right|} \tag{6}$$

$$\mathcal{V}_j^{occluding}(i) = \frac{\left|\{\mathbf{v} \mid i \in Z_j(\mathbf{v})\backslash\{Z_j(\mathbf{v})_{(n)}\} \ \wedge \ |Z_j(\mathbf{v})| > 1\,\}\right|}{\left|\{\mathbf{v} \mid i \in Z_j(\mathbf{v})\}\right|} \ , \tag{7}$$

Where $Z_j(\mathbf{v})_{(k)}$ is the k^th ordered element of $Z_j(\mathbf{v})$. Thus, a combination of normalised clear pixel count, occlusion and occluding percentages can be used to sort images in terms of view quality. A reference image, I_R, is chosen where:

$$R = argmax_j(\mathcal{V}_{\mathrm{j}}^{clear}(i)(1 - \mathcal{V}_{\mathrm{j}}^{occluded}(i))(1 - \mathcal{V}_{\mathrm{j}}^{occluding}(i))) \ . \tag{8}$$

When referencing pixel values using the image I_R, we use a normalised green value to measure the likelihood of leaf material existing at that location,

$$\mathcal{N}_{\mathrm{j}}(\mathbf{v}) = \frac{I_j(\mathbf{v})_{(\mathrm{green})}}{I_j(\mathbf{v})_{(\mathrm{red})} + I_j(\mathbf{v})_{(\mathrm{green})} + I_j(\mathbf{v})_{(\mathrm{blue})}} \ . \tag{9}$$

We can assume that normalised green values will be higher in pixels containing leaf material, and lower in pixels containing background. Where lighting conditions remain consistent over an image set, we can also assume that distribution of normalised green values are the same over the each image in I. However, between different image sets we cannot assume that the properties of the normalised green values are known. These properties must be ascertained before $\mathcal{N}_\mathrm{j}$ can be used to contribute to the v_{image} term in the speed function. We sample from all images those pixels that are projected into by the α-shapes, and use Rosin's unimodal thresholding approach [24] to threshold below the normalised green peak that is observed. Using this threshold, the mean and standard deviation of the peak are calculated, and used to produce an image speed function centred around the calculated threshold t, with a spread based on the standard deviation of the peak:

$$v_{image} = \begin{cases} \max\left(-1, \dfrac{\mathcal{N}_j(\mathbf{v}) - t}{2\sigma}\right), & \mathcal{N}_j(\mathbf{v}) < t \\ \min\left(+1, \dfrac{\mathcal{N}_j(\mathbf{v}) + t}{2\sigma}\right), & \mathcal{N}_j(\mathbf{v}) \geq t \end{cases}, \tag{10}$$

where t is the threshold calculated using Rosin's method, and σ is the standard deviation of the $\mathcal{N}_\mathrm{j}$ peak. A width of 2σ was chosen as a value that characterises the spread of the normalised green values.

The final component of the speed function, v_{inter}, works to reshape each surface based on the location and shape of nearby clusters. As each cluster may have different normal orientations, it is challenging to calculate their 3D intersections in terms of 2D positions in planar co-ordinates. Indeed, two nearby clusters that could be considered as overlapping, may not intersect in world co-ordinates. Instead we project each planar position into I_R, and examine the interactions in the 2D camera co-ordinate system.

Any overlapping projections are calculated by maintaining z-buffers that update as each region reshapes. The function v_{inter} is calculated such that each cluster in $Z_j(x)$ is penalised except for the front-most cluster. Thus for a cluster i, the function is calculated as:

$$v_{inter} = \begin{cases} p - v_{image}, & Z_j(\mathbf{v})_1 \neq i \\ 0, & otherwise \end{cases}, \tag{11}$$

where p is a small negative value such that the level set boundary Γ shrinks at this location. Note that the subtraction of v_{image} results in the image component being ignored where clusters are occluded.

The complete speed function is used to update each discrete position on the level set function φ. This process must be repeated until each cluster boundary has reshaped to adequately fit the underlying image data. The speed function will slow significantly as the boundary approaches an optimal shape. Where a level set boundary no longer moves with respect to the reference image (does not alter the number of projected pixels), we mark this cluster as complete and discontinue level set iterations. Any level sets that do not slow significantly will continue until a maximum time is elapsed, a parameter that can be set by the user. We typically use a value of 100-200 iterations as a compromise between computational efficiency and offering each level set adequate time to optimise.

2.5 Model Output

Once all clusters have been iterated sufficiently, each surface triangulation must be recomputed. The level set function has a known boundary that was not available during the original surface estimation. This can be used to drive a more accurate meshing approach that will preserve the contours of each shape. We use constrained Delaunay triangulation for this task [25]. A constrained triangulation will account for complex boundary shape when producing a mesh from a series of points, however it will not over-simplify the boundary by fitting surfaces across concave sections, and can retain holes in the surface if required. Points are sampled from the boundary of each surface, and a constrained triangulation is fitted. This process will automatically generate additional points, where required, within the shape itself. As each point in the new triangulation exists in planar co-ordinates, they can be easily back-projected into world co-ordinates to be output in mesh format.

3 Experimental Results

In this section we present results obtained when applying our reconstruction approach to multiple views of single plants. Verification of our approach is achieved using a novel artificial dataset, in which a model rice plant is rendered from multiple viewpoints to generate artificial colour images that are then treated in the same way as a real-world image set. This approach allows the reconstructed plant to be directly compared to the artificial target object, an impossible prospect when working with real-life plants, as no such ground truth can exist.

We have tested our reconstruction methods on datasets obtained from rice and wheat plants. Images were captured using a DSLR camera with a 35mm lens, at 8 megapixel resolution. The number, and nature of the images were left to the user to decide given the subject in question, though we recommend more than 30 images surrounding the subject for a single plant. No special consideration was given to the environment in which the plants were imaged, beyond avoiding large areas of green colour in the background. The rice dataset was captured in an indoor environment, the wheat in a glass house. These environments provide complex backgrounds, which raise additional challenges, but the plants can still be reconstructed using our methods. It is likely that a permanent installation with a more strict protocol for image capture would result in more consistent point cloud reconstruction between datasets, readers are encouraged to explore this option if using our methods over extended periods.

Fig. 1 shows the result of applying our reconstruction approach to two image sets containing wheat and rice plants. Quantitative evaluation of the effectiveness of any 3D shoot reconstruction is challenging due to a lack of ground truth models for comparison. Here we offer a qualitative evaluation of the benefits and shortcomings of our approach using live plants, followed by a quantitative evaluation using the virtual rice dataset.

First, live plants were reconstructed. Results showed that the initial surface estimate, obtained by calculating an α-shape over each cluster, will naturally reproduce any flaws present in the PMVS point cloud. Most notable are the lack of point information in areas of poor texture, and noise perpendicular to the leaf surface, where depth has not be adequately resolved. These issues can be caused by the heavy self-occlusion observed in larger plants or canopies, but are often caused in even simple datasets by a lack of image features in the centre of leaves.

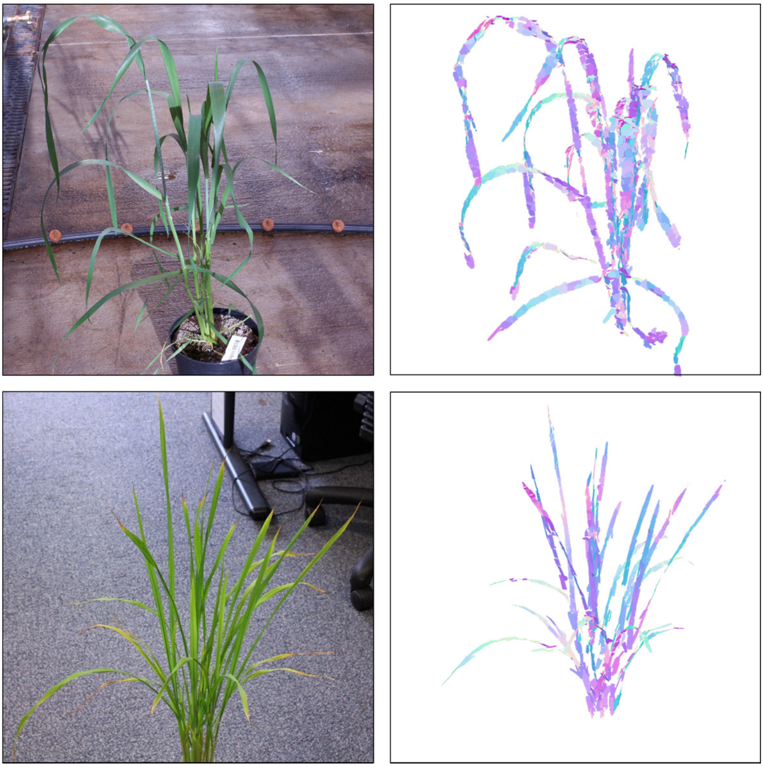

Fig. 1. Reconstruction of rice and wheat images. (*top left*) A sample image from the wheat dataset. (*top right*) A meshed reconstruction of the plant surface using our approach. (*bottom left*) A sample image from the rice dataset. (*bottom right*) A meshed reconstruction of the plant surface using our approach.

Depth noise is significantly reduced by the use of best-fit planes over small clusters, where all points are projected onto a single surface. However, the boundary of each surface is a function of the parameters used to create the α-shape, and the quality of the underlying data. As such, we can expect the α-shape boundaries to be a poor representation of the true leaf shape. With this in mind, we would characterise a successful reconstruction as one that significantly improves upon the initial surface estimate, through the optimisation of the surface boundaries.

Notable characteristics of the α-shape boundaries in both datasets are significant overlap between neighbouring clusters, and frequent missing surface sections (Fig. 2). Fig. 2 also shows the refined boundaries after the level set method has been applied, in which missing sections are filled, and overlapping surfaces have been reduced. The results in Fig. 2 are representative of the results over both datasets.

While the refined surfaces represent an improvement over both the initial point cloud, and the initial α-shape surface, there are still notable areas for improvement. By treating each section of leaf as an individually orientated plane, each plane orientation is susceptible to the error within the input cloud. Since each boundary is refined from one reference view, incorrect orientation of the best-fit plane might cause the surface boundary to be incorrectly aligned with the image, or neighbouring clusters. Consider Fig. 2 (right), in which two patches have been reconstructed in close proximity. When viewed from the reference view in which boundary refinement occurred, the boundaries of neighbouring patches are in good agreement. A rotated view of the same surfaces, however, shows that misaligned normal orientation can lead to gaps between neighbouring surfaces. Conversely, if the right-hand image had been chosen as I_R, the level set equation would increase the size of both boundaries, and overlap would be observed in the left hand view.

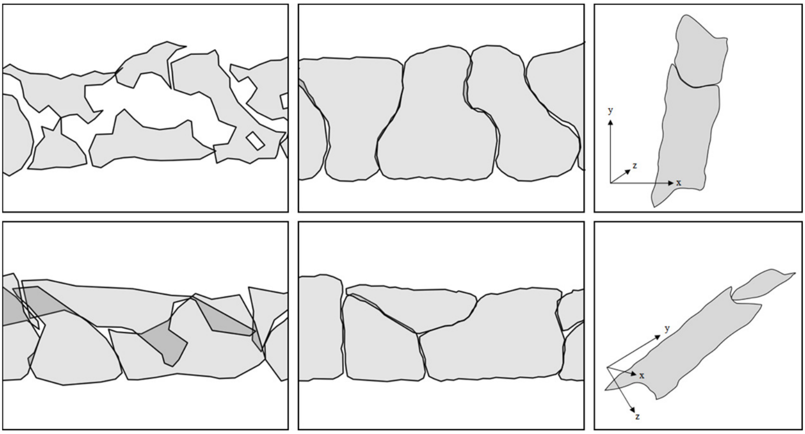

Fig. 2. Boundary refinement using the level set method. (*top left*) An initial surface estimate of a section of the wheat dataset. (*top middle*) A refined version of the wheat model after a level set was applied to each patch. (*bottom left*) An initial surface estimate of a section of the rice dataset. (*bottom middle*) A refined version of the rice model after a level set was applied to each patch. (*top right*) Two example patches, viewed from the same position as the reference image I_R. (*bottom right*) A different orientation of the same two patches.

In reality, for many clusters with very similar orientations these gaps will be negligible; as the clusters are limited in size, the distance between neighbouring plane orientations will be small, and the resulting gaps between boundaries will also be small. We have quantified the low level of discrepancy between an input model and the reconstruction below. We anticipate that further work on smoothing the normal orientations of neighbouring clusters or merging neighbouring clusters into a single curved leaf model will continue to improve results in this regard: this will be a focus of upcoming research.

An additional dataset was created based on the plant used in the rice dataset. The rice plant was first manually captured and modelled using the point cloud created by PMVS, and 3D graphics software [26,27]. This is a time consuming and subjective process, and should not be viewed as a suitable alternative to automatic reconstruction. However, it is possible to produce an easily quantifiable ground truth model that can be used as a target for automated reconstruction. This virtual plant was textured and coloured in order to emulate the original plant leaves. Finally, 40 distinct camera views of the model were rendered, simulating an image capture system moving around a static plant. The resulting dataset can then be reconstructed in the same manner as real-world data, while retaining the ability to compare the reconstruction with the original virtual plant, in particular keeping the same co-ordinate system and scale. The original model, and our reconstruction can be seen in Figure 3.

To quantify the similarity between the original model and the reconstruction, we use the Hausdorff distance, the greatest distance from any point on either mesh, to the nearest point on the other. This concept is extended in [28] to include a measure of the mean distance between two meshes.

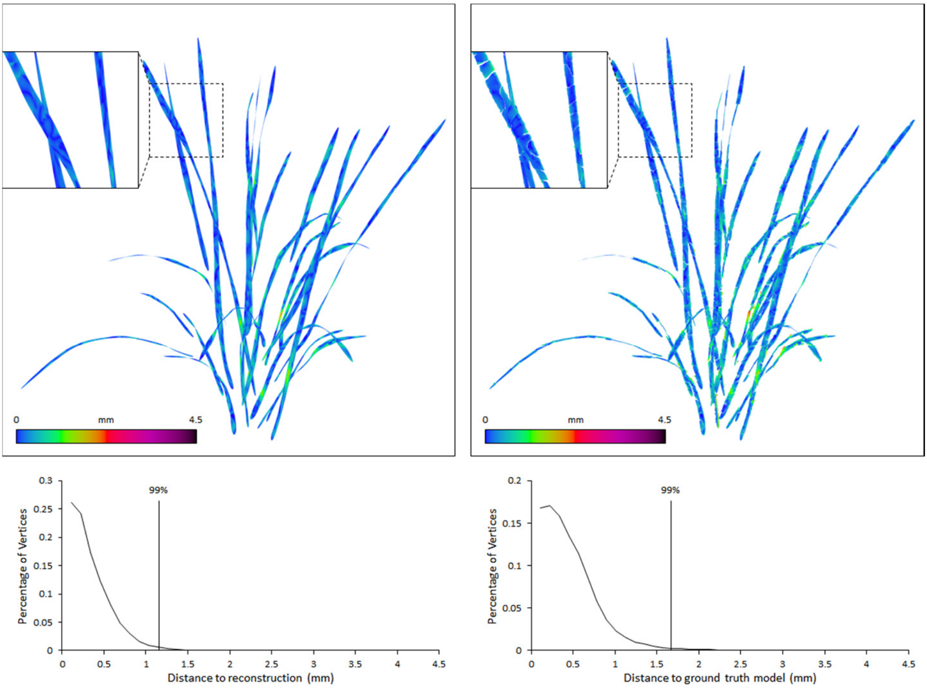

Fig. 3. (*top left*) The original rice plant model, based on the plant reconstructed in Figure 2. Vertices are coloured based on their mm distance to the nearest point on the reconstruction. (*bottom left*) Histogram of smallest distances from each vertex on the model to vertices on the reconstruction. (*top right*) The reconstruction produced by our approach. Vertices are coloured based on their mm distance to the nearest point on the original model. (*bottom right*) Histogram of smallest distances from each vertex on the reconstruction to vertices on the model.

A visual representation of these measures can be seen in Figure 3, in which each vertex is coloured based on the distance to the nearest point on the opposing mesh. This provides a visual clue as to our algorithm performance. The arbitrary world units used within the reconstruction were converted into mm measurements through the use of a calibration target of known size.

The furthest distance between points on both meshes is ~4.5mm, however the average distances between each mesh are significantly lower. The complete model is approximately 48cm tall. These one-sided measurements provide additional information, by distinguishing between the distances in either direction. Increasing distance from the model plant to the reconstruction indicates areas of the model that have not been accurately reconstructed. This is most likely where missing points in the initial cloud and surface estimates are not adequately refined through the level set method. In this case, the low mean and maximum distances show that these regions have been reconstructed successfully. Indeed, 99% of the vertices in the model are within 1.2mm of the reconstructed model.

Table 1. Distance measurements between the model plant and the results of the reconstruction approach. The two-sided Hausdorff distance is the maximum of both single-sided measurements

Vertex Distance (mm)	Model Plant	Reconstruction
Minimum	0	0
Maximum	4.576	4.496
Mean	0.289	0.411
RMS	0.379	0.534
Hausdorff Distance (mm)	4.576	

In the other direction, higher distances from the reconstruction to the original model represent areas that have deviated from the true position of the plant. This could be caused by a number of factors, such as misalignment between the orientation of a surface plane and the original surface, or surface boundaries extending beyond the true boundary of the leaves, possibly due to occlusion. The maximum and mean distances for the reconstruction remain low, and show that the reconstruction is a good reflection of the true model.

Table 2. Details and processing times for the datasets evaluated in this section. Each level set was iterated to a maxmimum of 100 times, or until it halted

Dataset	Cluster Count	Image Count	Time Taken
Rice	785	36	5m34s
Wheat	1486	62	18m28s
Model Rice Plant	517	40	2m11s

The mean distance and RMS error for this single-sided measure is higher than the reverse, which we believe may represent current technical limit of our approach. The distances around the boundaries of many surfaces appear slightly higher than in the centre, where the level sets can over-extend the leaf edge. This is a limitation within the level set speed function, but for the distances observed this usually represents an increase of size, outwards, of less than a pixel on average when projected into the reference image. This sub-pixel accuracy is not resolved by the speed function of the level set method that we use. An immediate improvement could be observed by simply increasing the resolution of the input image set, however this would add significant computational overhead.

The performance of our approach is closely related to the size of the image set, and the number of surface segments being evaluated. For small datasets, reconstruction usually takes a matter of minutes. For complex datasets containing thousands of small surface patches, we can expect performance to decrease. Table 2 shows details and processing times for the datasets evaluated in this section. Tests were run on an Intel Core i7 3820 machine. The algorithms detailed here are suitable for GPU parallelisation in the future if further optimisation is required.

4 Conclusions

The recovery of accurate 3D models of plants from colour images is challenging. A single plant constitutes a crowded scene in the sense of [13] and the construction of accurate 3D models of objects of this level of complexity is an active research topic. Images of plants exhibit high degrees of occlusion, with the occlusion relations between leaves varying from image to image. To complicate matters further, individual leaves are difficult to identify: most of the leaves on a given plant have similar colour and texture properties. Rather than address these issues in a single process that transforms a set of images into a three-dimensional model via feature correspondence or silhouette analysis, the approach presented here develops each leaf segment individually, automatically selecting an image likely to contain the necessary information. The proposed method reduces the effect of occlusion by choosing an image with a clear view of the target surface, and addresses the similarity problem by performing detailed analysis of the colours present in that image.

The mesh representation produced provides a detailed model of the surface of the viewed plant that can be used both in modelling tasks and for shoot phenotyping. At present the surface description output by the proposed technique comprises a large set of distinct planar patches, but it is anticipated to be a simple process to extend this to curved surfaces describing whole leaves if necessary. The level set method re-sizes and re-shapes each patch to maximise its consistency with neighbouring patches and the selected image, and as such the reconstructed patches provide an accurate approximation of the leaf surfaces. By avoiding a model fitting solution, the approach also remains general, and is flexible enough to be applied to a wide variety of plant species with differing leaf shapes.

References

1. Quan, L., Tan, P., Zeng, G., Yuan, Lu., Wang, J., Kang, S.B.: Image-based plant modeling. ACM Transactions on Graphics **25**(3) (2006) 599-604
2. Watanabe, T., Hanan, J.S., Room, P.M., Hasegawa, T., Nakagawa, H., Takahashi, W.: Rice morphogenesis and plant architecture: measurement, specification and the reconstruction of structural development by 3D architectural modelling. Annals of botany **95**(7) (2005) 1131-1143
3. Alarcon, V.J., Sassenrath, G.F.: Modelling cotton (Gossypium spp.) leaves and canopy using computer aided geometric design (CAGD). Ecological Modelling. **222**(12) (2011) 1951-1963
4. Houle, D., Govindaraju, D.R., Omholt, S.: Phenomics: the next challenge. Nature Reviews Genetics **11**(12) (2010) 855-866
5. White, J.W., Andrade-Sanchez, P., Gore, M.A., Bronson, K.F., Coffelt, T.A., Conley, M.M., Feldmann, K.A.: Field-based phenomics for plant genetics research. Field Crops Research **133** (2012) 101-112
6. Ma, W., Zha, H., Liu, J., Zhang, X., Xiang, B.: Image-based plant modeling by knowing leaves from their apexes. 19th International Conference on Pattern Recognition (2008) 1-4
7. Alenya, G,. Dellen, B., Torras, C.: 3D modelling of leaves from color and ToF data for robotized plant measuring. IEEE International Conference on Robotics and Automation (ICRA) (2011) 3408-3414
8. Clark, R.T., MacCurdy, R.B., Jung, J.K., Shaff, J.E., McCouch, S.R., Aneshansley, D.J., Kochian, L.V.: Three-dimensional root phenotyping with a novel imaging and software platform. Plant Physiology **156**(2) (2011) 455-465
9. Kumar, P., Cai, J., Miklavcic, S.: High-throughput 3D modelling of plants for phenotypic analysis. Proceedings of the 27th Conference on Image and Vision Computing New Zealand (2012) 301-306
10. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. International Journal of Computer Vision **38**(3) (2000) 199-218
11. Cai, J., Miklavcic, S.: Automated extraction of three-dimensional cereal plant structures from two-dimensional orthographic images. Image Processing **6**(6) (2012) 687-696
12. Qingfeng, S., Guilian, Z., Xin-Guang, Z.: Optimal crop canopy architecture to maximise canopy photosynthetic CO2 uptake under elevated CO2 – a theoretical study using a mechanistic model of canopy photosynthesis. Functional Plant Biology **40** (2013) 108–124
13. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(8) (2010) 1362-1376
14. Wu, C.: VisualSFM: A visual structure from motion system (2011)
15. Lowe, D.G. Object recognition from local scale-invariant features. The proceedings of the Seventh IEEE International Conference on Computer Vision **2** (1999) 1150-1157
16. Hartley, R., Zisserman, A.: Multiple View Geometry in computer vision. Cambridge University Press (2003)
17. Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C., Evans, T.R.: Reconstruction and representation of 3D objects with radial basis functions. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (2001) 67-76
18. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. Proceedings of the fourth Eurographics symposium on Geometry processing (2006)
19. Klasing, K., Wollherr, D., Buss. M.: A clustering method for efficient segmentation of 3D laser data. International Conference on Robotics and Automation (ICRA) (2008) 4043-4048

20. Edelsbrunner, H., Kirkpatrick, D.G., Raimund, S.: On the shape of a set of points in the plane, IEEE Transactions on Information Theory **29**(4) (1983) 551–559
21. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International journal of computer vision **1**(4) (1988) 321-331
22. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. Journal of computational physics **79**(1) (1988) 12-49
23. Sethian, J.A.: Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. Cambridge university press **3** (1999)
24. Rosin, P.L. Unimodal thresholding. Pattern recognition **34**(11) (2001) 2083-2096
25. Shewchuk J.R., Delaunay Refinement Algorithms for Triangular Mesh Generation, Computational Geometry: Theory and Applications **22**(1-3) (2002) 21-74
26. SC Pixelmachine SRL: Topogun, v2.0. www.topogun.com
27. Blender Foundation: Blender, v2.69. www.blender.org
28. Cignoni, P., Rocchini, C., Scopigno, R.: Metro: Measuring error on simplified surfaces. In Computer Graphics Forum **17**(2) (1998) 167-174