

Three Dimensional Reconstruction of Scenes from Multiple Views using Active Vision

Jonathon A. Gibbs, BSc.

School of Computer Science, The University of Nottingham

**Thesis submitted to the University of Nottingham for the degree of
Doctor of Philosophy**

Abstract

The need to understand the mechanisms underlying the growth of plants and crops (plant phenotyping) is becoming increasingly important in society, particularly as the quantity of food and biofuel will need to double to meet the demands of the expanding global population, which is likely to exceed nine billion by 2050.

The practical aim of this research is to contribute to reducing the bottleneck associated with plant phenotyping by generating a fully automated response to photometric data acquisition and the recovery of three-dimensional models of plants from multiple views without the dependency of botanical expertise, ensuring a non-intrusive and non-destructive approach.

Plants are complex objects displaying high degrees of concavity and self-occlusion and can be considered examples of crowded scenes. This unavoidable complexity makes careful camera placement a necessity. If a complete 3D reconstruction is to be achieved, viewpoints must be chosen to reflect the broad 3D structure of the plant.

Within this thesis, an Active Vision Cell (AVC), consisting of a camera-mounted robot arm, turntable and automatic image acquisition technique is proposed, along with a novel surface reconstruction algorithm. This approach provides a robust, flexible and accurate approach to automating 3D reconstruction of plants. The active vision method exploits volumetric shape representations to provide a compact image set well-suited to multi-view stereo. The reconstruction algorithm can reduce noise and provides a promising and extendable framework, improving on the current state-of-the-art. Furthermore, the pipeline can be applied to any plant species or form due to its application of an active vision framework combined with the automatic detection of key parameters for surface reconstruction.

Acknowledgments

I would like to thank Tony Pridmore for giving me the opportunity to pursue a lifelong dream of doing a PhD and for the guidance and support that he provided over a very fun, but stressful journey. I'd also like to thank Michael Pound, Andrew French, Darren Wells and Erik Murchie for all the support they have provided me with over the years. I'd also like to thank Joe Best for providing guidance and support throughout my four years here (and also a good gym partner).

I gratefully acknowledge the funding I received from Engineering and Physical Sciences Research Council which ensured it was possible to pursue my PhD.

My thanks go out to my friends, Amy, Sanjay and Chris, and partner, Alex, for tolerating me during this time; their patience over the years and moral support when it was very much needed helped me through the stressful times.

Most importantly I'd like to thank, and dedicate this, to my Mum who has always been there for me. You have always been a perfect role model and offered guidance whenever I needed it. You always believed in me, especially when I did not, and I certainly could not have done this without you.

Contents

Introduction	1
1.1 Background	1
1.2 Motivation.....	3
1.3 Research Aims	6
1.4 Contributions.....	7
1.5 Publications Arising from this Work	7
1.6 Synopsis of the Thesis.....	8
Background	10
2.1 Introduction.....	11
2.2 Plant Modelling.....	11
2.2.1 Plant Modelling Approaches	12
2.2.2 Semi-Automated Approaches	17
2.3 Image-based Plant Modelling	17
2.3.1 Active Methods.....	18
2.3.2 Passive Methods	21
2.4 Active Vision	30
2.4.1 Calibration	31
2.4.2 Next Best View.....	37
2.5 Discussion	42
Cell Composition.....	44
3.1 Hardware	45
3.2 Software Design.....	47
3.3 Calibration.....	49
3.3.1 Camera and Robot Calibration	51
3.3.2 Sensor and Base Calibration	54
3.3.3 Turntable calibration.....	56
Active Vision for Multi-View Stereo	60
4.1 Introduction.....	61
4.2 Approach.....	64
4.3 Plant Proxy Representation	65
4.4 Phase 1: Initial Image Capture	67
4.5 Phase 2: Image Selection	71
Stage 1: Clustering.....	71
Stage 2: Cluster Evaluation	73

Stage 3: Camera Placement	75
4.6 Phase 3: Image Removal	78
4.7 Results	78
4.8 Discussion	87
Point Cloud Processing.....	89
5.1 Introduction	90
5.2 Phase 1: Model Merging	92
5.3 Phase 2: Cluster Analysis	99
Step 1: Normal	102
Step 2: Separation	106
Step 3: Reclassification	108
5.4 Results	109
5.6 Discussion	117
Surface Reconstruction	119
6.1 Introduction	120
6.2 Phase 1: Triangulation	123
6.3 Phase 2: Level Sets	125
6.4 Phase 3: Mesh modifications	126
6.5 Phase 4: Cell Merging	129
6.6 Results	132
6.7 Discussion	147
Conclusions and Future Work	149
7.1 Main Contributions	150
7.2 Research Outcomes	150
7.3 Future Work	153
7.4 Summary	155
Bibliography	157

Table of figures

Figure 2.1 Three-dimensional modelling classification and uses for plant reconstruction shaded according to the key.....	13
Figure 2.2 An example of a laser scanning setup where radiation is emitted into the scene and the feedback is recoded by the camera (Zolfaghari and Khalili, 2008).	18
Figure 2.3 Plant reconstruction using laser scanning of <i>Ficus benjamina</i>) conducted by (Sanz-Cortiella <i>et al.</i> , 2011) in a controlled environment.	19
Figure 2.4 During the shape-from-silhouette algorithm, the visual hull is the result of two or more 2D silhouettes, from images, being projected. The resulting object is a 3D representation of the object (Laurentini, 1994).	22
Figure 2.5 During space carving, a bounding box encapsulates the object ensuring the furthest points are contained within the box.	23
Figure 2.6 An example of Space carving, removing voxels to create the 3D object. A. the initial bounding box, B. space carving being applied using first view of the object, C. the result of the finished space carving product from two views with occlusions and empty voxels still remaining (ETH Zurich).....	23
Figure 2.7 Stereo vision is the process of capturing images simultaneously of some object where each view differs slightly (G Calin and Roda, 2007).	25
Figure 2.8 Correspondence matching between one feature in two images taken at varying viewpoints.....	26
Figure 2.9 The Epipolar constraint is used to aid correspondence matching by reducing a two-dimensional problem to a one-dimensional search using the epipolar line.	27
Figure 2.10 The false boundary problem; the object appears to be much longer and wider due to a lack of views resulting in a false boundary.	28
Figure 2.11 Pinhole Camera Model; the projection of a three-dimensional point onto a two-dimensional plane (Zhang, 2003).	31
Figure 2.12 Forward and Inverse Kinematics to convert between joint space and Cartesian space	33
Figure 2.13 Denavit-Hartenberg (DH) Parameters specify how links to the end effector are connected, a) depicts the DH Parameters table, specifying four parameters for each joint, b) a visual representation of each of the four parameters (Denavit 1955).....	34
Figure 2.14 The relationship between transformations of matrix equation $AX=YB$ for the calibration of the robot and camera.	35
Figure 2.15 <i>Normal Algorithm</i> used to calculate normal scores for x, -x, y -y, z, -z; the six possible normals.....	40

Figure 3.1 The three main hardware components that make up the active vision cell; a. LT360 Ex Turntable, b. The UR5 robot arm, c. The Canon 650D camera	46
Figure 3.2 Communication between the devices is performed using an API, RS-232 and Sockets all commands come from the server which controls the functionality of all devices	47
Figure 3.3 The user interface for controlling the active vision system; Top: Main screen for loading. Middle: The active vision interface for manipulating the camera, robot and turn table. Bottom: The object loader to view and inspect the object reconstructions	50
Figure 3.4 Results of estimating the camera parameters using X and Y and the error of re-projection in pixels. a) Dornaika - Quaternion approach (Dornaika and Horaud, 1998), b) Shah - Kronecker Product (Shah 2013), c) Li - Dual-quaternions and Kronecker Product (Li et al. 2010). Where the x-axis is the error and the y-axis is the total number of images	56
Figure 3.5 Illustration of the world coordinates (0,0,0) at four calibrations for 0, 70, 180 and 250	57
Figure 3.6 The centre of rotation is determined by solving the line intersection problem using 4 unique camera calibrations and projecting into a single 2D space.	58
Figure 4.1 The CT scans for the seven plants used in this research, from left top to bottom. Column 1; Bromeliad (<i>Vriesea sp.</i>), <i>Aloe vera</i> , Cordyline (<i>Cordyline sp.</i>), Brassica (<i>Brassica napus</i>). Column 2; Chilli (<i>Capsicum sp.</i>), Pumpkin (<i>Cucurbita pepo</i>), Sugar Beet (<i>Beta vulgaris</i>). X-ray images were obtained using a GE v tome x M scanner housed in the University of Nottingham's Hounsfield Facility. Following noise reduction with a median filter, plant material was identified by applying a user-defined threshold to the image.....	63
Figure 4.2 Representation of an octree, a hierarchal space partition data structure which is subdivided to increase the knowledge of the object.....	66
Figure 4.3 Represents the three levels in which data is acquire for the proxy. The lowest camera position is in line with the turntable, the middle camera is the height of the object determined from the proxy where the camera is angled to look at the centre of the object and the top camera is double the height of the object.....	67
Figure 4.4 Top Left: Original image, Top right: Image separated from background with discarded outliers, Bottom Left: Evaluation of a pixel, Bottom right: The resulting border defining values RX, Lx, Tx, BX	69
Figure 4.5 left; an original RGB image captured using the cell, centre; the result of the euclidean colour filtering, separating the background and foreground, right; the binary matrix represented as a series of 1s.	70

Figure 4.6 Proxy representation; Left, a level 1 PPR, the initial proxy; Right, the final PPR showing more object features after acquiring additional viewpoints	71
Figure 4.7 Illustration of the voxel clustering algorithm where point X is placed in the top left cluster as the distance is minimal with respect to the other three clusters	73
Figure 4.8 Comparison of Uniform Ray Tracing (URT) versus Hierarchical Ray Tracing (HRT). a. the traditional Uniform Ray Tracing (URT) approach, where a set of dense rays are projected from the camera into the scene. b. the first stage of HRT, where a series of sparse rays are projected from the camera. c. the second stage of HRT, only when a voxel is hit does the ray resolution increase (Vasquez-Gomez et al. 2013)....	74
Figure 4.9 The incremental view sphere for determining camera positions. The red dot in the centre illustrates the optimal viewpoint defined by the normal of the cluster, the green is the first increment and the yellow the second. This continues across the entire view sphere	77
Figure 4.10An illustration of the different camera setups that are commonly used in plant phenotyping systems. In each case the plant is rotated, or circled by the user, and images are obtained. In the instance of one static a single viewpoint is used around the whole plant.	79
Figure 5.1 Left; the original point cloud data consisting of outliers. Centre; the PPR representation containing unverified voxels. Right; the merged model with the removal of outliers and voxels.....	91
Figure 5.2 The four representations. From left to right; 1. the original points including outliers outside of the voxel, 2. The voxel centre, 3. The all within method 4. The average method.....	94
Figure 5.3 Model merging results on a Bromeliad (<i>Vriesea sp.</i>). From left to right; 1. The original data, 2. The all within representation, 3. The centres representation and 4. The average representation	95
Figure 5.4 Model merging results for <i>Aloe vera</i> . From left to right; 1. The original data, 2. The all within representation, 3. The centres representation and 4. The average representation.....	96
Figure 5.5 Model merging results on a Cordyline (<i>Cordyline sp.</i>). From left to right; 1. The original data, 2. The all within representation, 3. The centres representation and 4. The average representation	97
Figure 5.6 Illustration of how a noisy point set of a wheat plant (left) can be merged with a volumetric model (middle) to produce a more accurate point set (right)	99
Figure 5.7 Illustration of the main constraints that need to be satisfied; from left to right, 1. the normal constraint, 2. the separation constraint, and 3. the outlier constraint.....	99

Figure 5.8 PCA eigenvalue decomposition illustrating the <i>min</i> , <i>mid</i> and <i>max</i> eigenvalues	103
Figure 5.9 Illustration of novel clustering algorithm. As you move along the array, the difference between the sum of angles on either side is calculated. When the minimum difference is found, the position to split the points is assigned.	104
Figure 5.10 An example of the proposed clustering algorithm (Algorithm 5.2) and how it works on a simple curve where <i>n</i> represents the normal for that position and <i>Min</i> (<i>A</i> – <i>B</i>) is the minimum deviation between angles and is where the cluster is split	105
Figure 5.11 An example of the proposed normal clustering algorithm for a complex curve. From left to right and top to bottom: the designated order in which the algorithm splits the point set and the final cluster set (bottom right).	106
Figure 5.12 Comparison of the original (Left) versus the modified (Right) DIVFRP algorithm. Use of the original DIVFRP algorithm leads to red connection across each side whereas the modified DIVFRP algorithm correctly partitions both datasets.	107
Figure 5.13 The nearest neighbour voxels, illustrating the surrounding 26 neighbours (blue) that a target voxel (grey) can have.	108
Figure 5.14 Example of cluster Segments. A rice plant (<i>Oryza sativa</i>) was segmented and 6 segments of varying complexity have been selected. The number of clusters and points are known, which is used as the basis for ground truth.	110
Figure 5.15 An example of the proposed clustering algorithm applied to a rice point set obtained using the AVC.....	118
Figure 6.1 The surface reconstruction process; 1. The starting clusters, 2. The triangulated clusters (cells), 3. Level sets applied to cells, 4. Smoothing of the cells, 5. Cell merging, 6. Final smoothing step	122
Figure 6.2 Delaunay Triangulation - representing correct (left) and incorrect (right) triangulation, (Peterson, 1998).	123
Figure 6.3 Delaunay Triangulation and the Voronoi Diagram, (Peterson, 1998).....	123
Figure 6.4 Boundary detection; A. The external boundary, B. The internal boundary which indicates the presence of any holes.....	127
Figure 6.5 The Hole Filling Process. A. The original triangulation, B. The extracted internal boundary points and centroid, C. The triangulation of the hole, D. The insertion of the new triangulation into the original surface	128
Figure 6.6 The merging method. A. The original triangulated clusters, B. External boundary detection, C. The extraction of boundaries and insertion of centroid, D. The triangulation of the boundary points, E. Extraction of triangles that contain a boundary point from both clusters, F. The merging of the original clusters and the new triangulated surface.....	131

Notation

P	A set of points, in 2 or 3 dimensions (specified throughout)
$\vec{p}$	An element of P representing a single point
V	A set of voxels
C	A set of clusters
$\vec{t}$	A 3x1 translation vector
R	A 3x3 rotation matrix
K	A 3x3 matrix of intrinsic camera parameters
A	Transformation matrix of camera
B	Transformation matrix of robot base to end
X	Transformation matrix of robot hand to eye
Y	Transformation matrix of world to base
E	Extrinsic camera matrix
T	The transformation between each joint of the robot
$\overrightarrow{Rv}$	A 4x1 rotation vector
Q	A 4x1 axis angle

Glossary

Active imaging methods	An imaging method which uses controlled radiation, for example a laser scanner
Active vision	An active vision system is one that can manipulate the viewpoint based on some knowledge
Euclidean colour filter	Filters pixels which have a colour outside of an RGB sphere with a specified centre and radius
Euclidean distance	Is the straight line distance between two points
Octree	A tree data structure in which each internal node has exactly eight children
Phenotype	A set of observable characteristics of a plant i.e. height, area
Plant proxy representation	(PPR) Is a volumetric model of the plant being modelled during the next best view stage
View sphere	A series of camera positions in a sphere shape around the object such that the objects is encapsulated in each view
Incremental view sphere	A point on a view sphere that incrementally expands when criteria is not met
Volumetric model	A 3D object comprised of a series of cubes, voxels.
Voxel	A cube in 3D space
Active Vision Cell	Refers to the entire active vision cell including the hardware and software relating to the acquisition of images
CT Scanner	Is the scanner situated at Hounsfield, Nottingham, used for obtaining X-Rays of plants used as a ground truth in this work

Tables

Table 3.1 DH-Parameters for UR5. Constant link length a_i , link twist α_i and link offset d_i and the variable joint angle θ_i	52
Table 4.1 Experiment One: evaluation of the AVC versus traditional image capture methods on a Bromeliad (<i>Vriesea sp.</i>).....	80
Table 4.2 Experiment Two: evaluation of the AVC versus traditional image capture methods on <i>Aloe vera</i>	81
Table 4.3 Experiment Two: evaluation of the AVC versus traditional image capture methods on a Cordyline (<i>Cordyline sp.</i>).....	82
Table 4.4 Experiment Four: evaluation of the AVC versus traditional image capture methods on a Brassica (<i>Brassica napus</i>)	83
Table 4.5 Experiment Five: evaluation of the AVC versus traditional image capture methods on a Chilli plant (<i>Capsicum sp.</i>).....	84
Table 4.6 Experiment Six: evaluation of the AVC versus traditional image capture methods on a Pumpkin (<i>Cucurbita pepo</i>).....	85
Table 4.7 Experiment Seven: evaluation of the AVC versus traditional image capture methods on Sugar beet (<i>Beta vulgaris</i>).....	86
Table 5.1 Comparison of merging results for Experiment One: a Bromeliad (<i>Vriesea sp.</i>)	95
Table 5.2 Comparison of merging results for Experiment Two: <i>Aloe vera</i>	96
Table 5.3 Comparison of merging results for Experiment Three: a Cordyline (<i>Cordyline sp.</i>)	97
Table 5.4 Comparison of the proposed clustering algorithm relative to existing clustering algorithms for Segment One.	111
Table 5.5 Comparison of the proposed clustering algorithm relative to existing clustering algorithms for Segment Two.....	112
Table 5.6 Comparison of the proposed clustering algorithm relative to existing clustering algorithms for Segment Three.....	113
Table 5.7 Comparison of the proposed clustering algorithm relative to existing clustering algorithms for Segment Four.	114
Table 5.8 Comparison of the proposed clustering algorithm relative to existing clustering algorithms for Segment Five.....	115
Table 5.9 Comparison of the proposed clustering algorithm relative to existing clustering algorithms for Segment Six.....	117

Table 6.1 Experiment One: evaluation of the proposed surface reconstruction algorithm applied to the AVC image set versus the canopy reconstruction algorithm (Pound et al. 2014) on a Bromeliad (<i>Vriesea sp.</i>).....	134
Table 6.2 Experiment Two: evaluation of the proposed surface reconstruction algorithm applied to the AVC image set versus the canopy reconstruction algorithm (Pound et al. 2014) on <i>Aloe vera</i>	136
Table 6.3 Experiment Three: evaluation of the proposed surface reconstruction algorithm applied to the AVC image set versus the canopy reconstruction algorithm (Pound et al. 2014) on a Cordyline (<i>Cordyline sp.</i>).....	138
Table 6.4 Experiment Four: evaluation of the proposed surface reconstruction algorithm applied to the AVC image set versus the canopy reconstruction algorithm (Pound et al. 2014) on a Brassica (<i>Brassica napus</i>).....	141
Table 6.5 Experiment Five: evaluation of the proposed surface reconstruction algorithm applied to the AVC image set versus the canopy reconstruction algorithm (Pound et al. 2014) on a chilli (<i>Capsicum sp.</i>).....	143
Table 6.6 Experiment Six: evaluation of the proposed surface reconstruction algorithm applied to the AVC image set versus the canopy reconstruction algorithm (Pound et al. 2014) on a pumpkin plant (<i>Cucurbita pepo</i>)	145
Table 6.7 Experiment Seven: evaluation of the proposed surface reconstruction algorithm applied to the AVC image set versus the canopy reconstruction algorithm (Pound et al. 2014) on Sugar beet (<i>Beta vulgaris</i>).....	147

Chapter 1

Introduction

1.1 Background

The recovery of three-dimensional (3D) descriptions of viewed objects from multiple images, also known as 3D modelling, is a longstanding problem in computer vision. While in recent years improvements with respect to both quality and performance have been made, the methods are often applied to trivial objects, usually man-made and often symmetrical, convex or, with relatively simple curvature and straightness attributes.

Crowded scenes, in which multiple, closely-packed objects occlude each other in many views, are beyond the current state-of-the-art. They constitute a more difficult challenge and are troublesome to accurately represent. Crowded scenes generate high levels of *occlusion* - where part of the object is not visible from the current view - and *parallax* - the effect of the object appearing to differ when viewed at different angles, making accurate reconstruction exponentially harder than for simple convex objects.

Image acquisition and selection are currently an insufficiently considered resource in Multi View Stereo (MVS) and are particularly important when the target objects are complex. MVS uses feature matching between multiple overlapping views of an object or scene to create a disparity map that is used to determine the position of the object in 3D space. Seitz *et al.*, 2006 demonstrate that the quality and speed of 3D modelling depends significantly on the quantity and selection of input images, and that each image does not contribute evenly to the overall quality of the model. Some images may contain redundant areas, significantly increasing processing time, while others may lack the detail necessary for feature matching, resulting in poor quality reconstructions.

These effects are amplified when attempting to reconstruct crowded scenes, making acquisition of the images needed to accurately represent the target object a major challenge. Without sufficient images, a faithful representation cannot be modelled, though, on the other hand, without a prior representation of the object, it is difficult to evaluate whether sufficient data has been acquired. For crowded scenes, even when a large number of images are available, there is a high probability that some 3D data will still be missing.

Point data from MVS is typically used to create a surface mesh representation. However, missing point data leads to mesh defects (Ju, 2009) which can be divided into two categories: *geometric* and *topological*. Geometric errors include non-manifold edges; edges that have more than two faces joined to a single edge, small and large gaps and duplicated vertices, while topological errors include simple and complex holes plus missing data/surfaces. Further imperfections in point clouds are expected due to the reconstruction process; the location of the points may be perturbed by unknown levels of noise, or the surface may not be well sampled, leading to holes, gaps and missing data. The point cloud may include outliers; points lying away from the object surface that are not part of the object being modelled. Further difficulties arising from the

complexity of the object include the need to accurately detect and preserve the boundaries of objects and the significantly larger processing times incurred when dealing with complex shapes.

In this thesis an automated approach to the reconstruction of crowded 3D scenes from 2D image data is proposed. The method combines MVS and volumetric reconstruction techniques with active image acquisition to produce rich descriptions of highly complex and variable natural objects: plants.

1.2 Motivation

Current 3D reconstruction work typically focuses on modelling man-made objects. Models of human artefacts are generally easier to produce, as constraints based on, for example, the curvature or straightness and symmetry of such objects, can be applied. Natural objects are often significantly more challenging than man-made structures, as the processes which created them, and the resulting structures, are much more complex. One particularly important class of natural objects are plants, which, due to their indeterminate growth, can be of widely varying shape and size. Plants comprise large numbers of leaves that often overlay each other, resulting in a high degree of occlusion and making plants very crowded scenes.

The need to understand the mechanisms underlying the growth of plants and crops is increasingly important in society. The quantity of food and biofuel available will need to double in order to meet the demands of the expanding global population, which is likely to exceed nine billion by 2050 (Sticklen, 2007; Faaij, 2008; Paproki *et al.*, 2012). The Food and Agriculture Organisation of the United Nations (FAO) have already defined in ‘The State of Food Insecurity in the World 2014’, that 805 million people in the world are considered ‘chronically undernourished’. This is the equivalent to one in nine people. To reduce this figure, an integrated approach to increasing crop yield is required, including improved agricultural productivity and increased access to new technologies.

The severity of the current situation is such that rice demand has recently exceeded supply for over two years (2009-2011) (Furbank, R. T., & Tester, 2011) and world stocks of grains are the lowest seen in 45 years (Furbank *et al.*, 2009). Changes in climate and the shortage of arable land constitute additional challenges for sustainable agriculture. Furthermore, the effects of global

warming have been shown to cause more frequent and severe flooding and drought, devastating crop yields, and amounting to an even larger deficiency of food (Adeloye, 2010). Recent work has shown that without crop climate adaption, productivity will actually deteriorate (Challinor *et al.*, 2014).

Current methods of increasing crop production are often environmentally damaging; destroying forestry and grasslands to create room for more livestock or crops. We cannot keep expanding to produce more food for an ever-increasing population. Consequently, additional pressure is being placed on agriculture to increase food production on the same amount, if not less, land, presenting a vast challenge for the plant science and agricultural community who are tasked with identifying methods to improve crop yields (Sutton *et al.*, 2011). One rapidly emerging area of research is focusing on the adaptability of plants, by conducting controlled experiments to analyse performance when exposed to various stresses and/ or environmental changes.

Phenomics, or phenotyping, is the study of plant growth, performance and composition, looking at the functional plant body and its development over time due to the dynamic interaction with the genetic background (genotype) and the environment. These interactions are measured to determine performance and tolerance to different ecological and physical environmental conditions. Examples of phenotypic parameters include height, leaf counts, photosynthetic performance, crop yield and nutrient uptake.

At present, phenotyping constitutes a major bottleneck in the construction of quantitative models of plant development and pre-breeding applications (Preuksakarn *et al.*, 2010). With insufficient capabilities of analysing the plant interaction with the environment, understanding the link between genotype and phenotype is unattainable. A comprehensive understanding of the range of plant phenotypes that are possible would aid breeding and genetic modification, facilitating the increase of nutrient use and photosynthetic efficiency, and thereby increasing crop yield and stability across diverse environments (Quan *et al.*, 2006).

In recent years there has been a rapid increase in interest in gathering phenotypic information through the automatic extraction of plant traits from individual and sequences of images. The recovery of accurate 3D models of plants from such images is urgently required. High quality 3D models can provide a wide range of morphological and gross developmental data. These models can be used to

support both simulations of plant function and the extraction of the trait measurements required to allow phenotyping. While image-based modelling has made significant progress over the past decade, creating accurate representations of plants is still incredibly challenging.

A number of companies now offer phenotyping systems, differing in complexity from small lab-based equipment, to larger scale glasshouse- and even field-based systems (Photon Systems Instruments, Drasov, Czech Republic; LemnaTec, Aachen, Germany; Qubit Phenomics, Kingston Ontario, Canada). These platforms often use robotics and imaging technologies (in the form of both hardware and software) to assess plant phenotypes. They are commonly based around the use of RGB, thermal, hyperspectral or fluorescence cameras or laser scanners and differ in whether the sensors are transported to the plant, or the plant to the sensor. For example, the PlantScreen Robotic XYZ system (Photon Systems Instruments, Drasov, Czech Republic) or field-based systems such as the PlantScreen field systems (Photon Systems Instruments, Drasov, Czech Republic; Qubit Phenomics, Kingston Ontario, Canada) and the Field Scanalyzer (LemnaTec, Aachen, Germany) contain sensors mounted overhead or gantries, which travel to the target plants. Alternatively, modular systems exist, often using conveyor belts or other robotics to transfer the plants to the sensors mounted in fixed positions. Such systems are generally bespoke which fixed sensor positions and are designed for a limited number of species or growth forms making them of limited application to a wider research sphere.

Limited phenotyping systems offering the production of 3D plant models are available such as the IMAG-3D (Walz, Effeltrich, Germany) which uses projected light and a fixed camera placed above the plant to create a 3D mesh; limited to small, generally horizontal growth forms such as *Arabidopsis thaliana*, or field-based systems or devices based around laser scanners. Such systems are generally restrictive in their accuracy and associated hardware and running costs.

The creation of 3D plant models is made increasingly challenging as plants are a sessile organism, acclimatising and evolving to their fluctuating environment, from short-term changes such as the reorganisation of foliage to long-term growth patterns. Due to their proficient dynamics and intricate phyllotaxis, as a consequence of their complex environment, plants are incredibly difficult to model (Ivanov *et al.*, 1995; Tan, Yuan and Wang, 2003; Reche-Martinez, Martin

and Drettakis, 2004). For example, a plant may consist of hundreds of leaves spanning arbitrary directions and angles. Consequently, even a small plant could require an unimaginably large number of polygons to define every minute facet digitally (Weber and Penn, 1995).

Existing image-based approaches can provide adequate results, but often rely on fixed cameras and intrusive or destructive techniques. *Intrusive* approaches involve moving the plant foliage to obtain further information. The *destructive* approach constitutes the removal of plant components, often requiring manual pruning of plants, increasing the length of the acquisition process and introducing the potential for irreversible error, for example, pruning too low, resulting in an inaccurate 3D representation of a plant. These approaches to plant reconstruction mean that it is impossible to conduct the same experiment on the same plant as it is no longer available, or the structure has changed from the original. A robust method to automatically acquire 2D images of an arbitrary plant and create a 3D model without human intervention would be a much-welcomed contribution to plant phenotyping and help reduce the associated bottleneck.

1.3 Research Aims

Aim

Identify computational techniques capable of recovering 3D models from multiple colour images of crowded scenes, with a focus on applications in plant phenotyping using a single RGB camera (which is a constraint of the project). While the aim is not to output phenotypic parameters, rather it is to generate a 3D model in which they can be extracted.

Objectives

1. Investigate active vision approaches with the goal of automatically capturing the best images, rather than selecting from a fixed, pre-existing set, to obtain a dense point cloud representation of some arbitrary object.
2. Identify a hybrid algorithm to automate 3D model reconstruction utilising available data in the form of 2D images, 3D point clouds, volumetric models and the environment, reducing the need for human intervention.

3. Evaluate reconstructed models on ground truth data, against existing 3D reconstruction algorithms and in the context of plant phenotyping.

Research questions

- I. Can we improve the process, and output, of 3D reconstruction by manipulating 2D images, 3D data and interacting with the scene to obtain knowledge without human intervention?
- II. Can we significantly decrease the point cloud data, reducing computational requirements, while maintaining an accurate representation of the structural parameters of the object during the surface reconstruction process?
- III. Can automated methods based on active image acquisition, surface reconstruction and mesh-repairing support robust and accurate plant phenotyping?

1.4 Contributions

We aim to address these issues by employing an active vision cell, a model merging process to combine volumetric and point cloud data, a top down clustering algorithm for tightly packed, crowded scenes and finally, a cell merging approach with level set methods for surface reconstruction.

The cell, consisting of a robot, camera and turntable, is automatically calibrated. A novel active vision approach is proposed to automate the process of capturing images from crowded scenes. A new approach to reduce and simplify point cloud data, removing noisy points and outliers is proposed which is clustered using a novel clustering method based on 1D data. Finally, a 3D surface reconstruction algorithm is proposed merging clusters based on connectivity information and using level sets to maintain boundaries.

1.5 Publications Arising from this Work

An overview of the current publications and work in progress are shown below;

Journal Articles:

Jonathon A. Gibbs, Michael Pound, Darren M. Wells, Erik Murchie, Andrew French, Tony Pridmore, (2016) **Approaches to three-dimensional**

reconstruction of plant shoot topology and geometry. *Functional Plant Biology* ISSN 1445-4408

Jonathon A. Gibbs, Michael Pound, Darren M. Wells, Erik Murchie, Andrew French, Tony Pridmore, (2018). **Plant Phenotyping: An Active Vision Cell for Three-Dimensional Plant Shoot Reconstruction.** *Plant Physiology*

Jonathon A. Gibbs, Michael Pound, Darren M. Wells, Erik Murchie, Andrew French, Tony Pridmore, (2018). **Active Vision and Surface Reconstruction for Crowded Scenes.** *IEEE/ACM Transactions on Computational Biology and Bioinformatics.*

Conference:

The British Machine Vision Association and Society for Pattern Recognition: Plants in Computer Vision (Update: Accepted 24th September 2016 takes place in November 2016) **A Fully Automated Active Vision Cell for 3D Reconstruction of Plant Shoots**

Jonathon A. Gibbs, Michael Pound, Darren M. Wells, Erik Murchie, Andrew French, Tony Pridmore, **Three-Dimensional Reconstruction of Plant Shoots from Multiple Images using an Active Vision System** In G. Kootstra, Y Edan, E van Henten, and M Bergerman (Eds.), *Proceedings of the IROS Workshop on Agri-Food Robotics*. Hamburg, October 2, 2015.

Demonstrations:

Digital Research Week

<https://www.nottingham.ac.uk/it-services/digital/dr-week.aspx>

Research Strategy: One Year On

<http://blogs.nottingham.ac.uk/researchexchange/2018/06/26/research-vision-celebrating-remarkable-year/>

1.6 Synopsis of the Thesis

The thesis is organised as follows; Chapter 2 introduces image processing and review of the current state-of-the-art and the challenges associated with this, moving on to a discussion of active vision and the next best view problem. Chapter 3 introduces the hardware setup and the software interface are discussed, explaining the process involved in the calibration and how the

approach was chosen. Chapter 4 discusses the active vision cell which is used to acquire data for the later stages. Chapter 5 relates to the data simplification, modelling merging and clustering of the data – a pre-processing step before reconstruction. Chapter 6 discusses the reconstruction method including level sets and cell merging.

Chapter 2

Background

This chapter reviews the state-of-the-art, putting the work here into context, starting with a broad discussion of 3D reconstruction. Plant modelling is then discussed, first highlighting the general difficulties associated with modelling crowded scenes, proceeding to an in-depth review of current methods of plant modelling. The second part of the chapter focuses on active vision, first discussing how calibration of numerous hardware components can be performed, before formulating the next best view problem and reviewing existing approaches. The chapter is concluded with a discussion and justification of the methods chosen in this work.

2.1 Introduction

In computer vision, 3D reconstruction is the process of generating a 3D model of an object from some data and representing it digitally as a series of triangles. The term 3D modelling refers to the process of acquiring data and using this in order to create a digital representation (Remondino and El-Hakim, 2006). 3D modelling has been applied to a wide range of scenarios from medical usage, e.g. creating 3D models of the brain using Magnetic Resonance Imaging, to the creation of environments for films and animations. The use of 3D models is present everywhere in modern times, becoming increasingly popular on the internet as modern, low-cost machines now have the capability of rendering these models. One recent example is the ability to render a 3D model of a face from a single image using deep learning (Jackson *et al.*, 2017).

With respect to plant phenotyping, biologists and computer scientists have been looking at how to represent the composition of plants as a 3D model. However, accurately representing such crowded scenes is complex and, despite the available resources, still requires considerable effort (Remondino and El-Hakim, 2006).

2.2 Plant Modelling

Phenotypes are analysed throughout the plants' lifecycle to determine performance and tolerance to different ecological and physical environmental conditions. Manual measures of phenotypic parameters are time consuming, inaccurate and often costly. Measuring the geometric structural parameters of plants is of utmost importance in understanding the underlying biological and physical processes, which is fundamental if crop yield is to be improved (Wang *et al.*, 2009).

Height, dimensions, leaf area index (LAI) and leaf angle distribution (LAD) are all important parameters that directly relate to the growth and light harvesting ability of plants and are obtainable measures from 3D plant models. It is therefore critical to construct precise 3D representations of plants to facilitate accurate assessments of physiology. Accurate 3D plant models can also support simulations and allow new traits to be identified, leading to comprehensive knowledge of plant structure and growth. Consequently, there is a need for a

robust and flexible image analysis tool for 3D plant reconstruction (Hartmann *et al.*, 2011).

Modelling a complex plant in 3D is a challenging problem (Seitz, 2000; Mulayim, Yilmaz and Atalay, 2003) and until the late 1960's botanical drawings were the primary means of representing plant architecture. The modelling of plants, digitally, dates back to the 1960s in which biologists wanted to investigate plant growth and developed a method for simulating plants (Lindenmayer, 1968). Today, with the use of high performance computers, many approaches to plant modelling exist, though there remain many challenges.

2.2.1 Plant Modelling Approaches

There are two broad categories of 3D plant modelling. These are known as *rule-based* approaches, which apply a series of mathematical rules to create a model and *image-based* approaches, which use actual data in the form of 2D images (Remondino and El-Hakim, 2006). Figure 2.1 illustrates the various modelling approaches and the corresponding category. With respect to plant phenotyping, 3D modelling seeks to recover quantitative information of plant traits from rules or image sensor data captured in laboratories, glasshouses, and field environments (Jin and Tang, 2009; White, Andrade-Sanchez, M. a. Gore, *et al.*, 2012).

Both categories are discussed next, providing an overview of the methods and the difficulties associated with them.

Rule-based methods

Rule-based methods can be used to simulate plant growth, creating synthetic plant structures that are particularly useful for Functional Structural Plant Models (FSPMs). FSPMs are used to study the ecophysiology, how plants sense and respond to environmental change, of a plant by combining the 3D representation with some physiological function (Vos *et al.*, 2010). Naturally, rule-based approaches require multiple parameters, specific to the plant being modelled, which determine its shape and structure.

A well adopted rule-based approach is the Lindenmayer System (L-System) introduced by biologist A. Lindenmayer in 1968 as a mathematical formalism for modelling multicellular organisms that form from linear or branch filaments,

focusing on the relationship between plant cells (Lindenmayer, 1968; Prusinkiewicz, 2003). This was later expanded to describe the complex branching structures of plants.

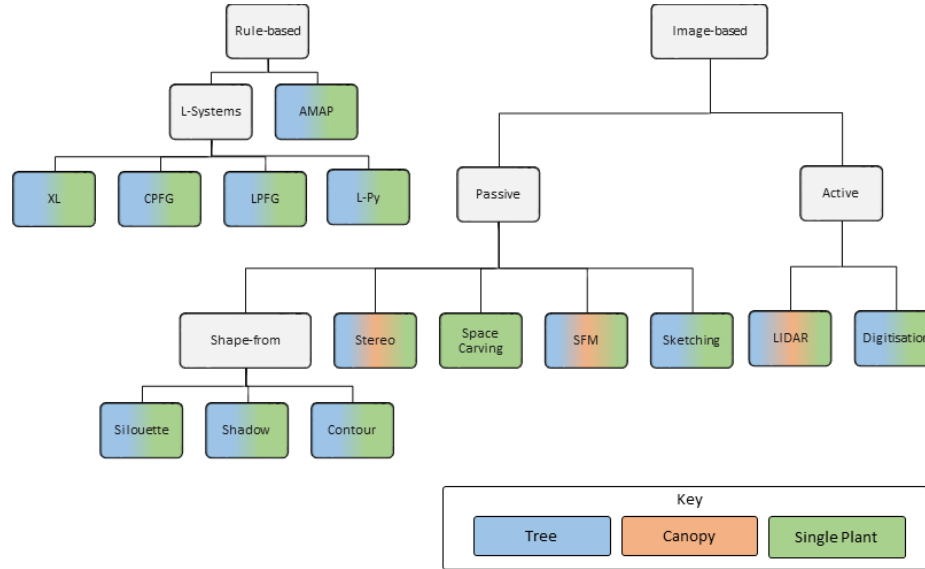


Figure 2.1 Three-dimensional modelling classification and uses for plant reconstruction shaded according to the key.

Over the past 20 years several implementations of the L-system have been designed, most notably the *plant and fractal generator with continuous parameters (cpfg)* (Prusinkiewicz, Hanan and Měch, 2000), *plant and fractal generator implementing the L+C language (lpfg)* (Karwowski and Prusinkiewicz, 2003), and *eXtended L-Systems (XL)* (Ole and Winfried, 2008).

More recently Boudon *et al.* (2012) developed the L-Py paradigm based on the classical L-system model using a dynamic language approach, Python. They apply an adaption of dynamic languages to the modelling of plant growth to overcome the limitations associated with statically typed languages that are unable to cope with changing or unknown conditions. In the last decade L-systems have emerged as a major paradigm for the modelling of synthetic plants.

Difficulties of rule-based methods

Rule-based methods have complex input parameters that require expertise or plant dependent information, making the task cumbersome. The construction of 3D plants requires knowledge of plant topology and multiple, intensive measurements of architectural features (i.e. leaf and stem length, leaf angle distributions, etc.). This means that rule-based approaches tend to be both time-

and parameter-intensive. Too few parameters are inadequate to accurately define plant characteristics, whilst large numbers of parameters tend to prove difficult to manage, typically being tedious to manipulate and comprehend (Boudon *et al.*, 2003). As such, few models are available for a select number of crops (e.g. Fournier *et al.* 2002; Evers *et al.* 2005; Song *et al.* 2013)

Rule-based methods are limited in their ability to portray a real-world plant, making it impossible to calculate geometric properties of an actual plant. This is because models are produced on averages of a number of parameters and therefore may not necessarily reflect inter- or intra-field variation. For example, the use of averaged measurements can result in large disparities when compared to models containing explicitly described leaf angles (Fourcaud *et al.*, 2007; Vos *et al.*, 2010; DeJong *et al.*, 2011; Sarlikioti, de Visser and Marcelis, 2011). This could mean that unique features of the canopy structure, which could determine light interception properties of the stand, are missed (Burgess *et al.*, no date; Sonohat *et al.*, 2002). Furthermore, the rules are species or variety specific and for that reason modelling a new plant would require additional variables resulting in the need for botanical expertise for each model, consequently increasing the complexity of development. Many approaches to reduce complexity have been proposed (Lintermann and Deussen, 1996; Deussen and Lintermann, 1997; ShlyakhterI. *et al.*, 2001; Boudon *et al.*, 2003, 2012), though the underlying complexity of model development remains for the application developer, who requires a certain degree of botanical awareness. Furthermore, in some studies real, rather than synthetic, plant measurements are required.

Image-based methods

Image-based methods differ from rule-based methods by extracting the geometry and trait data directly from the scene without the need for complex parameters. These approaches reduce the complexity of plant modelling, in comparison to rule-based methods, and delineate real world plants, overcoming the problem of manually specifying parameters by using a set of images (Neubert, Franken and Deussen, 2007).

Image-based plant modelling techniques are highly desirable as a method of plant phenotyping; however, the complexity of plant form means that using image-based approaches can be challenging. Similarities between multiple small leaf segments, lack of texture for feature matching and the high amount

of self-occlusion can lead to difficulties during model production (Pound *et al.*, 2014). Different image-based techniques result in different models; for example, the production of a point cloud cannot be used for modelling photosynthesis. When designing a phenotyping system, numerous aspects must be considered including the type of model required, the throughput; and accessibility to technology and resources. The level of detail is often constrained by the computational capabilities available and the method used to model the plant.

Image-based methods are categorised as either *active*, in which some form of controlled radiation is projected onto the plant, or *passive*, in which only natural illumination is used. Active methods are significantly more expensive and require specialist hardware, e.g. to project lasers into the scene. Passive techniques are typically portable and low cost, recording data using radiation already present in the scene. Light Detection and Ranging (LiDAR), also known as Laser Scanning, can be classified as active. Space Carving, Shape-from-Silhouette (SFS), Shape-from-Shading (Cryer and Shah, 1999), Shape-from-Contour, Stereo vision and Structure from Motion (SFM) are passive approaches, commonly using standard hand-held cameras to acquire data. The use of cameras can be a benefit as they are easily portable and can be used within the field or other inaccessible areas where more bulky or expensive equipment would not be feasible. These approaches are discussed in more detail in the following sections so will not be discussed here. Approaches such as Shape-from-Shading (Horn and Brooks, 1989), Shape-from-Texture (Kender, 1981) and Shape-from-Edges (Wahl, 2001) are uncommon in plant modelling due to the complexity of the object and the reliance on a single image, making them more susceptible to occlusion, a common occurrence in plant structures.

Image-based methods have an elusive goal of achieving photorealism, that is, reproducing some object as realistically as possible rather than producing a synthetic representation (Weber and Penn, 1995). Some authors feel image-based approaches offer the best solution to reconstruction, due to a simplistic acquisition process and reduced need for botanical expertise (Tan *et al.*, 2007).

Difficulties of image-based methods

Numerous difficulties need to be considered when using an image-based approach to 3D modelling. Determining the number of samples, i.e. images, to use to construct a 3D model, is often problematic. Sampling analysis is the

process of identifying the relationship between depth, number of samples, and the resulting resolution in order to obtain an optimal number of samples. Collecting excess samples is known as *oversampling* and amounts to a more intensive data acquisition model, higher capacity requirements and greater redundancy, consequently increasing computation time and the complexity of the application (Shum and Kang, 2000). Oversampling is often the result of trying to counter undesirable aliasing effects in output. Incomplete and inaccurate reconstruction is a classic consequence of *undersampling*, the process of acquiring an inadequate number of samples such that data is missing.

Overlapping areas of an object, for example leaves that occlude each other, is another reason that 3D information may not be available. A larger collection of samples can help alleviate this problem by capturing views from more angles, though sometimes a more complex approach is required. Various approaches to detecting and dealing with occlusions have been proposed by Brown *et al.* (2003), though in some cases approximation techniques, such as estimating the missing information intelligently, must be employed. Image acquisition strategies have the potential to reduce the complexity of image-based modelling by capturing images at defined intervals. For example, solving Next Best View (NBV) for some given object determines which views need to be captured to sample all visible surfaces. Though the quality of output depends heavily on the evaluation function driving the NBV algorithm.

Other approaches to plant modelling use *intrusive* or *destructive* approaches to acquire thorough information of the plant. This requires moving the plant, or the removal of plant parts, making it impossible to repeat the experiment as the plant is no longer available or has moved from its original position. Destructive approaches often require manual pruning, adding additional time to the acquisition process and increasing the potential for irreversible error. For example, pruning too low results in an incomplete acquisition process.

As rule-based approaches are unable to delineate real world plants they are not considered a suitable methodology for this research. Consequently, an image-based approach is necessary to accurately acquire plant traits and the remainder of the chapter considers only image-based methods.

2.2.2 Semi-Automated Approaches

Despite advances in technology and resources, current automated approaches for 3D reconstruction are cumbersome. Few fully automated approaches, those capable of capturing data, performing the intermediate steps, such as the removal of noisy points, and producing an output as a 3D model, have been proposed. Many image-based approaches (Section 2.3) require user input, most commonly during the data acquisition stage.

Although semi-automatic approaches provide adequate results, it is still vital that fully automated systems are introduced to overcome the bottleneck associated with phenotyping by rapidly producing 3D models. It is important to note that in this work a fully automated system is defined as one that can acquire images of some object and reconstruct a 3D model without user input. Previous work refers to automated systems as those that can simply generate 3D models, though the time and required accuracy of the preliminary data acquisition stage, which is performed manually, is often ignored, or oversimplified. These methods are therefore referred to as semi-automated approaches.

With these common modelling difficulties in mind, and the knowledge that plants are incredibly complex plus more irregular and variable in shape than synthetic structures, it is clear plants are testing objects to model (Ivanov *et al.*, 1995; Tan, Yuan and Wang, 2003; Reche-Martinez, Martin and Drettakis, 2004; Boudon *et al.*, 2012). The difficulties that have been identified need to be addressed to evolve and exploit current advances in technology. In the next section the approaches to 3D modelling and how these methods have tried to overcome the difficulties discussed here are reviewed.

2.3 Image-based Plant Modelling

The 3D modelling of plants has been achieved using a range of approaches and applied to a variety of plant species from small, sparse, indoor plants to dense forestry, with a common goal of understanding and analysing plant structure and function. The understanding of plant architecture and physiology is significantly important to determine desirable conditions and improve production.

The next section reviews the state-of-the art for active and passive image-based approaches to plant reconstruction, all of which are semi-automated approaches. The advantages and disadvantages present in these methods form the basis for

the current work, with features being exploited to create a fully automated system, overcoming some of the most challenging aspects of 3D reconstruction.

2.3.1 Active Methods

Active reconstruction methods, not be confused with active vision, discussed in section 2.4, is the more expensive of the two approaches, actively interfering with the scene either mechanically or using radiation, requiring expensive hardware capable of emitting radiation into the scene and recording the feedback. Figure 2.2 illustrates the process associated with active approaches; here a laser is projected onto the object and the reflected light is measured.

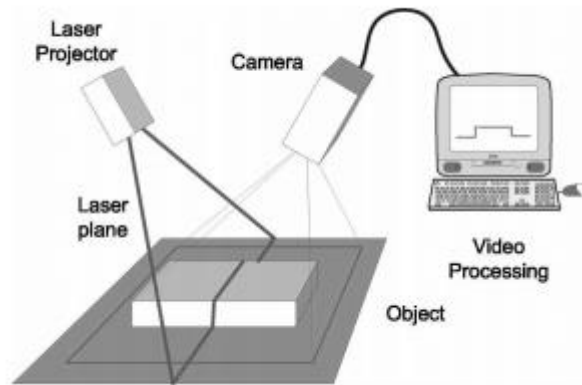


Figure 2.2 An example of a laser scanning setup where radiation is emitted into the scene and the feedback is recoded by the camera (Zolfaghari and Khalili, 2008).

Light Detection and Ranging (LiDAR), originating in 1963, is a remote sensing technology based on the extension of principles in radar technology, that is, detecting electromagnetic energy reflected from a target point. LiDAR measures the distance between itself, the scanner, and the object by illuminating the object with a laser and analysing the reflected light (Northend, 1967). Figure 2.3 illustrates a successful reconstruction of a 3D plant using a LiDAR approach in a controlled environment (Sanz-Cortiella *et al.*, 2011).

Active reconstruction methods using lasers have been commonly used to reconstruct trees. Trees are particularly imperative due to their functional roles in the environment and, to understand the connection between function and structure, extensive investigations are required. Trees have been studied in great depth, contributing a significant amount to the area of plant modelling (Weber

and Penn, 1995; Sinoquet and Rivet, 1997; Sakaguchi, 1998; ShlyakhterI. *et al.*, 2001; Boudon *et al.*, 2003; Reche-Martinez, Martin and Drettakis, 2004; Phattaralerphong and Sinoquet, 2005; Hosoi and Omasa, 2006; Rutledge and Popescu, 2006; Tan *et al.*, 2007; Neubert, Franken and Deussen, 2007; Livny *et al.*, 2010; Sanz-Cortiella *et al.*, 2011; Tang, Dong and Buckles, 2013).

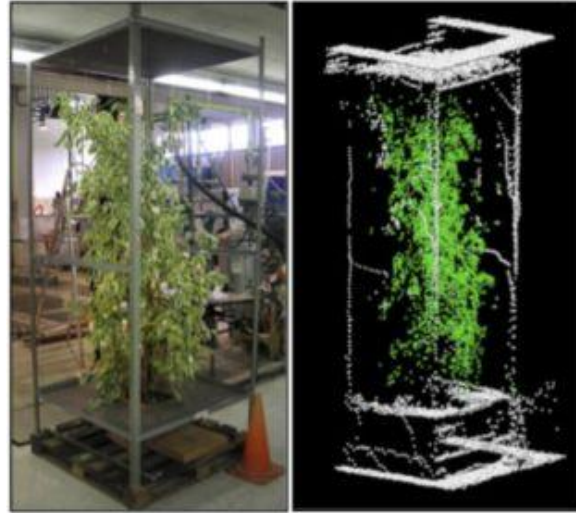


Figure 2.3 Plant reconstruction using laser scanning of *Ficus benjamina*) conducted by (Sanz-Cortiella *et al.*, 2011) in a controlled environment.

With recent advances in technology LiDAR is becoming more commonly used and can be conducted as both airborne and ground-based. Airborne LiDAR is important in tree reconstruction, as it can estimate forest attributes, such as height, diameter and canopy closure, all of which are an essential part of forest management (Hosoi and Omasa, 2006; Omasa, Hosoi and Konishi, 2007; Preuksakarn *et al.*, 2010; Van Leeuwen, Coops and Wulder, 2010; Tang, Dong and Buckles, 2013). Ground based methods are often limited in their capacity to retrieve individual tree and crown attributes due to occlusion, gaps or inaccessibility.

LiDAR based methods reconstruct surfaces of tree canopies from point clouds. For example, Tang, Dong and Buckles (2013) use raw data from the LiDAR point cloud and region-based level set methods to model canopies. Alternatively, tree skeletons can be obtained and provide vital information regarding the structure of trees. For example, Livny *et al.* (2010) create a tree skeleton from laser scans captured using a moving vehicle. A series of global optimisations, making it robust to noisy and incomplete data, are employed to consolidate a point cloud representing one or more tree objects as skeletal

structures. This optimisation aims to reconstruct the major branches of the captured tree(s), resulting in a graph structure known as the Branch-Structure-Graph (BSG), the finer branching structures are then reconstructed from the BSGs, which assume that the finer parts of the tree structure are made up of the same branching structure as the core of the tree.

A method known as Canopy Height Models (CHMs) is used to represent horizontal and vertical properties of tree canopies, measuring how far above the ground the tree crown is. The recovery of tree crown characteristics using CHM is incredibly beneficial. However, the use of CHM is challenging, and a number of difficulties have been identified. These include canopy gaps, underestimation of tree height, surface irregularities, inability to accurately retrieve individual trees and erroneous point clouds due to moving points, a consequence of being in a natural environment, all of which result in a reduction in the quality of the plant model.

Parametric Height Models are an airborne solution to CHM's difficulties, proposed by Van Leeuwen, Coops and Wulder (2010). They are used to overcome the problem of underestimating tree height in CHMs by describing the forest canopy as a series of cones fitted to the raw LiDAR point cloud, then, by applying simple geometric operations, the height can be adjusted and corrected.

Occlusion is a major problem in LiDAR approaches. Preuksakarn *et al.* (2010) use an approach influenced by the space colonisation algorithm which improves the robustness of the reconstruction and reduces issues associated with occlusion. This is achieved by shifting points from the border towards its medial axis using an adaptive contraction scheme. The result is a skeleton describing the branching structure of the tree. 3D modelling is performed by projecting the attractors, from the skeleton reconstruction, onto the plane perpendicular to the growth direction, estimating the occluded areas.

Terrestrial Laser Scanning (TLS), a form of LiDAR, is used by Bucksch and Lindenbergh (2008) who define a method for generating point clouds to obtain skeletons from natural trees using the CAMPINO algorithm – a skeletonisation algorithm - which takes a point cloud as input and generates a tree skeleton. They claim to have satisfactory results, though they found it difficult to capture all the data under unconstrained lighting conditions, as the daylight interferes with the reflection back to the scanner.

More general disadvantages of LiDAR include the initial cost of hardware which often is not affordable, and natural lighting which makes it difficult to capture data as the sun interferes with the reflection back to the scanner. Some LiDAR systems are affected by metal structures and magnetic sources, and, though this is not a problem for most rural areas, it makes it difficult in urban environments. Jin and Tang (2009) found that during experiments in natural conditions the acquisition of images under direct sunlight turned out to be severely saturated when compared with those taken under cloudy lighting conditions.

Active reconstruction techniques have also been applied to smaller plants. For example, Watanabe *et al.* (2005) model rice (*Oryza sativa*) plants using a cpfg approach, using a 3D sonic digitiser to capture the initial point cloud. The digitisation process is reported to take up to an hour to complete and Rakocovic, 2000 claim that the digitisation process for their approach to reconstruct White Clover (*Trifolium repens*) canopies required between 3 to 7 hours. They used electromagnetic digitising apparatus with corner flags to aid calibration, applying a destructive approach by pruning the canopy from the top downwards.

Paproki *et al.* (2012) use a mesh segmentation algorithm which partitions the plant, in this case cotton (*Gossypium hirsutum*) into morphological regions. It does so in four steps, namely; a coarse segmentation, stem segmentation, petioles segmentation and finally leaf segmentation. However, the segmentation process requires manual input which has the potential to be error prone and adds additional processing time to an already lengthy process.

A comprehensive review of 3D modelling using LIDAR sensors from the 1980s to early 2000 has been conducted (Blais, 2004). They conclude that the object shape alone is not enough to obtain optimal precision and that some form of 2D texture is also required to improve the accuracy of this approach, which remains the case today.

2.3.2 Passive Methods

Passive approaches to 3D reconstruction are gaining an increasing amount of popularity as they only require a standard off-the-shelf handheld camera to capture overlapped images of a scene using the radiation present, and a basic machine to process them. However, as the 3D data is generated from the 2D images, the 3D model can only ever be as good as the image set available.

Consequently, it is of utmost importance that the image data is of the highest quality and that the optimal amount of data from the plant is captured. Approaches used in the 3D modelling of plants using passive methods are now discussed.

Shape from Silhouette

Shape-from-silhouette (SFS) is 3D modelling algorithm introduced by A. Laurentini (1994) which assumes the background and foreground can be separated resulting in a silhouette of the object. SFS aims to reconstruct a 3D model by projecting the silhouette of multiple images. The intersecting projections are then used to produce the 3D model where a back-projected cone is defined containing the object from two or more 2D images. The intersection of two cones is known as a visual hull, which represents a 3D model of the projected images. Figure 2.4 illustrates a projection of two images to produce the visual hull.

Despite the advantages that volume intersection offers, such as ease of implementation and the ability to deal with difficult topology, it struggles to reconstruct concavities, is unable to produce photo-consistent scenes (a photo consistent voxel is one that has a similar colour across all images that can see it) and, in some cases, struggles to find the initial silhouette. Moreover, visual hulls produce the maximum shape that gives the same silhouette as some given images, therefore, excess volume is often contained in the visual hull approximation due to concavities. Consequently, SFS commonly has to be augmented with approaches that are capable of removing these excess voxels (Mulayim, Yilmaz and Atalay, 2003). For example, ShlyakhterI. *et al.*, 2001 use SFS to generate the axiom for an L-System and Reche-Martinez, Martin and Drettakis, 2004 aims to combine volumetric opacity estimates with view-dependent texturing to reconstruct trees from images.

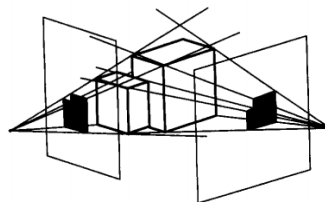


Figure 2.4 During the shape-from-silhouette algorithm, the visual hull is the result of two or more 2D silhouettes, from images, being projected. The resulting object is a 3D representation of the object (Laurentini, 1994).

Space Carving

Space carving introduced by Seitz (2000), is a form of SFS, differing in the fact in that it formulates the shape recovery as a constraint satisfaction problem. Space carving uses a bounding box to encapsulate the object ensuring the 3D object is fully enclosed (e.g. Figure 2.5). The box is partitioned into a series of voxels. The algorithm iterates through each voxel of the bounding box, removing (carving) those voxels that are empty – a non-photo consistent voxel. Each time a voxel is carved away it potentially uncovers a new voxel from a different image which requires evaluation for photo-consistency. The process continues until all visible empty voxels are removed.

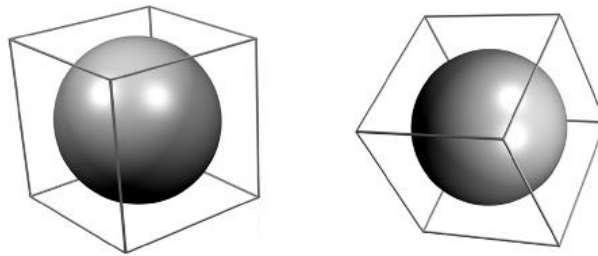


Figure 2.5 During space carving, a bounding box encapsulates the object ensuring the furthest points are contained within the box.

Figure 2.6 illustrates the space carving algorithm, where *A* is the initial bounding box divided into a series of uniform voxels, *B* is the start of the process of removing empty voxels and *C* is the result after all visible empty voxels have been removed. Figure 2.6(c) illustrates the difficulties with occlusion and depth that the space carving suffers from. To overcome this in simple structures, an adequate number of images need to be taken to ensure that all surfaces of the object have been captured, though as the object becomes more complex this is no longer a trivial task.

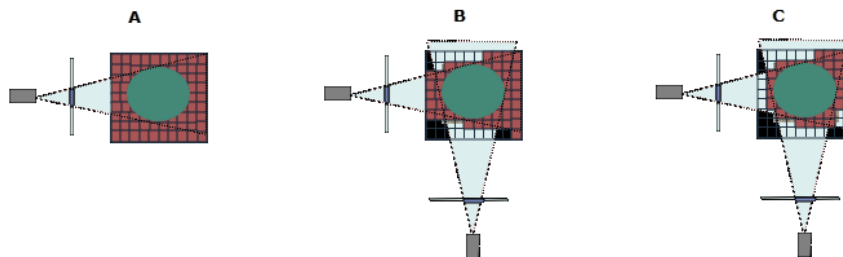


Figure 2.6 An example of Space carving, removing voxels to create the 3D object. A. the initial bounding box, B. space carving being applied using first

view of the object, C. the result of the finished space carving product from two views with occlusions and empty voxels still remaining (ETH Zurich).

With respect to plant phenotyping, Neubert, Franken and Deussen (2007) use a space carving approach with particle flows to estimate tree volume. They obtain a set of images, separating the background from foreground, then create a voxel grid by back-projecting from the input images' silhouettes, where each voxel contains an estimation of tree density. Directional fields for each 2D image are incorporated to control the flow of particles. During the particle simulation the flow is influenced by the density and defined directional fields moving them away or towards edges, producing an approximation of the tree volume. A method using directional force fields may converge to produce good results. However, it is shown in (Chapter 5) that this is not feasible for small objects.

Sketching

Manual sketching is sparsely used as a means for plant phenotyping due to the vast computing power and alternative resources available at easily affordable prices. However, it has the advantage that freehand drawing enables any shape to be captured and contours are easily identifiable (Anastacio *et al.*, 2006). Most sketching methods use an interface to enable direct manipulation of the plant simulation, allowing even novice users to create plant structures (Masry and Lipson, 2007).

Sketching is often laborious and does not explicitly represent a specific plant, rather a synthetic representation of a plant is modelled; like rule-based methods. Unfortunately, the disadvantages far outweigh the advantages, for example, once a sketch has been started the viewpoint is fixed, it is not possible to rotate around the object. Moreover, the sketch is final and cannot be changed and one of the biggest drawbacks is that each sketch must be recreated from scratch, resulting in a lengthy process.

A recent non-plant specific survey on the current state-of-the-art has been conducted by Olsen *et al.* (2009) who conclude that sketch-based systems are improving and that a system to store shapes and even create new shapes is a potential future development which will significantly improve the current available methods. However, it is unlikely to ever become a major paradigm to plant modelling.

Stereo vision

Computational stereo (Stereopsis), also known as stereo vision, is the process of determining 3D data from two or more images taken from distinct viewpoints, with the basis that a single 3D point projects to a pair of unique 2D image locations. Stereo vision is a method based on the functionality of the human eyes – the ability to see the same scene but at slightly different angles, achieved by distinct lateral positioning, known as binocular vision. The human brain receives two separate images and processes them, converting them into a single 3D representation of the viewed world. Stereo vision mimics this process, extracting 3D information by processing two 2D images captured simultaneously from slightly different horizontal viewpoints (Figure 2.7). Depth information is extracted by correspondence matching, matching points between two images (Dhond, U. R., Aggarwal, 1989). The disparity in a point's position between the left and right view is used to create a disparity map, which is directly related to depth and supports the construction of a 3D model.

Stereo vision has been an intense area of research for decades, with early work conducted in the 1970s, providing a fundamental problem in Computer Vision due to the complexities of shapes and scenes (Brown *et al.*, 2003; Biskup *et al.*, 2007). A review of early stereo vision dating back to the 1970's and 80's has been conducted by Barnard and Fischler, 1982 and Dhond, U. R., Aggarwal, 1989 respectively while Brown *et al.*, 2003 provide a more recent and comprehensive review, concluding that a thorough understanding of stereo vision has led to more real-time systems and accurate correspondence matching algorithms.

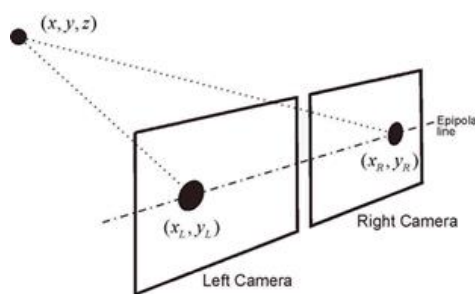


Figure 2.7 Stereo vision is the process of capturing images simultaneously of some object where each view differs slightly (G Calin and Roda, 2007).

Over the past 30 years as the field has evolved and gained an increased amount of attention; however, the goal of mimicking and understanding human vision has largely been abandoned. Current work leans more towards the general problem of reconstructing a 3D model, with less emphasis on biological vision.

Despite the increased awareness and understanding, challenges remain and arise from the start with the need for **calibration** and **correspondence**, discussed below. Furthermore, as the computational power of machines develops there is an increased demand for realism in reconstructions, resulting in denser output requirements to support sophisticated geometric shapes and physical objects (Biskup *et al.*, 2007).

Stereo calibration is a prerequisite for 3D reconstruction of this type; it is the process of finding the intrinsic parameters (focal length, principal point, radial and tangential distortion) of the cameras and the extrinsic parameters (rotation matrix and translation vector) representing the world coordinates as camera coordinates. It can be performed using a number of techniques; one particular open source method is VisualSFM, which performs automated camera calibration on a set of input images. It uses the scale-invariant feature transform (SIFT) (Lowe, 1999), a feature detection algorithm, to find corresponding points which are used to calculate the 3D position of the cameras. Other approaches include the use of planar targets or 3D objects. A more in-depth look into algorithms for camera calibration is found in section 2.4.1.1.

Correspondence needs to be established between features from two images that correspond to some physical features in space (Dhond, U. R., Aggarwal, 1989). *Correspondence* refers to the problem of determining the locations of pixels in a pair of 2D images that are the projection of the same physical point in space. Figure 2.8 depicts point X as the point being observed by two distinct cameras, C and C' . x and x' represent the projection of X into the left right image planes respectively. The resulting displacement of a projected point in one image with regards to another is termed disparity. A set of disparities is known as the disparity map. Features visible in one image but not another are said to be *occluded* and as a result cannot be included in the disparity map (Brown *et al.*, 2003).

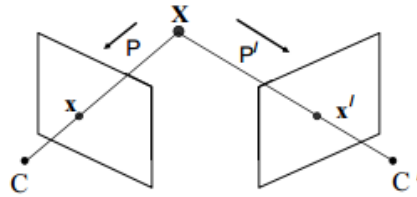


Figure 2.8 Correspondence matching between one feature in two images taken at varying viewpoints.

One concept used to obtain the associations between two images, assisting the correspondence process, is *epipolar geometry*. Epipolar geometry is the geometry of stereo vision, specifying the geometric relations between 3D points and their corresponding 2D points. The *epipolar points* are the points of intersection of the line joining the optical centres, the baseline (C to C'), with the image planes - e and e' , Figure 2.9. The *epipolar line*, is defined such that X is seen by the left camera, C , as a single point but seen as a line in the image plane of the right camera. The process is symmetrical for C' . The *epipolar constraint* reduces what appears to be a 2D search problem, from matching a point in one image with that in another, to a 1D search through the use of the *epipolar line* (Figure 2.9) matching can now take place on the line segment xe and $e'x'$ for the left and right image respectively.

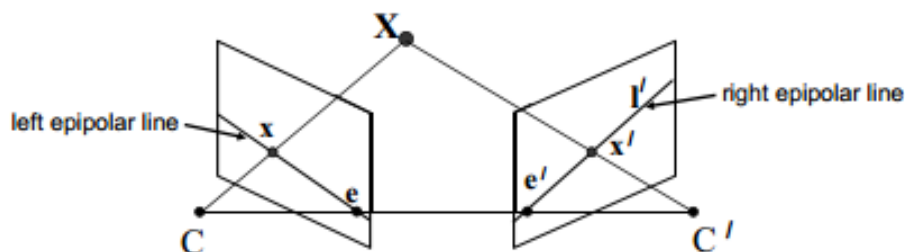


Figure 2.9 The Epipolar constraint is used to aid correspondence matching by reducing a two-dimensional problem to a one-dimensional search using the epipolar line.

Several other approaches have been proposed to solve the problem of correspondence. Some simple methods use the measure of absolute or sum of squared differences (SSD) of the pixels intensities, which measures the similarity between image patches (Andersen, Reng and Kirk, 2005). Other approaches employ window based matching (G. Calin and Roda, 2007), dynamic programming, simulated annealing (Andersen, Reng and Kirk, 2005), or place markers in the scene (Biskup *et al.*, 2007); (Ivanov *et al.*, 1995) for example use parallelepipeds as markers. These algorithms typically fall into *local methods* or *global methods*. (Brown *et al.*, 2003) provide a comprehensive review of the varying approaches and where they have been applied.

Andersen, Reng and Kirk (2005) propose an image-based approach which uses a relaxation stereo vision model with simulated annealing (SA) to aid the matching process, finding a global optimum instead of local, as a result enhancing the matching process. Furukawa and Ponce (2010) propose an

algorithm for calibrated multi-view stereopsis known as Patch-based Multi-view Stereo (PMVS). PMVS generates an initial point cloud that is a suitable basis for further modelling techniques. The algorithm is designed to work on objects, scenes and crowded scenes and produces relatively good results on the datasets used. The reconstruction is based on a set of colour images and requires that the camera is calibrated; that is the intrinsic and extrinsic camera parameters are known.

There are numerous difficulties with stereo vision, for example; the detection of curved and smooth surfaces, known as the false boundary problem, is a key problem in stereo vision. Based on the angle of the camera, a false boundary appears, that is the furthest visible edge to the camera from one viewpoint that may not necessarily be the last point of some object. Figure 2.10 depicts a simplistic example of the false boundary problem, where the grey area represents what may be mistaken for the object, a false boundary.

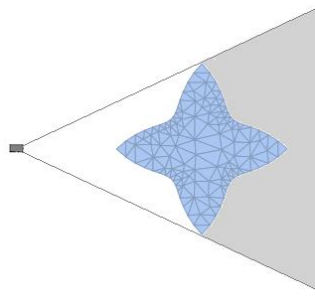


Figure 2.10 The false boundary problem; the object appears to be much longer and wider due to a lack of views resulting in a false boundary.

Two solutions have been proposed to resolve this issue. One is to place the cameras such that they intercept at the line perpendicular to the line between the two distinct boundaries; where these boundaries are the most bending curve of the surface. The second is to ensure that the camera is placed at a suitable distance from the object in order to capture the entire front facing surface of the object (Zheng, 1995). Alternatively, next best view is a suitable solution for this problem and is discussed in more detail in Section 2.4.2.

Stereo vision overcomes the need for an initially expensive setup and a large amount of literature has reported accurate plant reconstruction with commercially available hardware. However, many of these approaches still depend heavily on user input for segmentation or have some pre-defined constraints. For example, Biskup *et al.*, (2007) reconstruct small- to medium-

sized canopies using commercially available hardware down to single leaf level. They use a hue-saturation-value (HSV) colour space to overcome the traditional difficulties associated with a red, green and blue (RGB) colour space and apply a threshold to remove all but green plant pixels. They produce good plant models as a result, but the removal of blue and red makes this approach redundant for multi-coloured plants.

For those experiments conducted in natural conditions, scene radiation also causes problems. Jin and Tang (2009) propose a semi-automated stereo vision approach with a skeletonisation algorithm to identify individual corn plants and highlight leaves from stems and the centre of the plant. Grey-scale images were acquired using a real-time PCI stereo vision system in which they accurately identified 96.7% of the plants under the optimal lighting conditions. However, they report that the intensity of images of corn plants taken under direct sunlight turned out to be heavily saturated when compared with images taken under cloudy lighting conditions.

Structure from motion

Structure from Motion (SFM) follows the same broad goals as stereo vision, which are the recovery of 3D structure, motion recovery and correspondence. The differences between Stereo vision and SFM lie within the acquisition process. Where Stereo vision captures two images simultaneously, structure from motion captures them sequentially.

Pound *et al.* (2014) introduce an algorithm, *canopy reconstruction*, to produce 3D models of plant shoots from multiple images using structure from motion. The images are captured manually, and cameras calibrated by fitting a calibration target to the workspace, ensuring it is visible in each of the images. Modelling is performed by applying a simple clustering method, nearest neighbour, and performing 2D triangulation on each cluster. Level sets are applied to expand or shrink boundaries.

The work of Pound *et al.* is of importance in this work as it is used for evaluation of the algorithm proposed here. *Canopy reconstruction* produces good results, but numerous constraints and difficulties can be highlighted. The current data acquisition process requires that images are captured manually using a handheld camera. Not only is this time consuming it is to be expected to produce a less desirable model as the quality of the model is only ever as good as the images

captured. Input is limited to green plants as other colours are filtered out and a large number of manually captured input images are required to produce an accurate representation. Moreover, the manual use of multiple software environments is also a requirement, increasing the length of the process and manual input requiring expertise in each.

As the 3D models depend heavily upon the images they are constructed from it is crucial that an accurate and repeatable image acquisition process is available. Active vision offers a significant amount of flexibility and accuracy, eliminating the manual process. This is reviewed next.

2.4 Active Vision

Unlike static systems, an active vision (AV) system has the capability of interacting with the environment and adapting based on the information available in the scene. AV has a resemblance to human vision in that viewpoint can move in response to some data (Rao and Ballard, 1995). AV has been an area of research since the late 1980s' (Aloimonos, Weiss and Bandyopadhyay, 1988), with more recent work looking at manipulating the viewpoint in order to obtain more accurate results, for example, (Denzler, Zobel and Niemann, 2003) look at actively manipulating the focal length in order to acquire more precise information.

AV can be applied to numerous setups, for example attaching a camera to a car to aid navigation, which has been performed both on model and real cars, such as the current self-driving cars. Alternatively, as in the work conducted here, cameras can be attached to the end effector of a robot arm to acquire more knowledge of the environment and make informed decisions based on what can be seen.

AV has many advantages over traditional manual methods, particularly with the ability to overcome those challenges associated with 3D modelling; one of the most challenging being occlusions. An AV system has the potential to overcome this in most instances by manipulating the viewpoint such that it evades the part of the object causing the occlusion. Other advantages include the ability to overcome the limited field of view and develop image acquisition strategies with guaranteed accuracy and repeatability, which would not necessarily be possible with a manual approach. Moreover, manual methods require that some calibration target is visible in all images to calculate the required camera

matrices for 3D modelling. For crowded scenes it is often problematic to ensure the calibration target is visible. AV overcomes this by estimating and then maintaining the required camera matrices, discussed in the remainder of this section.

2.4.1 Calibration

To enable the manipulation of a viewpoint, it must be accurately calibrated; that is, the location of the camera with respect to the world must be known. This can be challenging and, depending on the system used, any number of calibrations may be required. For this work the calibration of the camera and robot with respect to the world, base and end effector needs to be performed. Robot calibration from the base to the end effector is known as forward kinematics and is easily obtainable from the robot joint angles. Camera calibration is performed by estimating the extrinsic parameters, the rotation and translation of the camera, and proves to be more difficult. Once these are known it is possible to calculate the position of the robot with respect to the world and the camera with respect to the end effector.

This section is divided as follows; the calibration of the camera is discussed next, moving on to discuss how the forward kinematics can be calculated, and finally a review of methods to calibrate the two remaining unknowns; the end effector to camera and world to robot base. Each of these calibrations results in a transformation matrix which maps the points of one space to another.

2.4.1.1 Camera Calibration

Camera calibration is the process of estimating the parameters of a pinhole camera model. It is required in order to map some 3D homogenous point $[x, y, z, 1]^T$ to some 2D image point $[u, v, 1]^T$ (Figure 2.11) where a 2D point m is formed by an optical ray passing through the optical centre, C , intersecting the image plane.

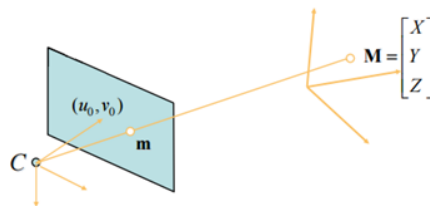


Figure 2.11 Pinhole Camera Model; the projection of a three-dimensional point onto a two-dimensional plane (Zhang, 2003).

To achieve this mapping between two- and three-dimensional points the internal geometry, intrinsic ($\mathbf{K}$), and external geometry, extrinsic $\mathbf{E} = [\mathbf{R}|\vec{t}]$, of the camera need to be estimated. The intrinsic matrix, (Eq. 2.1), contains 5 parameters representing the focal length α_x, α_y , the skew coefficient, γ , between the x and y axis and the principal point, ideally the centre of the image, u_0, v_0 .

$$\mathbf{K} = \begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. 2.1})$$

The extrinsic matrix denotes the coordinate system mapping from 3D world coordinates to 2D camera coordinates where $\mathbf{R}$ is the rotation and $\vec{t}$ is the translation, the position of the origin of the world coordinate system expressed as the coordinates of the camera centred coordinate system, this is represented in (Eq. 2.2)

$$\mathbf{E} = [\mathbf{R}|\vec{t}] = \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_1 \\ R_{21} & R_{22} & R_{23} & t_2 \\ R_{31} & R_{32} & R_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{Eq. 2.2})$$

The full equation mapping some 3D homogenous point to a 2D point is shown in (Eq. 2.3)

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{KE} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (\text{Eq. 2.3})$$

Camera calibration techniques can be roughly classified into two categories: self-calibration and photogrammetric. Self-calibration requires that three images are taken of some static scene, without the use of a calibration target, and that the internal parameters of the camera are constant. The correspondence between these images is sufficient to retrieve the intrinsic and extrinsic camera parameters (Zhang, 2004). This is still an emerging area of research and does not often provide as accurate results as photogrammetric methods.

Photogrammetry is the process used to obtain measurements of physical objects using 2D images. The most common photogrammetric methods include planar

targets which are easily augmented in the robot workspace. These are often printed and attached within the workspace, for example checkerboards in which calibration aims to identify each of the corners (Zhang, 2000); *augmented reality* (AR) markers (Baratoff, Neubeck and Regenbrecht, 2002), which enable some form of code or text to be assigned or *selection AR* which uses AR with a specific distance between each of the markers. The other photogrammetric approaches are based on either a 3D object whose geometry in space is already known, or 1D line-based calibration which consists of a set of collinear points. One particular approach uses a vanishing line to calculate camera parameters (Wang and Tsai, 1991).

2.4.1.2 Kinematics

Kinematics refers to the mechanics of the robot. It is used to describe the motion of joints without considering the cause of movement. Kinematics uses two different spaces, namely Joint space and Quaternion space, in which the former is a coordinate system that specifies each point as a pair of numerical coordinates and the latter is a complex number system used in 3D space (Bingul, 2006). The aim of kinematics is to provide a transformation between these two spaces known as inverse and forward kinematics (Figure 2.12).

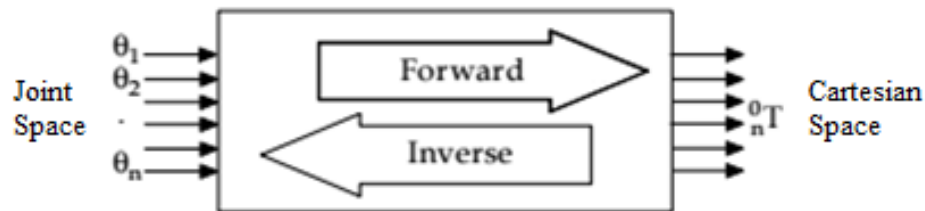


Figure 2.12 Forward and Inverse Kinematics to convert between joint space and Cartesian space

Forward kinematics is used to calculate the rotation and translation of the end effector with respect to the base using joint parameters. *Inverse kinematics* refers to the use of kinematics to acquire the joint positions of the robot that result in the position of the end effector. The inverse kinematics can often be obtained directly from the robot and specifies the 6 joint angles as a vector that can be used as parameters to determine forward kinematics.

A common approach to calculate the end effector when the joint positions are known is by using Denavit-Hartenberg (DH) parameters. Introduced in 1955 (Denavit, 1955), DH parameters are a convention specifying how links in a robot

are connected, enabling the calculation of the end effector using four parameters; joint offset d_i , joint length a_i , joint angle θ_i and joint twist α_i , (Figure 2.13).

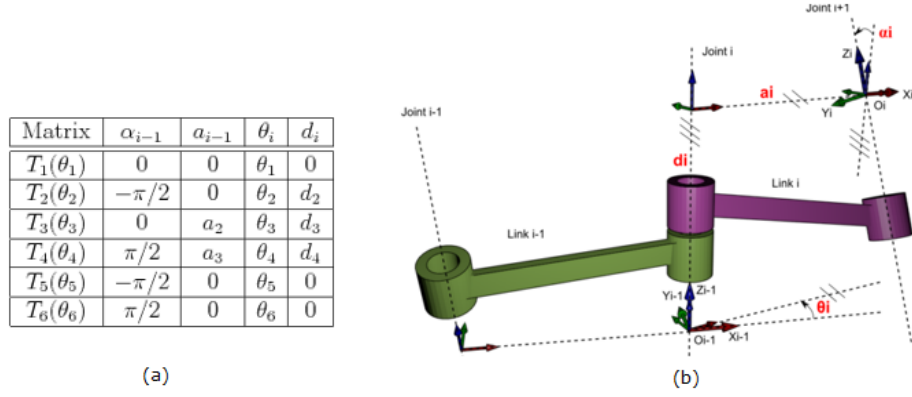


Figure 2.13 Denavit-Hartenberg (DH) Parameters specify how links to the end effector are connected, a) depicts the DH Parameters table, specifying four parameters for each joint, b) a visual representation of each of the four parameters (Denavit 1955).

Alternatively a method known as screw theory can be used (Minguzzi, 2013). Screw theory uses the three variables known as *screw*, a 6-dimensional vector made from a pair of 2D vectors, *wrench*, which is the force and torque, and *twist*, the velocity, to calculate the forward kinematics of the robot. A review of DH parameters and Screw Theory shows that both approaches are very similar, though DH parameters are reported to be used more in literature due to the ease of implementation (Rocha, Tonetto and Dias, 2011).

2.4.1.3 Robot Sensor Calibration

The problem of determining the position of the camera relative to the end effector, known as hand-eye calibration, has been extensively studied, with research going back to the late 1980's, first by Shiu and Ahmad (1987) and then by Tsai and Lenz (1989). To have a fully calibrated robot and camera, hand-eye calibration needs to be performed along with the calibration of the transformation from the world origin to robot base, known as the robot sensor calibration.

To achieve this there are four transformations that must be known. Two can be classified as known during the robot sensor calibration stage; the camera matrix (**A**) and the robot matrix (**B**), section 2.4.1.1 and 2.4.1.2 respectively, whereas the two unknowns are the end effector to the camera (**X**) (Hand-Eye) and the

world to the robot base, the object of interest, ($\mathbf{Y}$). Figure 2.14 depicts the association between each of the transformations.

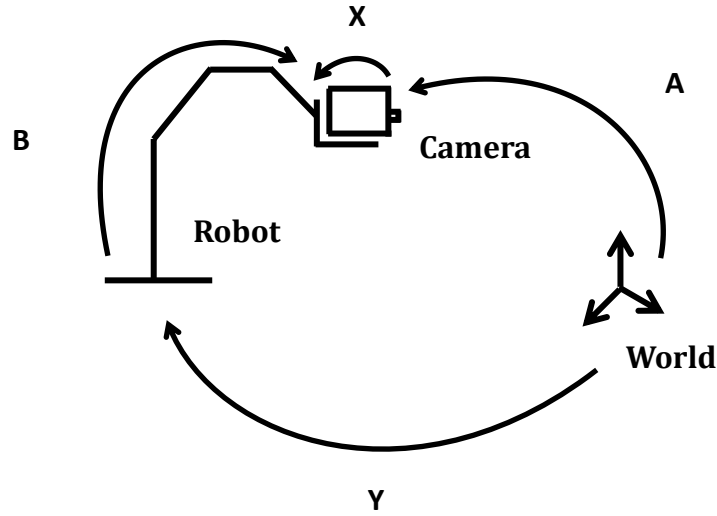


Figure 2.14 The relationship between transformations of matrix equation $\mathbf{AX}=\mathbf{YB}$ for the calibration of the robot and camera.

There are many approaches to calibrating a robot represented as matrix equations $\mathbf{AX} = \mathbf{XB}$ and $\mathbf{AX} = \mathbf{YB}$. The solutions to the matrix equations come in three forms; separable closed form solutions, iterative closed-form solutions and simultaneous closed-form solutions. The latter simultaneously solves the rotational and translational aspects of the equation, where iterative solutions solve both rotational and translational components iteratively and a separable closed-form solution calculates the two components individually.

Each approach has associated difficulties, for example, using a separable approach leads to errors being propagated to the translation of the estimation. Simultaneous solutions can produce matrices that are not orthogonal and iterative solutions often require vast computational power. Shah, Eastman and Hong (2012) review approaches to the robot sensor calibration problem, concluding that each approach suffers from difficulties and that users should adopt an approach most suited for their needs by making a trade-off between difficulty, accuracy and computational requirements. A brief discussion of both matrix representations is below;

$$\mathbf{AX} = \mathbf{XB}$$

This equation aims to solve for one unknown; the transformation between the end effector and the camera. This is represented as a homogeneous equation shown in (Eq. 2.4)

$$\begin{pmatrix} R_A & \vec{t}_A \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_X & \vec{t}_X \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R_X & \vec{t}_X \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_B & \vec{t}_B \\ 0 & 1 \end{pmatrix}$$

(Eq. 2.4)

where R_I and $\vec{t}_I$ denote rotational and translational components respectively. Various approaches have been proposed to solve both simultaneously and separately. Two approaches have been proposed by Horaud and Dornaika (1995) using unit quaternions to develop a closed-form separable solution and a non-linear technique for solving both the rotation and translation simultaneously. They state that although they prefer the non-linear approach the closed-form separable solution does sometimes produce better results.

One of the most commonly used approaches was proposed by Tsai and Lenz (1989) and despite being published almost 30 years ago remains one of the most accurate to-date. They designed a separable closed-form solution based on axis-angles to efficiently calculate for one unknown. From $AX = XB$, Y can be solved as one unknown.

$AX = YB$ (Also known as $AX=ZB$)

An equation of this type solves for two unknowns, which are the transformations between the world to base and end effector to camera. The homogeneous matrix equation, shown in (Eq. 2.5), can be applied to more than one type of robot setup, providing that there are two known and two unknowns.

$$\begin{pmatrix} R_A & \vec{t}_A \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_X & \vec{t}_X \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R_Y & \vec{t}_Y \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_B & \vec{t}_B \\ 0 & 1 \end{pmatrix}$$

(Eq. 2.5)

Dornaika and Horaud (1998) propose a closed-form separable solution to solve for two unknowns using unit quaternions; Li, Wang and Wu (2010) use dual-quaternions and a more recent approach by Shah (2013) uses the Kronecker product to solve simultaneously.

With a calibrated robot-camera cell it is possible to accurately and repeatedly capture images, calculating the camera matrices. A method to ensure that a complete image acquisition process is employed, however, needs to be

designed. There are various image acquisition strategies to ensure this, which are explained in the next section. These strategies help overcome the difficulties of over/under sampling and limit occlusions by calculating the next viewpoint to obtain the optimal amount of information.

2.4.2 Next Best View

Manually controlling the sensor to capture images of the object is a common strategy for image selection, but often requires an expert to ensure problematic areas of the object are sufficiently scanned. However, even this does not offer any guarantees on the quality of the output. Consequently, automated approaches are a preferred choice to acquire images of an object.

The definition of next best view (NBV), also known as the view planning problem, is the process of finding some camera position such that an evaluation function is maximised (Wong, Dumont and Abidi, 1998). However, the ‘best’ camera position is not necessarily always the one that uncovers the most unknown information. For example, SFM requires that views overlap to solve the correspondence problem, furthermore a complex object may require multiple views of the same point to accurately construct a 3D model. A successful evaluation function must capture these application-specific requirements.

Various factors need to be taken into consideration, for example the focal length of the sensor, the degrees of movement, the time taken to calculate the NBV and whether the ‘best’ position is reachable. The problem itself can be very complex, with many issues to consider. One of the primary issues is the cost associated with moving and calculating the NBV, it is crucial to minimise the time spent performing the evaluation of views. A trade-off between the quality of the NBV algorithm and time taken needs to be made. NBV must ensure that the camera is not too close or too far from the object based on some maximum and minimum values. Furthermore, after each movement a pause is necessary to ensure a steady sensor and decrease likelihood of blurring of the image. Other factors to consider are; the percentage of overlap between views, the distance of which the robot needs to move, how to represent the seen and unseen data, how to update the model used to represent the current visibility of the model and the determination of potential camera views.

Approaches to the view planning problem can be categorised as either *search based* approaches - which search a group of potential viewpoints for the best view - or *occlusion-based* approaches - which use occlusion boundaries in the current view to determine the next view. The problem is more formally defined in the next section before a discussion reviewing the state-of-the-art in NBV algorithms.

2.4.2.1 Problem Formulation

NBV can be formally defined as the process of finding a series of camera positions for some unknown object and performing an evaluation on intermediate representations comprised of seen and unseen data to determine those positions. The main problems lie in finding the next best view and designing the intermediate representation of the object.

A number of constraints need to be considered in view planning;

1. **New information:** the NBV must see unknown surfaces to ensure the object is completely seen
2. **Position constraints:** The view must be reachable by the robot
3. **Sensing constraint:** The surface to be seen must be within the sensor's field and depth of view
4. **Overlap constraint:** There must be sufficient overlap between images to allow reconstruction, greater overlap is necessary for more complex objects
5. **Voxel View constraint:** Each voxel must be seen in n images and the angles between views must exceed a threshold of α .

Further, relaxed, constraints include:

1. **Distance constraint:** The distance that the robot needs to travel should be minimal
2. **Perpendicularity constraint:** If the angle formed between the surfaces' normal and the sensors orientation is close to 0 the scan will have better quality.

If the depth of the object is unknown, i.e. the sensor used is a standard RGB camera, then further constraints need to be considered

1. **Object position:** Are we sure we are capturing information from the whole object?
2. **View constraints:** Are we sure this seen voxel is part of the object?

Algorithm 2.1 illustrates the common NBV algorithm. NBV is initialised with as a single voxel. An arbitrary image of the object is captured and the volumetric model is updated, increasing the depth and knowledge of the object. Until some evaluation criterion is met the NBV algorithm iterates, evaluating the model V and determining the NBV. New images are captured and V is updated using this.

Algorithm 2.1 Next Best View Algorithm

Input : *An example volume V*

Output : *A series of 2D images*

- 1: *Initialise empty volume V*
 - 2: *Capture initial arbitrary image I*
 - 3: *Update the volumetric model*
 - 4: *while($nbv = true$)*
 - 5: *Calculate the NBV*
 - 6: *Capture the image*
 - 7: *Update the volumetric model*
 - 8: *if termination criteria is met, $nbv = false$*
 - 9: *end*
-

2.4.2.2 Existing approaches

In 1985 (Connolly, 1985) formally introduced the NBV problem and proposed two algorithms. These are known as the planetarium algorithm and the normal algorithm, both of which use an octree data structure to represent seen and unseen data.

The planetarium algorithm creates a sphere around the object which is evenly sampled. Each sample is a potential camera position directed towards the centre of the object, this is known as the view sphere. All samples on the view sphere are evaluated and the one uncovering the most unknown information is selected. The process terminates when no improving views can be found or when a certain percentage of the object is seen.

The normal algorithm uses the sum of the number of face normals for each unseen voxel to calculate the viewpoint, resulting in 6 potential viewpoints (Figure 2.15). The normal with the largest unseen sum is selected. The problem with this approach is that if the object is self-occluding then it is impossible to

complete the object, as there are only 6 potential camera positions. Numerous approaches have stemmed from these two algorithms, where the core concept remains the same, but improvements have been made to the efficiency.

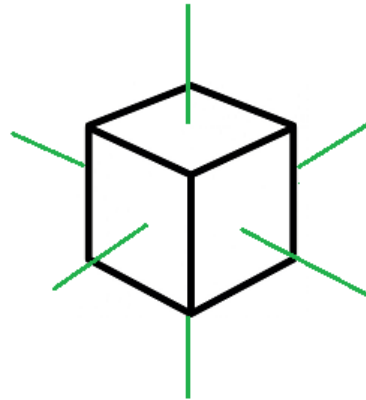


Figure 2.15 *Normal Algorithm* used to calculate normal scores for x , $-x$, y , $-y$, z , $-z$; the six possible normals.

Massios, Nikolaos A. (1998) use a voxel representation to store the known information about the object. They create a view sphere and project each voxel to each viewpoint to determine whether it can be seen by the corresponding camera. They select the view with the highest score, typically the one that maximises the amount of unseen data. A quality is assigned to each voxel that aims to maximise the quality of seen data by improving the view angle using a least squares fitting to calculate the normal and compare with the angle of the viewpoint. However, the objects used in the experiments tend to be very small and lack any complexity. Moreover, the time to calculate the NBV is 23.5 seconds and the largest object their approach was applied to was $5 \times 10^7 \text{cm}^3$

Wong, Dumont and Abidi (1999) use a volumetric approach and three algorithms. The *planetarium algorithm* is defined as the optimal method as given a densely sampled view sphere it is possible to completely capture the object. The *normal algorithm* sums the surface normal of voxels that point in each axis direction, the sums are used to define 8 viewpoints that give an estimation of poses from which to acquire new knowledge. Finally, the *adaptive algorithm* combines the previous optimal and normal algorithms to reduce run time by using the normal approach but guarantee it can be completed by incorporating the optimal method. The optimal method can take up to 192 seconds to determine the NBV whilst the adaptive method takes up to 17 seconds. The main disadvantage here is that it has not been applied to the real

world and therefore it does not take into consideration the feasibility of camera positions.

A. Hornung (2008) focus on selecting images for stereovision and have adjusted the standard NBV algorithm to do so. They propose a three-step algorithm, in the first iteration they focus on ensuring that every voxel has been seen at least once. The second step ensures that each voxel has been seen at least k times, a user defined variable. The final iteration aims to gather more information for problematic regions. In the proposed algorithm they do not capture images and instead select images from a pre-defined set. While this shows promise, the process of capturing images has already been conducted and the true ‘best’ view might not be available within the pre-defined set. They have shown that they can maintain the quality and reduce the size of the dataset using this, but it is impractical for complex highly occluded objects as it offers no guarantees that the image required is in the image set.

Vasquez-Gomez, Lopez-Damian and Sucar (2009) propose a volumetric approach with a utility function based on the navigation distance, occlusion, quality and unseen areas, to determine the NBV. The view sphere is sparsely sampled with dense sampling applied to higher ranking viewpoints to reduce computational requirements associated with calculating the NBV. However, evaluating each viewpoint is still computational expensive with reports of up to 8 seconds for under 80 views.

Work done by Vasquez-Gomez, Lopez-Damian and Sucar (2009), and later expanded by Vasquez-Gomez, Sucar and Murrieta-Cid (2013), uses a voxel representation with a depth of 16 to represent the object. The main contribution is known as Hierarchical Ray Tracing (HRT), an adaption of Uniform Ray Tracing (URT). URT uses a high-density sampling size and projects rays at the maximum depth. HRT starts with a low density set of rays and if an occupied voxel is hit by a sparse ray a denser set of rays are cast, the process repeats until the maximum depth is reached. This significantly reduces the computational requirements. They produce a utility function based on overlap, distance and unseen surface that ranks the 1,500 potential viewpoints, this takes 10.3 seconds per iteration with a further 6.3 seconds required to update the octree after each view is captured. This is a significant improvement on their previous work. The author does however state that when using HRT more images may be required than in dense ray tracing.

Vasquez-Gomez apply the HRT method and utility function to a real environment using a mobile manipulator (J. I. Vasquez-Gomez, 2014). They use an occupancy grid map to store object information and a Kinect to gather depth information. They use the three simple objects for their experiments which they successfully reconstruct, with an average of 11 seconds taken to calculate the NBV. They determine camera positions in the configuration space, starting with 10,000, and refine them as the search progresses using the utility function. Their main evaluation is the distance travelled by the robot, the overlap of images and the number of unknown voxels seen in the new view, improving their previous work.

Juan Irving Vásquez (2011) use a view sphere to determine an initial set of candidate views and a voxel grid to represent the search space. They were one of the first to propose a utility function that takes into consideration both the distance travelled by the sensor and the amount of unknown information acquired in each view. All seen voxels are marked with normals and quality values, the quality relates to how perpendicular the camera is to the voxel. They refine the view sphere to reduce the number of potential views by first using a sparse view sphere and selecting the highest-ranking views based on the utility function and increasing the density, repeating the process until the maximum density is reached.

Li and Xiao (2015) use multiple cameras and robots that work in parallel, successfully reducing the time required to calculate the NBV. Unlike the approaches discussed so far, Li calculates the potential viewpoints in the configuration space eliminating the need to test views using the robot's inverse kinematics. Their main contribution is with regards to the potential views; not only do they calculate the views in configuration space, they generate the initial set based on the specification of the robotic joints (angles and lengths). This reduces the view sphere from 10,000+ viewpoints to 2,000-5,000; reducing the computational requirements by an order of magnitude.

2.5 Discussion

The work described in this thesis draws on several distinct areas of the literature, which have been reviewed here. Different approaches to plant modelling have been examined: reconstruction remains a challenging computer vision problem in which advances in both reconstruction technology and optimal data

acquisition techniques are required. It is evident from the literature that more uniform and automated approaches are required if the bottleneck associated with phenotyping is to be reduced. Potential advantages of active vision have been identified and how it can be used to improve the plant reconstruction process has been discussed.

A significant amount of literature proposes methods of segmentation that rely either on manual methods of identifying or aiding the detection of contours or by separating the background from foreground using algorithms that assume the plant is green. Although these approaches do work it is not acceptable to assume the plant is always going to be green, a problem that will be addressed in this research.

Reconstruction of plants, and in general, is an expanding area of research, particularly as the computational capabilities of machines increase. More demanding algorithms and reconstruction techniques are being developed but the difficulties remain. The research conducted here aims to address some of the issues associated with producing accurate reconstructions including: occlusion, determining adequate sample numbers for accurate reconstruction, calibration and data acquisition through active vision and a fully automated non-destructive, non-intrusive process without the prerequisite of botanical or computer expertise. We expand on the structure from motion approach to 3D reconstruction due to its ease of use, lack of reliance on botanical expertise and the potential to produce a cheap set up using standard cameras.

Chapter 3

Cell Composition

The chapter is divided into three broad sections discussing the setup of the hardware and software environment designed and developed in this work. In the first section the chosen hardware is discussed and justified, providing a brief explanation of the technologies used to enable communication between each of the components. In section two the software environment developed in this work is discussed, with an overview of its functionality and key attributes. Section three, the core of this chapter, evaluates the state-of-the-art calibration methods. An explanation of the calibration performed in this work is then presented before moving on to explain how the calibration problem has been expanded to include the turntable.

3.1 Hardware

Image acquisition in plant phenotyping is currently reliant on either manual (Pound *et al.*, 2014) or static cameras (Photon Systems Instruments, Drasov, Czech Republic; LemnaTec, Aachen, Germany; Qubit Phenomics, Kingston Ontario, Canada) that are unable to adapt to a specific plant species or variety. Consequently, there is an increase in labour, reduction in accuracy and often very expensive systems are designed for a single plant species (Arvidsson, Pérez-Rodríguez and Mueller-Roeber, 2011; Junker *et al.*, 2015; Arend *et al.*, 2016). This research employs an active vision approach to reduce the manual requirements of image capture, ensuring that sufficient data is obtained, irrespective of plant species. Moreover, this work is applicable to any object that can be positioned on the turntable and within the reach of the robot arm.

To make this possible, a workspace has been composed consisting of three primary hardware components.

1. A high precision turntable (LT360EX – Linear X Systems, Portland, USA), Figure 3.1.a) which is used to rotate the plant, enabling a full 360 degrees of rotation, providing repeatability and accuracy of 0.001 degrees.

2. A robot arm, The Universal Robot 5 (UR5 – Universal Robots, Odense, Denmark), providing 6 degrees of freedom (6DOF), which, in conjunction with the turntable, ensures the whole object can be viewed. The UR5 consists of 6 revolute joints which allow a sphere workspace of 170cm diameter; Figure 3.1.b depicts the joint setup.

The UR5 has been chosen to manipulate the view point of the camera due to its collision detection features which mean no safety fences or extreme precautions need to be taken within the workplace, allowing use among people. Moreover, the UR5 is incredibly lightweight and does not need to be fixed to the floor, as a result it can be moved freely and repositioned as necessary.

The UR5 comes with a tablet-like device with a graphical user interface known as PolyScope, which supports basic commands and allows for simplistic programs to be created. However, PolyScope is restricted in its capabilities; offering limited flexibility and does not support communication with multiple systems, making the robot, as is, a standalone machine.

3. A camera, the Canon 650D (Canon 650D – Canon, Tokyo, Japan), for acquiring data with a fixed focal length. The 650D was chosen as this is affordable and captures high resolution images of up to 18 megapixels, Figure 3.1.c.

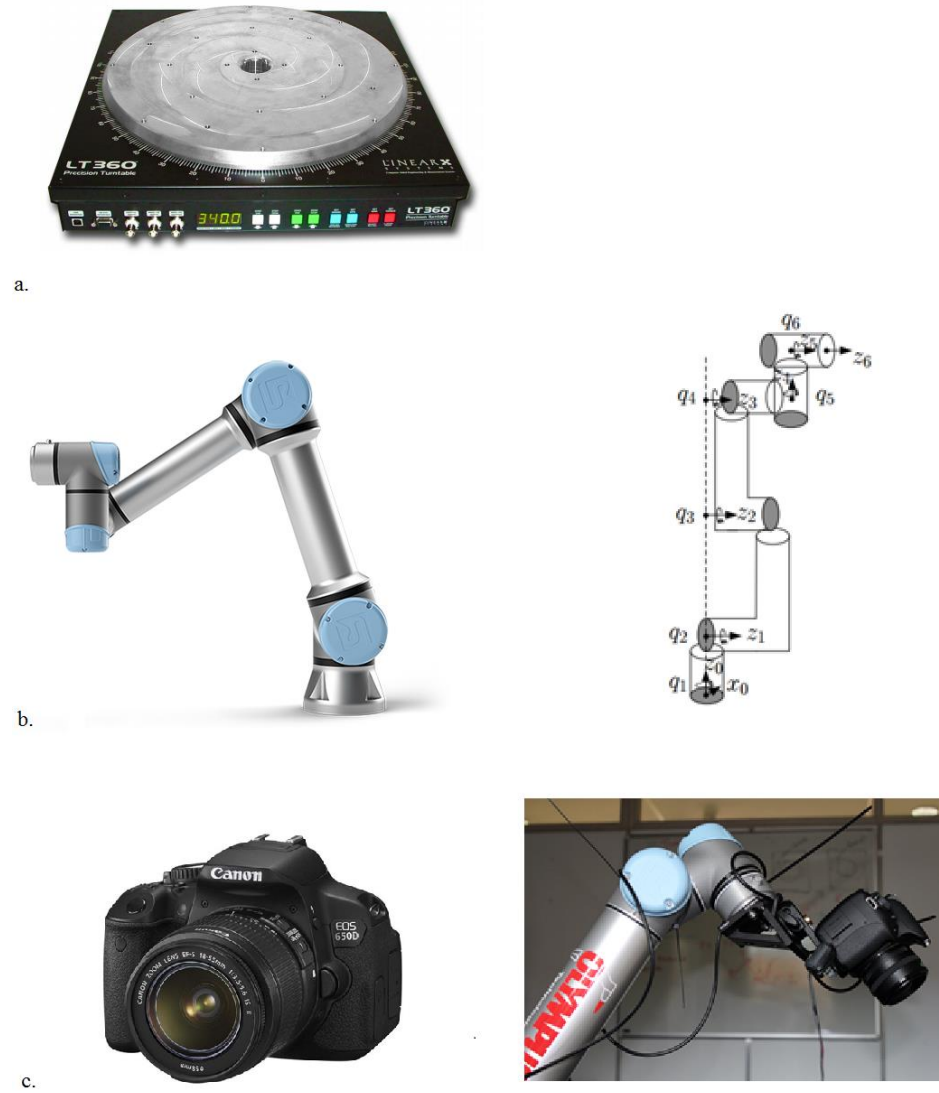


Figure 3.1 The three main hardware components that make up the active vision cell; a. LT360 Ex Turntable, b. The UR5 robot arm, c. The Canon 650D camera

The interface designed here (discussed in Section 3.2) enables communication between the robot and desktop computer, known from this point forward as the host, and is achieved over TCP/IP. Communication between the turntable and host can be performed using serial port communication via sockets (RS-232) and communication between camera and host via USB, Figure 3.2.



Figure 3.2 Communication between the devices is performed using an API, RS-232 and Sockets all commands come from the server which controls the functionality of all devices

3.2 Software Design

The software, **Plant 3D**, comprises a large proportion of the project. An interface to manipulate the hardware, allowing communication between each component, has been designed and implemented. The primary goal of the interface is to produce a simplistic user experience allowing non-specialists (i.e. biologists) to use the system, Figure 3.3 illustrates the user interface. The main features of the interface are;

1. **Hardware communications:** the interface has been designed to communicate with the three key hardware components, enabling commands to be sent and received. For example, moving the robot, capturing images and rotating the turntable.
2. **Calibration:** automatic calibration has been included in the interface allowing calibration of the camera, robot and turntable to be performed easily. This is discussed in section 3.3.
3. **Live view:** a live view from the camera's feed has been included allowing the user to see the view of the camera as it gathers data or performs calibration. An Euclidean colour filter can be applied in real time allowing users to view the silhouette of the object before capturing data, a crucial aspect of the performance of the generation of a volumetric model; discussed in Chapter 4.

4. **Active Vision:** the interface supports the entire active vision process, as discussed in Chapter 4, and can acquire data for an arbitrary object with a single click and without the need for manual parameter setting.
5. **Miscellaneous:** a series of miscellaneous features have been included namely; **object loader** - which allows 3D reconstructions to be viewed, **camera view** - which allows the camera positions of a set of captured images to be seen with respect to the object, **PMVS** - integration of PMVS and the surface reconstruction work (Chapters 5 and 6).

Further software designed and developed in this thesis includes;

1. All active vision cell software has been developed including the octree representation, the volumetric model evaluation and representation and the incremental view sphere.
2. Point cloud processing in which a series of algorithms have been designed and implemented from scratch to manipulate point data; i.e. loading and saving point sets, manipulating size, location, colours and grouping points into clusters.
3. Surface reconstruction has been designed and developed in conjunction with level sets to produce 3D fully connected models. This includes the merging of clusters and smoothing of surfaces.
4. A virtual environment was created allowing data obtained using the CT-Scanner to be used in the robot setup virtually; supporting the process of the real world setup.

A variety of software, both premium and open source, has been used throughout this work, namely: **OpenGL** which is used to support the rendering of 3D models in the interface (<http://www.opengl.org/>) and **EmguCV**, a C# adaption to OpenCV, which is used to aid the calibration of the camera (<https://www.emgu.com/>). The **Canon SDK** to aid communication with the camera and is adapted for our needs. **Matlab** was used for various testing purposes particularly the ray intersection discussed in Chapter 4 (<http://www.matlab.com>). Finally, **Visual Studio** which was used for application development.

3.3 Calibration

The calibration process is summarised in this first paragraph before providing a more in-depth explanation for the remainder of the chapter. To determine the position of components in world space, the robot, camera and turntable must be fully calibrated; that is, the translation and rotation in world space must be known. To begin, $\mathbf{A}$, the camera position, and $\mathbf{B}$, the robot kinematics, must be found. $\mathbf{A}$ can be calculated by performing camera calibration and $\mathbf{B}$ can be calculated from the robot joint angles (Section 3.3.1). $\mathbf{X}$, the end effector of the robot, and $\mathbf{Y}$, the world position of the object, are the two unknowns which need to be calculated (Section 3.3.2). $\mathbf{ABXY}$ are all transformation matrices consisting of rotation and translation. To calculate the matrices, the robot needs to move at least $n = 3$ times to obtain $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$ and $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$. Previous work (Shah, 2013), explains the need for 3 movements to obtain a unique solution. However, the accuracy improves as more movements are made, up to some threshold where improvement plateaus as images become redundant; this is discussed and evaluated later in the chapter.

To calibrate the turntable (Section 3.3.3), the centre of rotation is calculated by calibrating for four unique, opposite world origins, by rotating the turntable four times. Once the calibration is complete the checkerboard is removed and $\mathbf{A}$ becomes the one unknown for each robot movement.

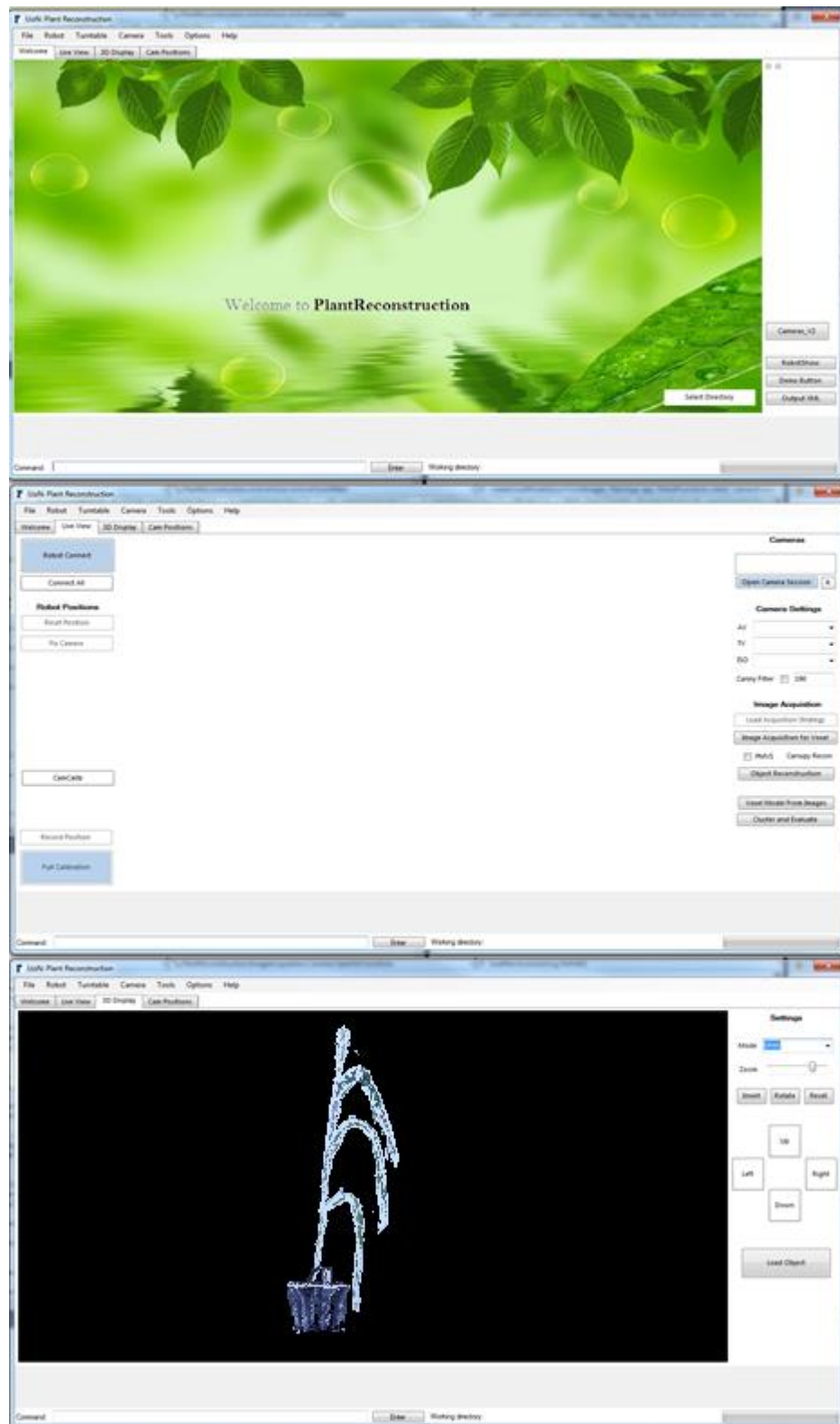


Figure 3.3 The user interface for controlling the active vision system; Top: Main screen for loading. Middle: The active vision interface for manipulating the camera, robot and turn table. Bottom: The object loader to view and inspect the object reconstructions

3.3.1 Camera and Robot Calibration

Camera Calibration (A)

Camera calibration is achieved using a planar target: in this instance, images of a checkerboard are taken at distinct viewpoints. The process uses EmguCV, a cross platform .NET wrapper for OpenCV, which is a library of functions particularly suited to image processing. Image acquisition is performed using pre-configured robot positions to capture images of the checkerboard with only one constraint; that each corner of the checkerboard must be visible in all views. From these images the position of each square is determined and then re-projected to solve the rotation and translation of the camera position by calculating a conversion between the camera's natural units (pixels) and real-world measurements (millimetres). The outputs of this is the camera model which consists of the intrinsic (the internal workings of the camera, namely; focal length, sensor format and principal point) and extrinsic parameters (the external workings, namely; a rotation and translation), and a re-projection error which specifies the accuracy of the calibration. For the research conducted here, any error of less than 1mm is acceptable. An error greater than 1mm could reduce the accuracy of reconstruction, as error propagates through each transformation matrix.

It was found that allowing EmguCV to estimate the intrinsic parameters decreases the re-projection error by 20% but significantly reduces the quality of reconstruction, due to the fact the camera parameters are being estimated for a specific dataset and as a result, reduce the overall accuracy when applied to other datasets. When calculating the unknown A from XY using an estimated intrinsic matrix the error increased by up to 200%, in extreme cases, when compared to results obtained with a pre-defined intrinsic matrix. As a result, the intrinsic parameters are obtained directly from the camera and manually specified. Intrinsic parameters are typically available on the camera manufacturers' website and should be specified when known.

(Zhang, 2000) states that 10-15 images are sufficient for camera calibration and that error does not decrease significantly with the addition of more images. The calibration process involves five primary steps; 1. Capture a series of images of the checkerboard and distinct viewpoints. 2. Find the checkerboard corners in each of the images. 3. Find the checkerboard sub-corners for each image, the corner

of each square. 4. Calculate reprojection and re-project the points. 5. Calculate the error and determine whether additional views are required.

Forward Kinematics (B)

To solve the forward kinematics for the robot, Denavit–Hartenberg (DH) Parameters are used in which only one of the four necessary parameters, the joint angle θ_i , is variable and the other three parameters, link length a_i , link twist α_i and link offset d_i are all constant. The parameters for the UR5 are shown in Table 3.1.

Joint	a_i (m)	α_i (rad)	d_i (m)	θ_i
1	0	$\pi/2$	0.089459	q_1
2	-0.42500	0	0	q_2
3	-0.39225	0	0	q_3
4	0	$\pi/2$	0.10915	q_4
5	0	$-\pi/2$	0.09465	q_5
6	0	0	0.08230	q_6

Table 3.1 DH-Parameters for UR5. Constant link length a_i , link twist α_i and link offset d_i and the variable joint angle θ_i

The variable joint angle can be acquired from the UR5 using the interface designed here which returns a vector of the 6 joint angles in radians that results in the orientation and translation of the end effector at the current position. By solving (Eq. 3.1) we can obtain the transformation matrix $\mathbf{B} = [\mathbf{R}|\vec{\mathbf{t}}]$, from base to end effector

$$\mathbf{B} = T_6^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 \quad (\text{Eq. 3.1})$$

where T_j^i denotes the homogenous transformation from joint i to j and

$$T_i^{i-1} = \text{Trans}_{z_{i-1}}(d_i) \cdot \text{Rot}_{z_{i-1}}(\theta_i) \cdot \text{Trans}_{x_i}(a_i) \cdot \text{Rot}_{x_i}(\alpha_i) \quad (\text{Eq. 3.2})$$

where,

$$\begin{aligned}
 Trans_{zi-1}(d_i) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 Rot_{zi-1}(\theta_i) &= \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 Trans_{xi}(a_i) &= \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 Trans_{xi-1}(d_i) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

(Eq. 3.3)

which can be written as,

$$\begin{aligned}
 T_i^{i-1} &= \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \cos(\alpha_i) & \sin(\theta_i) \sin(\alpha_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \cos(\alpha_i) & -\cos(\theta_i) \sin(\alpha_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

(Eq. 3.4)

The UR5 uses axis-angles, a unit vector indicating the direction of an axis of rotation, and an angle describing the magnitude of the rotation about the axis, to perform movements. This is calculated using (Eq. 3.5), and results calculated here are compared to those visible on PolyScope to verify correctness. For each of the movements, the axis-angle vector was accurate to three decimal places, which is sufficient for the work being conducted here.

$$\theta = \arccos\left(\frac{\text{trace}(B) - 1}{2}\right)$$

$$w = \frac{1}{2\sin(\theta)} \begin{bmatrix} B(3,2) - B(2,3) \\ B(1,3) - B(3,1) \\ B(2,1) - B(1,2) \end{bmatrix}$$

$$\text{Axis Angle} = \theta w$$

(Eq. 3.5)

40-50 images are captured to solve the full calibration of the robot and camera simultaneously, effectively reducing computational requirements. The reasons behind this are discussed later in the chapter. After performing camera calibration, the matrices for each of the camera and robot positions are known which can be used to solve X and Y (discussed below); known as the sensor and base calibration.

3.3.2 Sensor and Base Calibration

Once matrices A and B are known, it is possible to solve the linear equation for X and Y , (Eq. 3.6), fully calibrating the robot and camera setup.

$$\begin{pmatrix} R_{Aj} & \vec{t}_{Aj} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_X & \vec{t}_X \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R_Y & \vec{t}_Y \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_{Bj} & \vec{t}_{Bj} \\ 0 & 1 \end{pmatrix}$$

(Eq. 3.6)

(Eq. 3.6) is split into and solved as two components, the rotation and translation, (Eq. 3.7).

$$R_{Aj}R_X = R_YR_{Bj}$$

$$R_{Aj}\vec{t}_X + \vec{t}_{Aj} = R_Y\vec{t}_{Bj} + \vec{t}_Y$$

(Eq. 3.7)

Three calibration methods for solving the linear equation (Dornaika and Horaud, 1998; Li, Wang and Wu, 2010; Shah, 2013) were evaluated for accuracy to determine which is most suited for this work. The evaluation function proposed is based on the re-projection error. Re-projection was chosen as the error function due to its importance for volumetric intersection (Chapter 4) and level set methods (Chapter 6) which both require the projection of 3D world space coordinates to 2D image coordinates to obtain pixel data. Evaluation is performed as follows; A is calculated for each image using the robot position B

and estimated X and Y values, obtained from each of the calibration methods, (Eq. 3.8).

$$A_i = (X(YB_i)') \quad (\text{Eq. 3.8})$$

Using the camera matrix $A = \begin{bmatrix} R & \vec{t} \\ 0 & 1 \end{bmatrix}$ obtained from the camera calibration stage, a 3D point $P = [x \ y \ z \ 1]$ is projected to the 2D image using (Eq. 3.10).

$$\begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \left(\begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot A \right) \cdot P$$

$$x_{i'} = \frac{x_i}{z_i}$$

$$y_{i'} = \frac{y_i}{z_i} \quad (\text{Eq. 3.9})$$

The same 3D point is then projected using the estimated camera matrix. The error, e , is calculated as a sum of differences between the position of the point obtained through camera calibration, classified as the correct position, and those that are estimated, (Eq. 3.10).

$$e = \left(\sum_i^1 Abs(\vec{A} - \vec{A}_i) \right) / i \quad (\text{Eq. 3.10})$$

An experiment was conducted by capturing 200 images with known camera parameters, A , actual robot position B and estimated X and Y . The results are shown in Figure 3.4. When additional images are used, accuracy first improves and then begins to plateau. Given the trade-off between speed and accuracy, it is reasonable to set 50 images as the upper limit for this research.

From the results shown in Figure 3.4, the Dornaika and Shah approaches outperform that of Li quite significantly. Even when using 200 images the error remains above 5. The results produced by Dornaika and Shah are almost identical, both with low errors of which are acceptable. The closed-form approach proposed by Dornaika (Dornaika and Horaud, 1998) has been chosen

to solve the linear equation $AX = YB$ as it overcomes the limitation of angles being equivalent to 0 or pi, which the approach by Shah is unable to do. Moreover, it is easy to implement, and quaternions are quicker to manipulate. The chosen approach uses algebraic properties associated with quaternions to solve simultaneously.

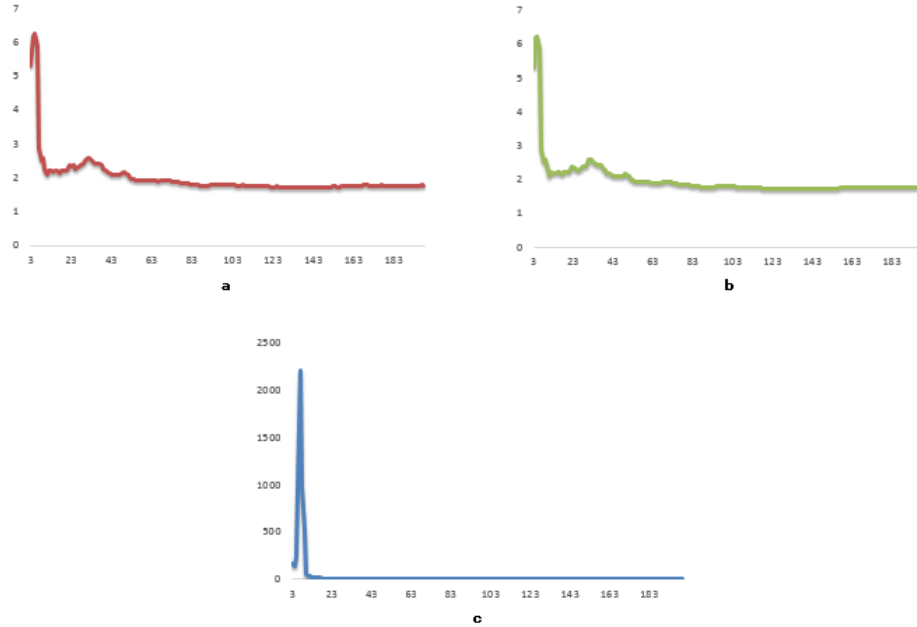


Figure 3.4 Results of estimating the camera parameters using X and Y and the error of re-projection in pixels. a) Dornaika - Quaternion approach (Dornaika and Horaud, 1998), b) Shah - Kronecker Product (Shah 2013), c) Li - Dual-quaternions and Kronecker Product (Li et al. 2010). Where the a-axis is the error and the y-axis is the total number of images

3.3.3 Turntable calibration

The final calibration component to be solved is the rotation of the turntable, in which the centre of rotation needs to be calculated. The turntable is vital for this work as the reach of the robot is unable to obtain views from a low position on the opposite side of the object - that furthest away from the robot. By rotating the object the robot is able to scan the entire object maximising the number of potential views and reducing the likelihood of being unable to capture data of self-occluded areas. To solve for the turntable centre, camera calibration needs to be performed at four rotations to acquire four unique origins, where the origin is the bottom right corner of the checkerboard, or (0,0,0) in world space. Two sets of two opposites, 180 degrees apart, are used calibrating for 0, 180, 70 and

250 degrees. Once four unique calibrations have been calculated the 3D point $p = [0 \ 0 \ 0 \ 1]$ is projected to 2D image coordinates for each, as shown in Figure 3.5. The centre of rotation can then be solved as a line intersection problem.

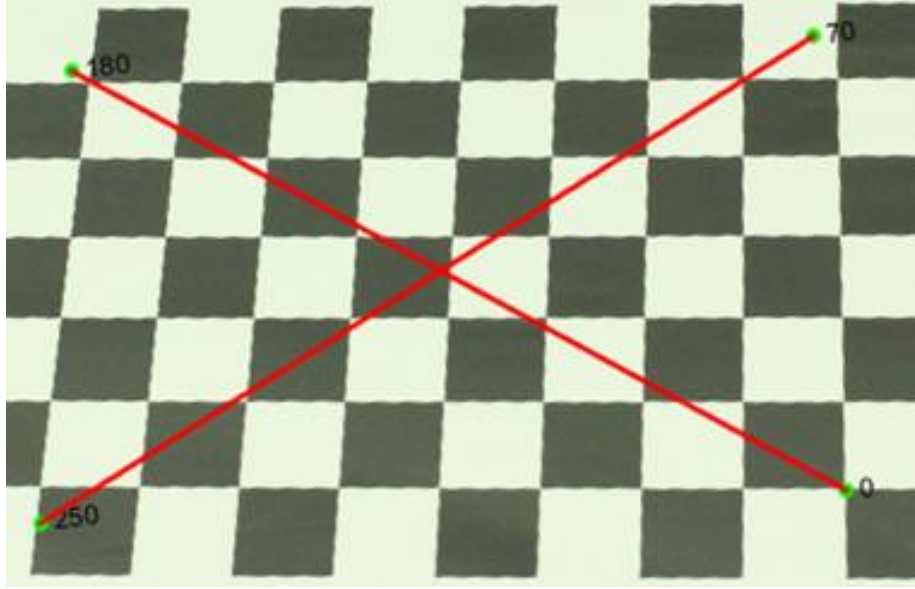


Figure 3.5 Illustration of the world coordinates (0,0,0) at four calibrations for 0, 70, 180 and 250

Coordinates for the two lines $P^0 \rightarrow P^{180}$ and $P^{70} \rightarrow P^{250}$ are defined by two 2D points, from which the point of intersection can be calculated first by forming the line equations (Eq. 3.11), and then by calculating the slope of the lines (Eq. 3.12).

$$\begin{aligned} Y &= m^1 * x + b^1 \\ Y' &= m^2 * x + b^2 \end{aligned}$$

(Eq. 3.11)

where

$$m^i = (P_y^i - P_y^{i+1}) / (P_x^i - P_x^{i+1})$$

(Eq. 3.12)

From the slope and equation of the lines, the point of intersection can be calculated by subtraction

$$X'' = Y - Y' = m^1 * x + b^1 - m^2 * x + b^2$$

$$Y'' = m^1 * x + b^1$$

(Eq. 3.13)

where X'' and Y'' are the coordinates of the line intersection.

Figure 3.6 illustrates the output of the estimate of the centre of rotation through line intersection.

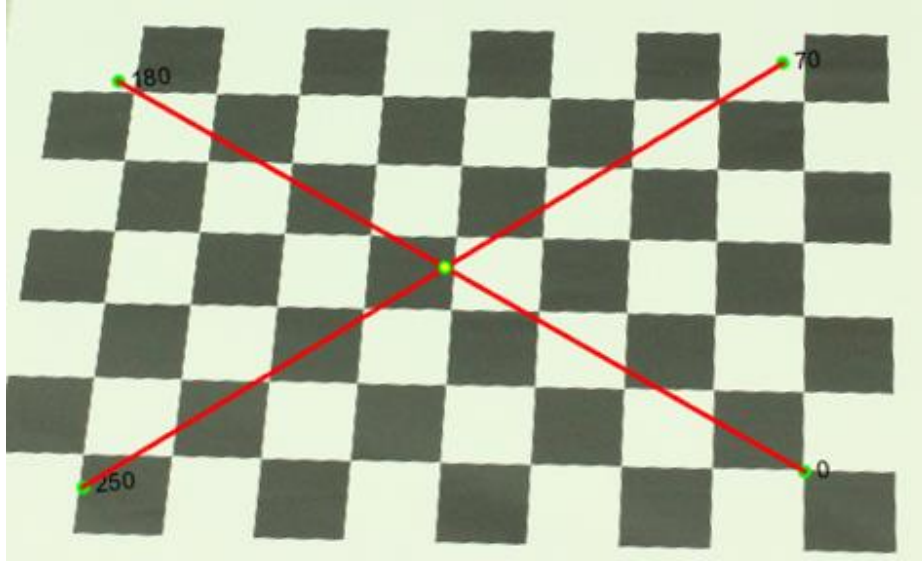


Figure 3.6 The centre of rotation is determined by solving the line intersection problem using 4 unique camera calibrations and projecting into a single 2D space.

Given the centre of rotation, Y is adjusted to consider the rotations of the turntable by rotating the robot position, the transformation from the world to base, in accordance to the rotation performed. Therefore, once solved for X and Y , it is possible to calculate A at any given time using (Eq. 3.14).

$$A_i = (X(Y_j B_i))'$$

(Eq. 3.14)

where X is the known transformation matrix, B_i is the robot kinematics for position i and Y_j is the robot transformation, where j is the rotation. To calculate this, project to the centre of the turntable, which is known from the calibration process. From this it is possible to calculate Y_j , the rotation of Y (Eq. 3.16).

$$Y_j = [YR_j \overrightarrow{Yt_0}] * \overrightarrow{Yt_j} \quad (\text{Eq. 3.15})$$

where,

$$YR_j = (Y'RotZ(j))' \quad (\text{Eq. 3.16})$$

The translation, Yt , requires that the difference between centre of the turntable before and after rotation is known; (Eq. 3.17), where $\vec{c}$ is the homogeneous position of the centre of the turntable. x' and y' is multiplied by Yr with its original translation from Y_0 to obtain Y_j

$$\begin{aligned} \vec{p} &= \vec{c} - (RotZ(j)\vec{c}) \\ x' &= -\vec{c}_x + (-\vec{p}_x) \\ y' &= -\vec{c}_y + (\vec{p}_y) \\ \overrightarrow{Yt_j} &= [x' y' 0 \ 1]' \end{aligned}$$

$$(\text{Eq. 3.17})$$

Now that the position of the camera is, or can be, known for each movement, the setup is said to be fully calibrated. The checkerboard can be removed from the environment and it is now possible to accurately obtain the position of the camera with respect to the world, which enables 3D reconstruction. Traditional methods of camera calibration (i.e. those that rely on planar targets), require that some calibration target is visible in all images to obtain camera parameters, particularly for highly complex scenes with lack of texture or distinct features, such as plants. Consequently, views are restricted and large objects often become problematic as the calibration target is occluded by the object. The calibration method here eliminates this constraint and can calibrate the camera for any position without the need for a calibration target, once the robot and turntable calibration parameters have been estimated.

Chapter 4

Active Vision for Multi-View Stereo

In this chapter the novel Next Best View (NBV) algorithm used to acquire the 2D images and intermediate object representation is discussed. The chapter is broken down into seven sections. Section one provides an introduction and puts the work into context. Section two provides an overview of the algorithm and a discussion of the novel aspects with respect to the state-of-the-art. Section three defines how the intermediate model is represented and updated. In section four, the first of the novel contributions is presented; how 2D images are acquired to increase knowledge of the object. Section five is the second novel contribution of this chapter and looks at how intelligently positioning the camera can improve the quality of the 3D modelling while reducing the computational requirements of existing NBV algorithms. Section six describes how the work presented here further reduces the necessary computation. Finally, section seven presents an evaluation of the cell against existing methods with respect to the quantity and quality of data obtained. The results show that intelligently gathering images has advantages over traditional methods; those that rely on static or arbitrary viewpoints. Evaluation is performed against ground truth data obtained using an X-ray CT-Scanner.

4.1 Introduction

Chapter 3 presented an active vision cell (AVC) intended to support the recovery of 3D models from 2D images. This chapter, and the next, present the active vision and 3D modelling algorithms needed to achieve this. Here, a novel framework is proposed to capture a series of images for 3D modelling via multi-view stereo (MVS). MVS requires a certain degree of redundancy in the input images, particularly for highly occluded, complex objects, where feature matching becomes more challenging. MVS is favoured over other approaches as it can deal with the deep concavities and small object areas of plants, which volumetric methods do not, and at the same time avoid issues that lasers present with respect to reflective surfaces and illumination (see Chapter 2).

The aim here is to increase the flexibility and automation of MVS whilst at the same time improving the quality of the 3D model. The aim is not to recover a surface model of the scene, which is performed at a later stage (Chapter 6), but rather to find a set of sensor positions, that best acquire a 3D point set, by maximising a utility function (proposed in this chapter) which ensures sufficient views and angles between views have been obtained.

The proposed method differs from existing NBV algorithms in that it uses neither a depth sensor, nor a predefined set of images. Moreover, the algorithm presented here reduces the search for viewpoints by using clusters of voxels and an incremental view sphere - as view sphere that increases the number of potential camera positions over time, as opposed to evaluating all views and all voxels. Redundant images are discarded to further improve efficiency. The novel contributions of this chapter are; an approach to select the first image for NBV, a methodology for acquiring images for NBV without depth, an incremental view sphere to determine potential viewpoints and a utility function aimed at capturing data for MVS.

To evaluate the AVC's point clouds, X-ray images of target plants were obtained using a GE v|tome|x M scanner housed in the University of Nottingham's Hounsfield Facility. The v|tome|x M provides volumetric images with a voxel resolution of 5 - 150 μm and, more importantly, is not subject to the occlusion problems faced by visible light imaging. Though some X-ray segmentation tasks are highly challenging, plant material and air are easily separated in the density data provided by μCT and, following noise reduction

with a median filter, plant material can be identified by applying a user-defined threshold. From this a complete image of the plant is formed. The surface of each plant can be represented in standard triangular mesh format, providing a data structure (i.e. a ground truth model) against which point clouds obtained from the AVC can be compared. The algorithm proposed here is evaluated against the X-ray CT scans based on the number of points obtained and the distance from points to the surface of the ground truth. Figure 4.1 illustrates the data obtained, consisting of examples of seven species of plants that vary in terms of size and complexity.

It is worth noting that while the X-ray CT Scanner produces accurate, highly detailed models suitable for evaluation, it is ill suited to most plant phenotyping tasks due to size restrictions, time requirements and a very costly setup. Moreover, it is possible for the scanner to miss thin areas of the plant, resulting in an incomplete model. Note: Models used for the ground truth in this work have been successfully scanned: the test plants have thick enough leaves that the X-ray CT scanner is able to accurately reproduce the plant. The species and scanning method used here were chosen so that there is a complete ground truth model in which to evaluate the results against.

The results in chapter 4, 5 and 6 are all compared to the ground truth data obtained in the CT scanner. The images used to create the 3D model for evaluation against the ground truth in this thesis have been obtained in a virtual setup which mimics the robot and active vision setup. This is primarily down to the fact that plants are easily disturbed and may not return to their original setup, should the plant move, results would not be comparable.

The rest of the chapter is organised as follows. The approach to actively capturing images is discussed, broadly explaining the three phases of the NBV algorithm. The representation of the intermediate model used is defined, providing details of how it is represented and updated. The intermediate image capture algorithm is described, explaining how this is used to create the model for evaluation. This is followed by a discussion on how to identify areas requiring more images, proposing a utility function, and how to capture images for these inadequately scanned areas. Finally, an explanation of the removal of redundant data is provided before evaluating the results of the active camera placement algorithm against more traditional methods.



Figure 4.1 The CT scans for the seven plants used in this research, from left top to bottom. Column 1; Bromeliad (*Vriesea sp.*), *Aloe vera*, Cordyline (*Cordyline sp.*), Brassica (*Brassica napus*). Column 2; Chilli (*Capsicum sp.*), Pumpkin (*Cucurbita pepo*), Sugar Beet (*Beta vulgaris*). X-ray images were obtained using a GE v|tome|x M scanner housed in the University of Nottingham's Hounsfield Facility. Following noise reduction with a median filter, plant material was identified by applying a user-defined threshold to the image.

4.2 Approach

The NBV algorithm proposed here can be broadly classified into three phases;

- **Phase 1** select views that support fast convergence towards an initial (proxy) volumetric representation, which broadly represents the object; this is evaluated in phase 2 to determine quality and amount of coverage (the number of voxels that have been seen).
- **Phase 2** view planning to ensure sufficient coverage of the proxy based on a view evaluation metric which is applied to clusters of voxels such that a sufficient set of images and feature matches across images are available for MVS
- **Phase 3** removal of redundant images from the initial step to reduce unnecessary computational requirements

A more detailed specification of the process is given in Algorithm 4.1 and then discussed in the remainder of this chapter. Algorithm 4.1 requires that the vision cell is calibrated and outputs a series of 2D images and corresponding matrices. A volumetric model is initialised, Algorithm 4.1 Line 1, and the robot is moved to the starting position in line with the turntable, Algorithm 4.1 Line 2. For each level (the position of camera along the vertical axis) a colour filter is applied to the cameras live stream and the robot arm is moved along the x and y-axis until the object is encapsulated in the cameras field of view (FOV), Algorithm 4.1 Line 4, 10 images are captured at the current level rotating the turntable after each image and estimating camera matrices, Algorithm 4.1 Line 5-10. The volumetric model is then updated, Algorithm 4.1 Line 11, and the process is repeated for the next two levels. The next step involves active vision to obtain images of areas of the object that have not been viewed sufficiently. The voxels making up the volumetric model are clustered, and a view sphere is created, Algorithm 4.1 Line 13-15. The clusters are evaluated and if another view is necessary, the view to best see the cluster is determined based on the utility function, Algorithm 4.1 Line 17-21. If the model requires further evaluation, repeat the active vision process, Algorithm 4.1 Line 23. Once the model is sufficiently scanned the volumetric resolution is increased and redundant images are removed, Algorithm 4.1 Line 24-25.

Algorithm 4.1 The NBV algorithm**Input :** *A calibrated active vision cell***Output** *A series of Images, volumetric model, camera calibration*

```

1: Initailise a volumetric model
2: Determine the starting position of the robot
3: for 1 to 3 do
4:   Readjust position of robo
5:   Adjust position of turntable
6:   for 1 to 10 do
7:     Capture image of object
8:     Calculate camera calibration for position
9:     Rotate the turntable
10:   endfor
11:   Update the volumetric model with new information
12: endfor
13: Define a set of points from the volumetirc model
14: Cluster the point set
15: Define view sphere based on size of volumetric model
16: for each cluster
17:   Calculate the score
18:   If the score is less than 1:
19:     Expand over incremental viewsphere until view is found
20:     Update the volumetric model
21:   endif
22: endfor
23: If score is still less than one for any of the clusters, return to 18
24: Remove redundant images
25: Increase the resolution of the volumetric model

```

4.3 Plant Proxy Representation

The intermediate plant model, known as the Plant Proxy Representation (PPR) is represented as a series of voxels, using an octree which is a hierarchical space partitioning data structure which is a tree with a branch factor of 8 (Figure 4.2). Each voxel has a series of attributes, namely; scores based on its visibility, the *images* in which it has been seen, a list of *neighbours*, a *3D coordinate* representing its position in world space and a *label* indicating its type:

- **Seen.** The voxel has been intersected by at least one ray from the images and is part of the surface
- **Unseen.** The voxel has not yet been seen
- **Inside.** The voxel lies within the object and is surrounded by seen

voxels

- **Empty.** The voxel is empty and is not part of the object

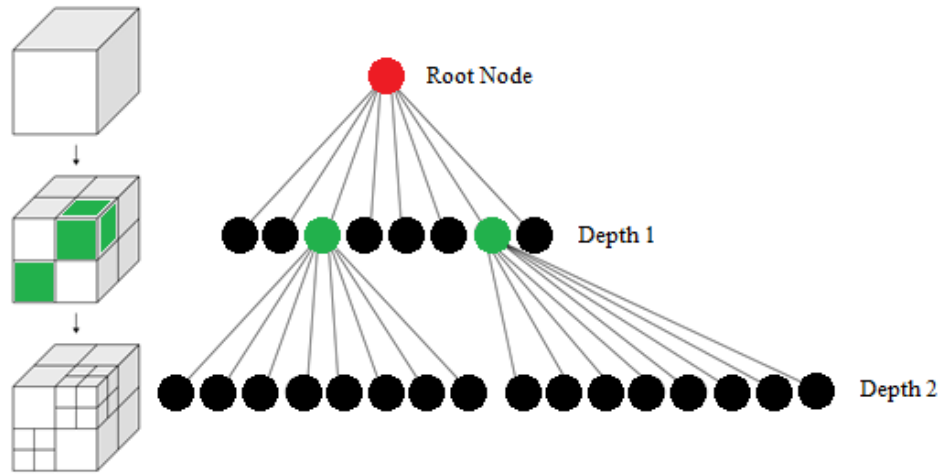


Figure 4.2 Representation of an octree, a hierarchal space partition data structure which is subdivided to increase the knowledge of the object.

The PPR is initialised as a single voxel at the root of the tree. Images are captured and the PPR is updated, increasing knowledge of the object and the depth of the model. The update involves five steps to carve away empty voxels and increase the depth of those that are occupied. Given a new view, these steps are;

1. Determine the camera position in the world with respect to the PPR and create the transformation matrix that transforms voxels in the PPR to 2D image coordinates
2. Apply a Euclidean colour filter to the image to separate the background from foreground and represent the resulting image, a silhouette, as a binary matrix
3. Iterate through the voxels in the PPR that are not marked as empty and determine their 2D coordinates in the binary image using the 3D points, the corners of the voxels, in space and camera matrices, creating a 2D polygon
4. Determine whether the polygon lies on the object; if the corresponding pixel value is 1 then it is part of the object. If the projected polygon consists of part of the object and outside of the object, i.e. a 0-value pixel, split the voxel into 8 further quadrants and repeat, if it does not consist of part of the object, then mark as outside.

5. Update the voxel neighbours and labels to indicate current voxel status

It is important to note that the PPR is unsuitable for the final representation of an object as it is often crude, blocky and is unable to deal with concavities. It does, however, provide a good intermediate representation and allows for quick evaluation and ray tracing, resulting in reduced computational requirements (Vasquez-Gomez, Sucar and Murrieta-Cid, 2013).

The proxy representation has a resolution of 128^3 , which A. Hornung (2008) state to offer the best trade off in terms of accuracy and speed, increasing this to 256^3 for the final stage to aid the 3D modelling process. The following section discusses how the initial image set is selected to create the PPR.

4.4 Phase 1: Initial Image Capture

To construct the PPR a series of images are captured at 3 *levels* (Figure 4.3); where *level* refers to the position of the camera along the z-axis, vertical, and each *level* starts at a different turntable rotation (0, 12, 24 degrees), adding more variety to images.

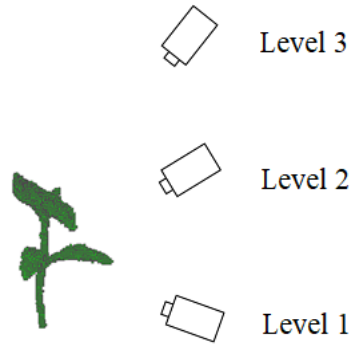


Figure 4.3 Represents the three levels in which data is acquire for the proxy. The lowest camera position is in line with the turntable, the middle camera is the height of the object determined from the proxy where the camera is angled to look at the centre of the object and the top camera is double the height of the object.

The first set, *level 1*, are obtained by positioning the camera in line with the turntable, along the z-axis (the vertical axis), and directed to the centre of the turntable; both of which are known from the calibration stage. The position of the robot can be calculated by rotating the end effector of the manipulator by constructing a rotation vector aimed at the centre of the turntable, defined as;

$$\vec{Rv} = [x, y, z] = [Q_y \times Q_x, Q_z \times Q_x, Q_\theta \times Q_x]$$

(Eq. 4.1)

where Q is an axis-angle $Q = [Q_x, Q_y, Q_z, Q_\theta]$ defined as;

$$\begin{aligned} Q_x &= \frac{n_z - o_y}{\sqrt{(n_z - o_y)^2 + (o_x - m_z)^2 + (m_y - n_x)^2}} \\ Q_y &= \frac{o_x - m_z}{\sqrt{(n_z - o_y)^2 + (o_x - m_z)^2 + (m_y - n_x)^2}} \\ Q_z &= \frac{m_y - n_x}{\sqrt{(n_z - o_y)^2 + (o_x - m_z)^2 + (m_y - n_x)^2}} \\ Q_\theta &= \arccos((m_x + n_y + o_z - 1)^{1/2}) \end{aligned}$$

(Eq. 4.2)

where

$$\begin{aligned} \hat{O} &= [o_x, o_y, o_z]^T = \frac{\vec{c} - \vec{P}}{\|\vec{c} - \vec{P}\|} \\ \hat{M} &= [m_x, m_y, m_z]^T = \frac{\vec{U} \times \vec{Z}}{\|\vec{U} \cdot \vec{Z}\|} \\ \hat{N} &= [n_x, n_y, n_z]^T = \hat{O} \times \hat{M} \end{aligned}$$

(Eq. 4.3)

and $\vec{c}$ is a 3x1 vector representing the 3D position of the turntable as obtained from the setup; $\vec{P}$ is the current kinematics of the robot as a 3x1 vector representing the 3D coordinates in space and; $\vec{U} = [0 \ 0 \ -1]$ is the up vector representing the up direction in world space. This rotation vector is applied to the robot position and one final transformation is applied to the last joint of the robot to ensure the camera is parallel to the ground.

A Euclidean colour filter is applied to the camera's live stream to separate background from foreground. This keeps pixels within a specified range, defined by a radius, where the radius is defined such that the white background is filtered. The radius is obtained from user input, which allows user to easily

see if the background has been filtered out, though a default value is fixed at 120. This is shown in Figure 4.4.

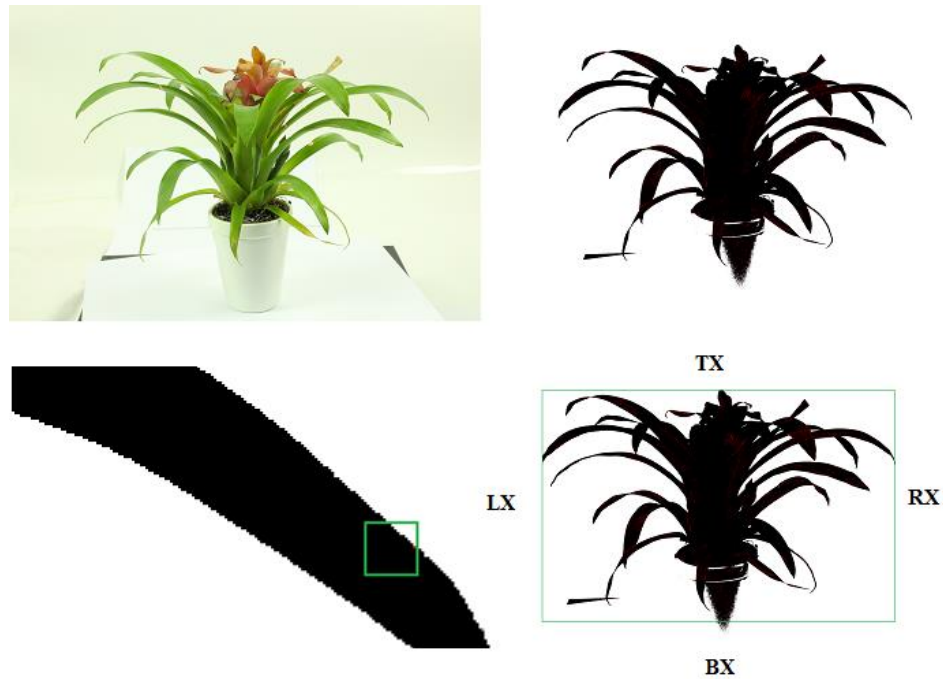


Figure 4.4 Top Left: Original image, Top right: Image separated from background with discarded outliers, Bottom Left: Evaluation of a pixel, Bottom right: The resulting border defining values RX, Lx, Tx, BX

From the resulting silhouette, a series of simple rules are applied to determine whether the plant lies within the cameras' FOV. The camera is moved under the control of a set of heuristic rules until the plant is either fully enclosed by the image boundary but without excess space or the arm is at its maximum reach. The rules employed are intuitive, but effective:

- If there are 50 or more pixels of white space surrounding the plant (TX, LX, BX, RX all > 50) a forward movement is made.
- If the plant region is close to the boundary at either the top and bottom or left and right a backward movement is made.
- Forward and backwards movements are large, 75cm, unless a movement in the opposite direction has just been made, in which case they are small, 15cm. This introduces a degree of fine-tuning and prevents oscillation.
- If LX is large and RX is small, rotate left.

- If RX is large and LX is large, rotate right.
- If TX is large and BX is small, rotate upwards.
- If BX is large and TX is small, rotate downwards.

Once a viewpoint has been determined, a series of images are captured by repeatedly rotating the plant 36 degrees, so that 10 images are captured per level. Space carving (Seitz, 2000) is used to carve the empty voxels, generating the *level 1* PPR. Space carving is performed by applying a Euclidean colour filter to each image, creating a series of silhouettes. These silhouettes are then represented as a binary matrix where 1 represents part of the object and 0 is an outlier (Figure 4.5). By projecting 3D voxels to 2D coordinates, creating a polygon, it is possible to determine whether the voxel is part of the object by comparing to the binary matrix, carving away voxels which are not. The image is represented as a binary matrix to reduce the computational requirements of working directly with images.

Another silhouette can be detected in some instances; for example, the bottom right corner of the centre image in Figure 4.5. This can be due to a number of reasons; for example, an incorrect radius for segmentation, poor or varying illumination, or clutter in the background. To ensure that smaller ‘noisy’ silhouettes do not interfere with the main object, a blob detection method is applied to the camera’s live stream, ensuring that only the plant is considered when moving the camera; detecting the centremost, largest, object.

When modelling, if the plant species is expected to be sufficiently rotationally symmetric, the search for camera locations need only be performed once per level, and the same camera positions can be used at each turntable rotation. However, given a species that may not be rotationally symmetric, a new search may be performed for each turntable rotation.

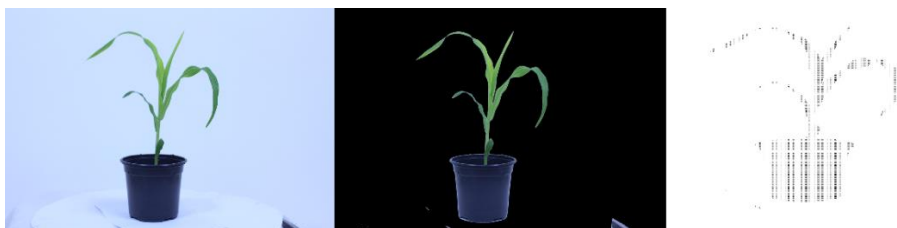


Figure 4.5 left; an original RGB image captured using the cell, centre; the result of the euclidean colour filtering, separating the background and foreground, right; the binary matrix represented as a series of 1s.

To determine the position of the second level to capture images, the object height is estimated from the initial PPR. The camera is moved in line with the top of the plant, and directed towards its centre, which is estimated from the bounding box the current PPR, and the same set of rules are applied to ensure the plant lies within the cameras FOV. Another set of images are captured, updating the PPR to further increase knowledge of the object and produce the *level 2 proxy*. A final set of images are captured at twice the height of the object to gather images for the third level. The PPR is illustrated in Figure 4.6. Three *levels* were chosen to ensure that the object is adequately scanned from different angles and heights such that it can be seen from under, inline and above.

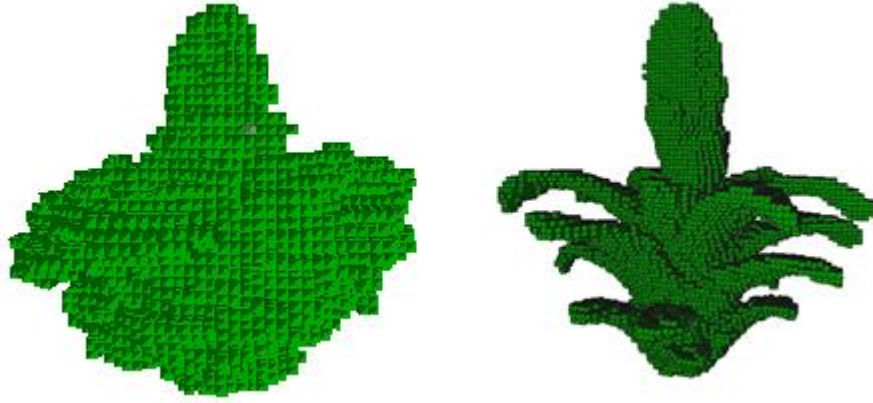


Figure 4.6 Proxy representation; Left, a level 1 PPR, the initial proxy; Right, the final PPR showing more object features after acquiring additional viewpoints

4.5 Phase 2: Image Selection

The final PPR is evaluated to identify inadequately scanned areas of the object that require additional viewpoints. Unlike traditional NBV algorithms that evaluate and determine viewpoints on a per voxel basis, the evaluation in this work is performed on small clusters of voxels. The algorithm can be categorised into three key stages; 1. Clustering of voxels, 2., Evaluation of clusters, and 3., Determination of camera position in world space. The three stages are discussed in more detail below.

Stage 1: Clustering

In the algorithm proposed here (Algorithm 4.1), the computational requirements for image selection are reduced by clustering voxels, as opposed to evaluating

on a per voxel basis. For example, given a model with V voxels, when evaluating on a per voxel basis, n evaluations are required, where n is the number of potential viewpoints. By clustering voxels into small groups of size, S , and applying the incremental view sphere, this can be reduced to $\frac{|V|}{S}$ with the upper bounds remaining significantly less at $\left(\frac{|V|}{S}\right) \cdot n$.

The algorithm for clustering presented here is based on Klasing, Wollherr and Buss (2008) and expanded as in Pound *et al.* (2014). The algorithm aims to partition a series of points into k clusters using the spatial extent of data. A point set is created that maps each voxel in the PPR to a single point, where the centre of the voxel represents the position of the point. Given a set of distinct points originating in 3D space, P is defined as (Eq. 4.4).

$$P = \{\vec{p}_i(x, y, z), i = 1, \dots, n\},$$

(Eq. 4.4)

The clustering method finds the nearest neighbours to a point, p_i , which are within some radius, r , of the centre of the cluster, the starting point. Radius r can be calculated using the density of points as follows (Eq. 4.5) and clusters by selecting the nearest neighbours.

$$r = 2 \frac{1}{n} \sum_{p_j \in P} |\vec{p}_j - \vec{p}_i|$$

(Eq. 4.5)

A KD-tree is implemented; a binary tree for organising points in K -dimensions, 3 in this case. KD-trees are particularly useful for nearest neighbour searches as they can quickly eliminate large portions of data; this is used numerous times throughout this work and during the clustering presented here.

A minimum k_{min} and maximum k_{max} are defined for the cluster size where $k_{max} = \frac{|V|}{5}$ is defined such that the maximum size of a cluster does not exceed 5cm and k_{min} is 3cm. These values are chosen to ensure that at least 3 voxels are in each cluster, so that a triangle can be produced, and the size does not exceed 5cm, so that volume does not become too large, given that it is known that each voxel can be 2mm. A set of clusters are defined such that the points within the cluster are closest to the centre of that cluster than any other. This is illustrated in Figure 4.7; the distance between point X and the four clusters is

determined. The top left cluster minimises the Euclidean distance, consequently X is placed in this cluster.

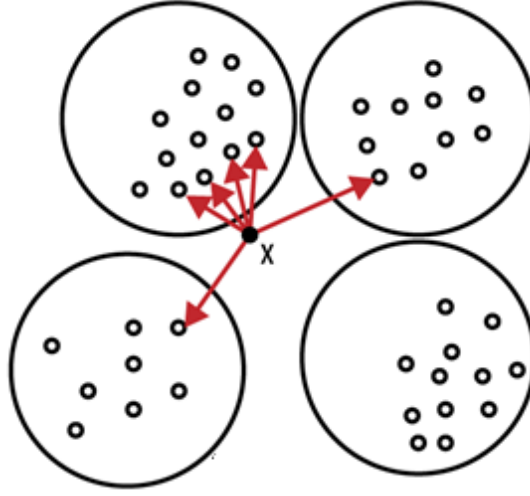


Figure 4.7 Illustration of the voxel clustering algorithm where point X is placed in the top left cluster as the distance is minimal with respect to the other three clusters

The result of Stage 1 is a clustered volumetric model that is used for evaluation in Stage 2.

Stage 2: Cluster Evaluation

Ray tracing is performed to determine visibility for each of the clusters in the PPR. A common method of ray tracing is Uniform Ray Tracing (URT), in which rays equal to the depth of the proxy are projected from the camera into the scene, determining intersection with the PPR. However, this often results in large computational requirements. Hierarchical Ray Tracing (HRT) is therefore used, as opposed to URT. This traces sparse rays through the proxy, only increasing the resolution when voxels are touched by a ray, until the maximum resolution is reached (Figure 4.8). Performing ray tracing in this manner significantly reduces the computational requirements, as dense rays are not used for empty voxels. In the method proposed here, rays are projected from each of the cameras to the clusters to determine the visibility of that cluster to some camera. Should a single voxel within the cluster be seen, the voxels within inherit the same classification.

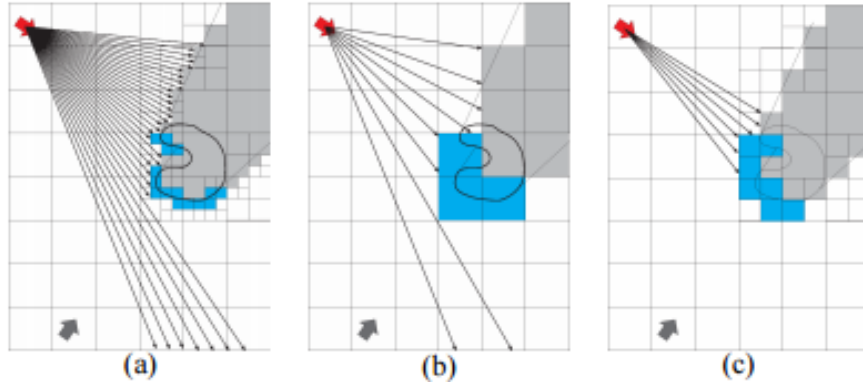


Figure 4.8 Comparison of Uniform Ray Tracing (URT) versus Hierarchical Ray Tracing (HRT). a. the traditional Uniform Ray Tracing (URT) approach, where a set of dense rays are projected from the camera into the scene. b. the first stage of HRT, where a series of sparse rays are projected from the camera. c. the second stage of HRT, only when a voxel is hit does the ray resolution increase (Vasquez-Gomez et al. 2013).

A utility function has been devised (Eq. 4.6), based on the number of times the cluster has been seen and the angles between the cameras that have seen it.

$$f(C) = \frac{1}{|C|} \sum_{j=1}^{|C|} \frac{1}{2} \left(\frac{\text{seen}(C_j)}{\text{imgCrit}} \frac{\text{maxAngle}(C_j, C)}{\text{angleCrit}} \right)$$

(Eq. 4.6)

In Eq. 4.6 $\text{seen}(C_j)$ is the number of times the cluster has been seen by a camera and imgCrit is the number of times a point must be seen to ensure an accurate representation; 3 is used to match the PMVS settings, as PMVS produces more accurate results when using a parameter of 3. $\text{maxAngle}(C_j, C)$ is the maximum angle between any of the cameras, C , that can see the cluster C_j , and angleCrit is the minimal angle difference to ensure different views; 20 degrees is used, to ensure that a sufficient amount of ‘new’ data is seen. While 3 images are optimal, if the angle in which the voxel has been seen is large enough then 2 views becomes sufficient for reconstruction.

Increasing the quality and accuracy of points, achieved by correctly matching features in MVS, ensures that a faithful representation of the object can be constructed. The utility function devised here assists this process by ensuring the object has been adequately scanned; at least 3 times and at varying angles for each cluster. Three views have been chosen as a minimum to match the

requirements of feature matching in PMVS. While a higher value could have been used, in complex objects excessively scanning the object can result in outliers, noisy data and large computational requirements. If the cluster evaluation score is less than 1 it is assumed that the minimal criteria to construct the object have not been met. A score of less than 1 indicates that a voxel has not been seen 3 times and/or that the angle variation between views is not great enough, in which case more images need to be captured. A low evaluation score results in a poor output from MVS which in turn results in an unfaithful reconstruction.

Stage 3: Camera Placement

An additional viewpoint needs to be calculated for each cluster that does not satisfy the utility function. Traditionally in NBV, a view sphere is placed around the object and each potential view is evaluated. In some instances, this means that 10,000+ views need to be considered. In the setup here, for example, even if robot movements are made in whole degrees, making it possible to move 180 positions from top to bottom and 360 positions around the object, this results in a possible 64,800 camera positions. The algorithm proposed here significantly reduces this by determining an optimal position that is defined as the normal of the cluster, obtained through principal component analysis. Such an approach is not feasible when evaluating on a per voxel basis as the normal is unknown. The NBV method proposed here for capturing images of under sampled clusters involves 7 key steps:

Step 1. The size of the view sphere is calculated using camera parameters and object size, ensuring the whole object is visible, (Eq. 4.7)

$$Distance = \frac{\max(w, h, d) \cdot 0.5}{\sin(FOV \cdot 0.5)}$$

$$FOV = 2 \cdot \text{atan}\left(\frac{1}{2} \cdot \frac{s}{f}\right)$$

(Eq. 4.7)

where s is the sensor size, f is the focal length, both of which are known from the camera, and $\max(w, h, d)$ is the dimension of the object and is calculated from the PPR by determining the maximum and minimum position of the PPR for

each axis. The centre of the view sphere, $\overrightarrow{vs}$, is calculated as the centre of the PPR (Eq. 4.8).

$$\overrightarrow{vs} = \begin{bmatrix} vs_x \\ vs_y \\ vs_z \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} \min(V_x) + \max(V_x) \\ \min(V_y) + \max(V_y) \\ \min(V_z) + \max(V_z) \end{bmatrix} \quad (\text{Eq. 4.8})$$

Step 2. To determine the camera position, as an intersection with the view sphere, the normal of the cluster (i.e. the vector perpendicular to the surface) is calculated and traversed for the radius of the view sphere to determine endpoint $p_0 = [x_0 \ y_0 \ z_0]$. The normal is then inverted and traversed to determine the opposite endpoint $p_1 = [x_1 \ y_1 \ z_1]$. Intersection I , the *optimal* camera position, is then calculated as follows (Eq. 4.12).

$$I = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_o + q \cdot d_x \\ y_o + q \cdot d_y \\ z_o + q \cdot d_z \end{bmatrix} \quad (\text{Eq. 4.9})$$

where,

$$\begin{aligned} q &= \frac{-b - \sqrt{b^2 - 4ac}}{2a} \\ a &= d_x^2 + d_y^2 + d_z^2 \\ b &= 2[d_x(x_0 - vs_x) + d_y(y_0 - vs_y) + d_z(z_0 - vs_z)] \\ c &= x_0^2 + vs_x^2 - 2x_0vs_x + y_0^2 + vs_y^2 - 2y_0vs_y \\ &\quad + z_0^2 + vs_z^2 - 2z_0vs_z - r^2 \end{aligned} \quad (\text{Eq. 4.10})$$

and,

$$d = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} x_1 - x_0 \\ y_1 - y_0 \\ z_1 - z_0 \end{bmatrix} \quad (\text{Eq. 4.11})$$

Step 3. The camera position is evaluated for feasibility by converting from world coordinates to robot coordinates and performing inverse kinematics to

ensure the position is reachable. A visibility test is then conducted by performing ray tracing to determine whether the voxel is occluded and therefore not visible from the camera position.

Step 4. If either of these evaluations fail (Step 3), the search area incrementally expands over the view sphere one step at a time (Figure 4.9). Expansion continues outwards from the *optimal* position, the red dot, evaluating view positions until one is found that passes both criteria; first evaluating the green, then yellow and so on. This is repeated for the remaining clusters with evaluation scores of less than 1. Should the potential viewpoint satisfy both evaluations (feasibility and reachability) then the camera position is accepted. The first position to satisfy the evaluations is accepted and further expansion over the view sphere is halted.

Step 5. Once all the camera positions have been determined, merging of potential camera positions is performed to reduce the total number of images and movements required. Merging is performed simply by averaging sets of viewpoints that are close; that is within a fixed angular threshold of 20 degrees, and by performing the two tests for occlusion and reachability; if both pass the camera positions are merged.

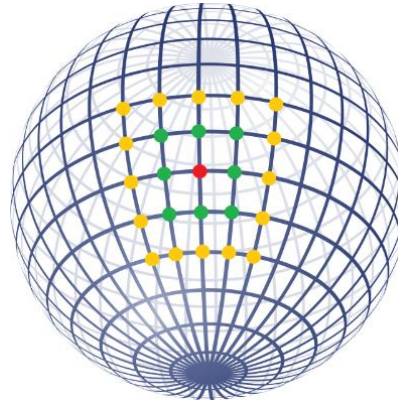


Figure 4.9 The incremental view sphere for determining camera positions. The red dot in the centre illustrates the optimal viewpoint defined by the normal of the cluster, the green is the first increment and the yellow the second. This continues across the entire view sphere

Step 6. To capture the images, the rotation between the robot base and the camera position needs to be calculated, to position the camera correctly with respect to the view sphere. The rotation is performed around the z-axis, given

that it is a flat surface, and can be calculated simply as: $rotation = atan2(c_x r_y - r_x c_y, c_x r_x + c_y r_y)$ where r is the robot base position and c is the camera position. The camera positions are sorted into ascending order according to the rotation of the turntable to improve efficiency, as turntable rotations take significantly longer than robot arm movements.

Step 7. Once data has been captured for all under sampled clusters the proxy is evaluated for the final time. If any clusters fail the evaluation (i.e. there is insufficient data) steps 2 to 6 are repeated, otherwise the algorithm completes.

4.6 Phase 3: Image Removal

The final phase evaluates the images to determine if any can be removed without reducing the quality of our model. It is already known which cameras have seen which voxel, so it can be determined whether a view can be removed using the cluster evaluation method discussed in stage 2. If any image can be removed while ensuring all clusters maintain scores of above 1, it can be assumed that the image can be successfully removed without deteriorating the final model.

4.7 Results

The algorithm proposed here is evaluated against three methods. Namely; *One static*, a single static camera placed at the side of the plant at a fixed distance, this is the common method used in plant phenotyping (Photon Systems Instruments, Drasov, Czech Republic; LemnaTec, Aachen, Germany; Qubit Phenomics, Kingston Ontario, Canada). *Two Static* uses two fixed cameras, one directly in line with the plant, the other placed slightly higher looking down. Finally, *Arbitrary*; a series of random views taken from distinct positions. Figure 4.10 **Error! Reference source not found.** illustrates these three camera placement strategies. Static and arbitrary camera placements are common in phenotyping systems. The results presented here show that intelligently selecting camera positions to capture images provides a greater insight into the object structure and composition by providing a series of points that more faithfully represent the object. The acquisition of these points is reliant on an image set that captures high quality images of the object through a set of carefully selected views.

Performance evaluation is achieved by determining the distance of points generated by MVS to the surface of the X-ray CT scans. The quality of images,

which is the average number of points acquired per image, and how close these points are to the surface are also included in the evaluation.

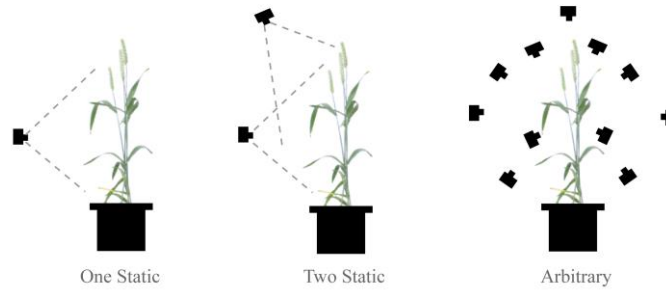


Figure 4.10 An illustration of the different camera setups that are commonly used in plant phenotyping systems. In each case the plant is rotated, or circled by the user, and images are obtained. In the instance of one static a single viewpoint is used around the whole plant.

The results are presented in the remainder of the chapter and are organised into two tables for each plant. The summary of findings presents the mean distance in cm, μ , of the acquired points to the ground truth, computed from the Euclidean distance between the points obtained here and the ground truth surface. The variance, σ^2 , between the dataset consists of the distances to the surface, the standard deviation of the points, σ_X , the total number of points acquired and the average points per image. When using a point cloud to drive surface reconstruction (e.g. Pound et al. 2014), higher numbers of points allow a finer granularity of reconstructed surface patches, and a higher number of points per image indicates that more information can be generated for each image captured. Lower mean and standard deviation values also impact the quality of the surface reconstruction; where lower values indicate a more accurate representation when compared to the ground truth. The AVC improvement table shows how much of an improvement has been made by the algorithm proposed here when compared to each of the approaches, which consists of the same table headings, however, the ‘image’ column represents the percentage of images that are used in comparison to the corresponding approach. A lower number of images reduces the computational requirements for reconstruction (discussed further in Chapters 5 and 6).

Experiment One



Experiment one is conducted on a Bromeliad (*Vriesea sp.*), a member of the Bromeliaceae. Bromeliaceae is a family of monocot flowering plants in which over 3,400 species are known natively to the tropical Americas, and many are commonly found as house plants in the UK. While foliage takes different shapes and forms the one used in this experiment is thin, broad and flat. Consequently, views from above the plant, clearly viewing the wide leaves, will offer the greatest amount of information into the plant size and structure. However, occlusion makes this problematic for static cameras that may be unable to see the underlying leaf surfaces.

Summary of Findings

	μ (cm)	σ^2	σX	Points	Images	Per Image
One Static	0.3574	1.3913	1.1795	141,073	40	3,526.8
Two Static	0.3442	0.5551	0.7450	155,396	40	3,884.9
Arbitrary	0.2693	1.2694	1.1267	227,338	40	5,683.5
AVC	0.1959	0.4587	0.6773	290,236	36	7,637.8

AVC Improvement

	μ	σ^2	σX	Points	Images	Per Image
One Static	45.19%	67.03%	42.58%	105.73%	10%	116.56%
Two Static	43.10%	17.35%	9.09%	86.77%	10%	96.60%
Arbitrary	27.26%	63.86%	39.88%	27.67%	10%	34.39%

Table 4.1 Experiment One: evaluation of the AVC versus traditional image capture methods on a Bromeliad (*Vriesea sp.*)

The results show that the AVC cell significantly out performs the two static methods, obtaining more than 115% more points in the first case, this is due to the way the leaves bend over, making it difficult for static cameras to fully view all leaf surfaces. In comparison to the arbitrary viewpoints, it is possible to increase the points per image by almost 35%, indicating that intelligently selecting viewpoints does indeed improve the results in comparison to random selection. Moreover, the reduction in mean distance to the surface of 27%, shows that a more accurate point cloud is being generated.

Experiment Two



Experiment two was conducted on an *Aloe vera*. The upwardly-oriented leaves occlude plant structures that lie directly behind them making it challenging for views that are side on. Like the bromeliad from experiment one, it consists of flat wide surfaces with little texture

Summary of Findings

	μ (cm)	σ^2	σX	Points	Images	Per Image
One Static	1.4517	13.4128	3.6624	159,870	40	3,996.8
Two Static	1.6911	13.0630	3.6143	160,592	40	4,014.8
Arbitrary	1.8963	20.8614	4.5674	183,027	40	4,575.7
AVC	1.3329	12.9099	3.5930	216,791	31	5,705.0

AVC Improvement

	μ	σ^2	σX	Points	Images	Per Image
One Static	8.19%	3.75%	1.89%	35.60%	22.50%	42.74%
Two Static	21.18%	1.17%	0.59%	34.99%	22.50%	42.10%
Arbitrary	29.71%	38.12%	21.33%	18.45%	22.50%	24.68%

Table 4.2 Experiment Two: evaluation of the AVC versus traditional image capture methods on *Aloe vera*

The AVC algorithm presented here outperforms each of the standard imaging methods, obtaining at least 18% more points while using 22.5% less images. *One static* obtains the fewest points, unable to deal with the concavities caused by the wide upright leaves. *Two static* also has fewer points; despite having two views it is unable to obtain the data occluded by the outer leaves. *Arbitrary* viewpoints do overcome some of the occlusions but does not capture enough information to deal with it completely. The AVC deals with the occlusions and obtains more points and more accurate points with a reduced image set.

Experiment Three



stems.

Experiment three uses a Cordyline (*Cordyline sp.*), a class of approximate 15 species of monocotyledonous flowering plants in the family of Asparagaceae. Unlike the previous two experiments, experiment three focuses on a thin upright plant, which is particularly crowded and occluded towards the base becoming relatively sparse towards the tips of the

Summary of Findings

	μ (cm)	σ_2	σ_X	Points	Images	Per Image
One Static	0.7565	4.0670	2.0167	122,851	40	3,071.3
Two Static	0.8638	7.9084	2.8122	94,193	40	2,354.8
Arbitrary	1.0284	22.6708	4.7614	80,154	40	2,003.9
AVC	0.7384	4.2812	2.0691	143,049	26	3,764.4

AVC Improvement

	μ	σ_2	σ_X	Points	Images	Per Image
One Static	2.39%	-5.27%	-2.60%	16.44%	35.00%	22.57%
Two Static	14.52%	45.87%	26.42%	51.87%	35.00%	59.86%
Arbitrary	28.20%	81.12%	56.54%	78.47%	35.00%	87.86%

Table 4.3 Experiment Two: evaluation of the AVC versus traditional image capture methods on a Cordyline (*Cordyline sp.*)

The AVC approach significantly out performs the *arbitrary* and *two static* views, but unlike other cases has a smaller improvement over the traditional *one static* view. As the plant contains few occlusions and has very thin non-drooping leaves, it is possible to capture a significant amount of information from a side view. However, despite the results between *one static* and the AVC being close, the AVC uses 35% less images (26 relative to 40) than the single view and obtains, on average, 22% more points per image used. This again shows that manipulating the viewpoint using knowledge of the scene can improve the accessibility to data and thus optimise the processing power and time required to generate a 3D model.

Experiment Four



Experiment four is conducted on a Brassica (*Brassica napus*), an agriculturally important member of the Brassicaceae family. Images were captured at seedling stage meaning that it is a very small plant (<10cm). Thus, to avoid missing information, views need to be taken much closer than the previous experiments. A traditional static image

acquisition strategy may struggle if not specifically designed for small plant species, as the camera will be positioned much further away from the plant than desirable. This is a disadvantage as a system using static or arbitrary camera placements must be designed for a specific plant species or growth stage.

Summary of Findings

	μ (cm)	σ^2	σX	Points	Images	Per Image
One Static	0.2007	0.5196	0.7208	97,191	40	2,429.8
Two Static	0.0867	0.1959	0.4427	146,743	40	3,668.6
Arbitrary	0.1682	0.2987	0.5466	178,418	40	4,460.5
AVC	0.0354	0.1530	0.3912	349,311	21	16,633.9

AVC Improvement

	μ	σ^2	σX	Points	Images	Per Image
One Static	82.35%	70.55%	45.73%	259.41%	47.5%	584.58%
Two Static	59.13%	21.91%	11.63%	138.04%	47.5%	353.41%
Arbitrary	78.93%	48.79%	28.44%	95.78%	47.5%	272.92%

Table 4.4 Experiment Four: evaluation of the AVC versus traditional image capture methods on a Brassica (*Brassica napus*)

The AVC significantly captures more data with almost half the images. This demonstrates, as discussed previously, that images in MVS reconstruction do not contribute evenly to the success of a reconstruction, but rather the quality of the images used has a significant effect on results. While *two static* has fewer points than the *arbitrary* method, the mean and standard deviations are both higher than the AVC, which indicates a more accurate reconstruction by the

method presented here. *One static* again does not perform well, as views from sufficient angles are not captured.

Experiment Five



Experiment five is conducted using a Chilli plant (*Capsicum sp.*), which are widely grown in many countries as a cash crop. Similar to experiment four, this plant is at an early growth stage and thus is very small. Static cameras may miss data, particularly as the leaves and stems are thin.

Summary of Findings

	μ (cm)	σ^2	σ_X	Points	Images	Per Image
One Static	0.2380	0.8874	0.9420	113,284	40	2,832.1
Two Static	0.1843	0.1802	0.4245	247,442	40	6,186.1
Arbitrary	0.2536	4.9398	2.2226	199,023	40	4,975.6
AVC	0.1022	0.2101	0.4584	285,381	28	10,192.2

AVC Improvement

	μ	σ^2	σ_X	Points	Images	Per Image
One Static	57.08%	76.32%	51.34%	151.92%	30%	259.88%
Two Static	44.58%	-16.62%	-7.99%	15.33%	30%	64.76%
Arbitrary	59.71%	95.75%	79.38%	43.39%	30%	104.84%

Table 4.5 Experiment Five: evaluation of the AVC versus traditional image capture methods on a Chilli plant (*Capsicum sp.*)

The results show again that the AVC can capture more, and, importantly, more accurate data points from fewer images as opposed to traditional methods. However, the *two static* camera approach does have a lower variance and standard deviation due to the shape and structure of the plant, but this is only achieved with more images (40 relative to 28). While the mean is the most important factor in determining accuracy, the standard deviation and variance provide an insight as to whether outliers exist. Overall the reconstruction obtained by the AVC still produces a more accurate model. It is also worth noting that the AVC cell is later merged with the volumetric model to remove outliers (Chapter 5).

Experiment Six



Experiment Six is conducted using a Pumpkin plant (*Cucurbita pepo*). The large flat leaves make occlusions for data acquisition a major problem, with the leaves often blocking the stem. Moreover, flat surfaces of plants are often problematic to reconstruct due to a lack of texture.

Summary of Findings

	μ (cm)	σ^2	σX	Points	Images	Per Image
One Static	1.1220	3.4872	1.8674	715,222	40	17,880.6
Two Static	1.2104	12.0569	3.4723	517,039	40	12,926.0
Arbitrary	0.6982	3.3126	1.8200	852,426	40	21,310.7
AVC	0.3588	1.9108	1.3823	1,048,576	30	34,952.5

AVC Improvement

	μ	σ^2	σX	Points	Images	Per Image
One Static	68.03%	45.20%	25.98%	46.61%	25%	95.48%
Two Static	70.36%	84.15%	60.19%	102.80%	25%	170.41%
Arbitrary	48.62%	42.32%	24.05%	23.01%	25%	64.01%

Table 4.6 Experiment Six: evaluation of the AVC versus traditional image capture methods on a Pumpkin (*Cucurbita pepo*)

The large surface areas presented in this species explain why such a high number of points have been produced for this model relative to the other experimental plants. This plant has a large surface area, which often results in large computational requirements during surface reconstruction and, because the large surface area contains minimal texture, the variance and standard deviation for all methods is greater than for previous plants. This is due to the difficulties associated with feature matching in PMVS. The AVC can improve all traditional methods with 25% less images whilst also obtaining a more accurate set of points.

Experiment Seven



Experiment Seven was conducted on Sugar beet (*Beta vulgaris*). The sugar beet is another example of a plant that has a large surface area, though differs from the pumpkin plant due to the long thin stems. Furthermore, the long, drooping, wide

leaves make this object difficult to acquire data for and, with little texture on the leaves, feature matching becomes problematic.

Summary of Findings

	μ (cm)	σ^2	σX	Points	Images	Per Image
One Static	0.3142	0.5984	0.7736	217,369	40	5,434.2
Two Static	0.3071	0.5844	0.7644	190,676	40	4,766.9
Arbitrary	0.2471	0.6141	0.7837	306,103	40	7,652.6
AVC	0.0639	0.1417	0.3764	576,633	30	19,221.1

AVC Improvement

	μ	σ^2	σX	Points	Images	Per Image
One Static	79.67%	76.32%	51.34%	165.28%	25%	253.70%
Two Static	79.20%	75.75%	50.76%	202.42%	25%	303.22%
Arbitrary	74.15%	76.93%	51.96%	88.38%	25%	151.17%

Table 4.7 Experiment Seven: evaluation of the AVC versus traditional image capture methods on Sugar beet (*Beta vulgaris*)

Intelligently selecting the images produces higher quality data, with fewer images, and has a minimum of 151% more points per image than any of the traditional methods. The *single static* camera and *two static* cameras perform almost equally as well, while the *arbitrary*, on this occasion, out performs both other traditional methods with respect to mean and number of points per image. Again, the AVC method uses less images than traditional methods to produce a more accurate model.

4.8 Discussion

An AVC for capturing 2D images in a controlled environment has been proposed. The algorithm is evaluated on complex objects and compared to more traditional methods of arbitrary camera positions and static cameras in terms of both the amount of data produced and the quality of the points.

In each case AVC produced more, and more accurate points with a reduced image set. While static camera placement can work, success depending on the structure of the target plant, it is certainly not the most efficient method. By employing active vision, the amount of data obtained can be significantly increased: the AVC can adapt to objects of different shapes and sizes with varying levels of occlusions and complexity. However, one potential constraint is the limit on the maximum size of the target object, which is based on the reach of the robot; though this can be overcome simply by employing a larger robot, e.g. the UR10. Alternatively, multiple-arm systems could be designed, which would have the added advantage of removing the need for the turntable and so reducing plant movement during the image capture process. Static approaches often have cameras fixed into the environment making it extremely difficult and costly to adjust to different object sizes which is crucial in plant phenotyping when trying to obtain growth information, or when creating a system capable of measuring multiple different species and growth stages.

The approach described here requires minimal user input; can be applied to any plant type or structure, with the only limitation on size being the reach of the robot arm. It is more accurate and requires fewer images than previous, static imaging approaches (Tables 4.1-4.7) and offers more flexibility than existing large-scale phenotyping systems by adapting to the natural variation of individual plants. Existing state of the art for plant imaging relies on static cameras - such as static one - for obtaining information and does not adapt for plant species or age. The method is automatic with user input limited to changing the plant and is relatively quick with image capture and analysis relative to other methods, taking minutes as opposed to hours. Moreover, the method has reduced set up and running costs compared to some phenotyping systems such as X-ray CT scanning.

Errors in both the AVC and static methods discussed here are most common around thin areas of the plant, in which size is over estimated. Over estimation of plant area occurs as leaves and stems are thin (leaf thickness can be as little 1mm) and thus even a small error in calibration can result in increased leaf thickness.

As discussed, traditional NBV typically relies on evaluating individual viewpoints on a per voxel basis. Clustering and incrementally expanding the view sphere is used to improve efficiency by reducing the search space and there is no reason why it could not be employed in approaches that use lasers or depth information. Evaluation of views incrementally ensures that fewer computations are needed compared to traditional NBV methods that evaluate every potential viewpoint. Moreover, the NBV algorithm here is not constrained to any specific angle and can be positioned anywhere on the incremental view sphere.

We discuss the future direction for our AVC in the final chapter, Conclusion and Future Work (Chapter 7).

Chapter 5

Point Cloud Processing

This chapter describes the first stage of the reconstruction algorithm proposed in this work; the simplification and clustering of 3D data points. The chapter is organised into five primary sections. Section one introduces the work and provides a summary of the contributions made and algorithm proposed. Section two presents a novel model merging process, with results to show how data can be significantly reduced while maintaining the integrity of the object. In section three, a novel clustering method is discussed, which is required to partition the data into semantically meaningful clusters. Section four explains how points are reclassified (moved from one cluster to another) accordingly. Finally, section five evaluates the proposed clustering algorithm against the state-of-the-art, showing that it can improve the results over traditional methods on the data considered here, while significantly reducing the computational requirements.

5.1 Introduction

The next best view (NBV) algorithm discussed above (Chapter 4) ensures that the target object is adequately scanned. However, it does not prevent difficulties such as noise, outliers and missing data, all of which are particularly common for complex objects. This is due to incorrect feature matching in multi view stereo (MVS), caused by a lack of texture or differences in object features. The model merging algorithm proposed here eliminates many of these difficulties and is novel in that it combines volumetric data with point cloud data to create a new set of points, immediately dealing with some of the more challenging aspects of 3D modelling. Following the merging of datasets, adaptive clustering is performed to segment points into meaningful groups in preparation for surface reconstruction (Chapter 6).

The use of the active vision cell (AVC) provides numerous data, namely; a volumetric model (the Plant Proxy Representation- PPR), a 3D point set generated by Patch-Based Multi-View Stereo (PMVS), a series of camera calibration matrices and a set of 2D images. It is important to note that the 2D images are not required for clustering or model merging but are used at a later stage for 3D modelling. There exist a number of problems with using each of these individual datasets. The point set often contains outliers. The PPR contains voxels that cannot be verified, that is, it is unknown as to whether they exist in the model. The 2D data is not suitable for plant phenotyping on its own as depth is not known, consequently limiting the ability to measure plant traits.

A method to improve the accuracy of data by merging the PPR and point set is proposed. While this is in practical terms a trivial task, it can significantly reduce the computational requirements of 3D modelling, e.g. reducing a complex 3D point cloud consisting of 300,000+ points, to a set of 3,000 points which more accurately represent the 3D object. Figure 5.1 illustrates the two starting datasets, point cloud and PPR, and the resulting merged model, from which the outliers have successfully been removed, yet the integrity of the object remains.

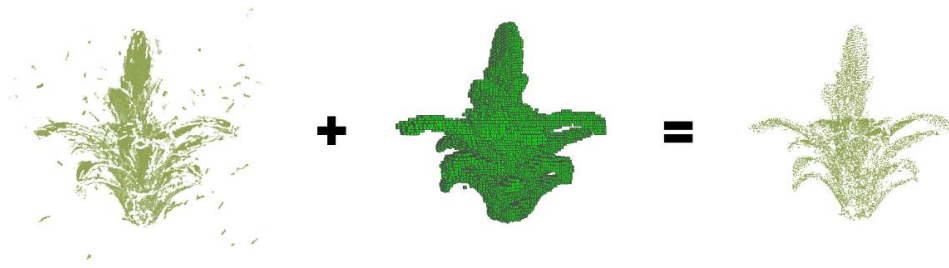


Figure 5.1 Left; the original point cloud data consisting of outliers. Centre; the PPR representation containing unverified voxels. Right; the merged model with the removal of outliers and voxels.

The aim of the merging and clustering algorithm is to produce a more accurate point set and reduced number of outliers, ensuring that the object shape is faithfully represented and that the clustering of points into groups that satisfy a number of constraints (discussed below) is possible. The algorithm for merging and clustering is summarised below (Algorithm 5.1) and discussed in more detail for the remainder of the chapter.

Algorithm 5.1 Data Merging and Clustering

***Input :** A list P all points, V a volumetric model*

***Output :** A list, C , of clustered points*

```

1:  for each point in  $P$  do
2:    for each voxel in  $V$ 
3:      If the point lies  $p$  lies within voxel  $v$ 
4:        add the point to the voxel
5:    end
6:  end
7:  Remove any voxels that have zero points assigned
8:  Define a new point set removing those that do no lie in a voxel
9:  while (true)
10:   for each cluster
11:    if the criteria has not been met
12:      Split the cluster into two
13:      Remove original cluster from set of clusters
14:    if the criteria has been met for all clusters
15:      return false
16:  end

```

Algorithm 5.1 Line 3 determines whether some point is contained within a voxel and if it is, adds the point to voxel. Algorithm 5.1 Line 8 determines the new point set (obtained from merging volumetric and point data), which can return one of three representations, discussed later. This modified point set is used for

the remainder of the work. Algorithm 5.1 Line 11 returns true if the cluster satisfies the constraints for a cluster to be correct (discussed in Section 5.3) and false otherwise. If the criteria function returns false the cluster is split, Algorithm 5.1 Line 12, which returns two new clusters and appends them to the end of the cluster list. Once all clusters meet the criteria, Algorithm 5.1 Line 14, clustering is complete.

The novel contributions of this chapter are as follows. A model merging algorithm (Section 5.2), which produces a more accurate set of 3D points by merging data from both volumetric and point data; a hybrid clustering algorithm (Section 5.3), comprised of a novel clustering method based on PCA and further dimensionality reduction and; an adapted version of the automatic divisive hierarchical clustering (DIVFRP) algorithm, overcoming the difficulties associated with it. Clustering is a crucial element in this work as the surface reconstruction (i.e. triangulation) occurs in 2D. Consequently, poorly clustered points result in a poor representation of the model, losing curvature or connecting individual and separate leaf components and making the surface area of the object seem much larger than it is. Distinct object segments can be triangulated individually when clustered ‘correctly’, eliminating these errors and producing a more accurate reconstruction.

5.2 Phase 1: Model Merging

The model merging phase takes the point cloud and the volumetric model and combines the two to produce a reduced, yet more faithful, point set to represent the object of interest, as a basis for clustering and surface reconstruction.

To merge the representations, a function is devised, (Eq. 5.1), to ensure that some point $\vec{p}$ lies within some voxel v . Here, $\vec{p}_i \in P$ is a vector representation of the position and $v_i \in V$ are the voxels, in which the minimum and maximum position for each axis is known.

$$\begin{aligned} f(\vec{p}, v) = & (p_x \geq \min(v)_x \wedge p_x \leq \max(v)_x) \\ & \wedge (p_y \geq \min(v)_y \wedge p_y \leq \max(v)_y) \\ & \wedge (p_z \geq \min(v)_z \wedge p_z \leq \max(v)_z) \end{aligned}$$

(Eq. 5.1)

Each point in the point cloud is evaluated and if the point lies within one of the model voxels it is added to the voxel, if a point does not lie within a model voxel it is marked as an outlier and is discarded, as the PPR represents the maximum shape of the object. Any voxel that consists of more points than an automatically calculated value, w , is automatically accepted and is considered part of the object (Eq. 5.2).

$$w = \left(\frac{|P|}{|V|} \right)^\sigma$$

(Eq. 5.2)

where σ is between 0.1 and 0.5; it was found that 0.2 works as a good benchmark across all datasets used here. If σ is greater than 0.5 few, if any, points are automatically accepted. A voxel with less than w points is evaluated in two ways to determine whether it is part of the object. First, it is determined whether adjacent voxels contain points. Secondly, if they do, a colour consistency measure is applied to check if the points inside the voxels are within the colour range to the neighbouring voxels points. If they are the voxel and the internal points are accepted, otherwise they are discarded. If the voxel has no neighbours containing points it is classified as an outlier and discarded.

The merged model produces a point set which is used for the basis of cluster analysis and surface reconstruction and can be represented in four ways; **1.** Use the original unchanged point set. **2.** Add every point from the point set that lies within a voxel; this removes outliers, though it does not significantly reduce the complexity of the data as hundreds of thousands of points remain. **3.** Place a single point in the centre of each voxel; this ensures a uniform point set but potentially changes the shape of thin objects. **4.** The average position of all points in a voxel determine where a single point within that voxel will lie. These four approaches are illustrated in Figure 5.2.

Each of the four representations are evaluated and numbered as follows for simplicity in the upcoming results tables; **(1) original** **(2) all** acceptable points model, **(3) centres** model **(4) average** model. The resulting point clouds are evaluated against the ground truth, the X-ray CT scans, which have been acquired and discussed in the previous chapter. The evaluation is performed using the Euclidean distance from points to the surface of the ground truth to evaluate accuracy. The mean distance, in cm, μ , of points to the surface, the

variance in points, σX , the standard deviation σ^2 , and finally, the total number of points, P , in each model, are all considered. A lower mean value (i.e. a smaller distance to the ground truth) results in a more accurate point cloud, which is of utmost importance if the object is to be faithfully represented. The variance and standard deviation show the spread of the data, higher values represent a greater difference in distance to the ground truth, which illustrates that the data is less consistent. Finally, a smaller, but more accurate, set of points can significantly reduce the overhead of clustering and surface reconstruction. Results are analysed on three different plants; a Bromeliad (*Vriesea sp.*), *Aloe vera* and a Cordyline (*Cordyline sp.*), equivalent to experiments 1-3, Chapter 4.

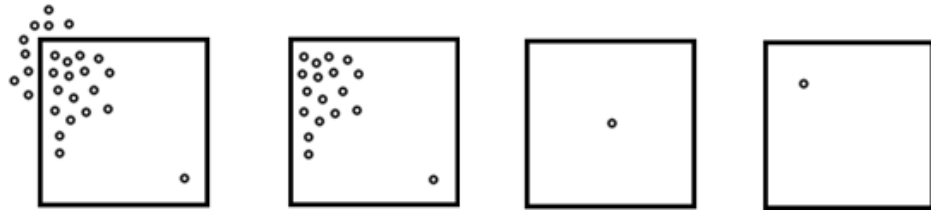


Figure 5.2 The four representations. From left to right; 1. the original points including outliers outside of the voxel, 2. The voxel centre, 3. The all within method 4. The average method

Experiment One

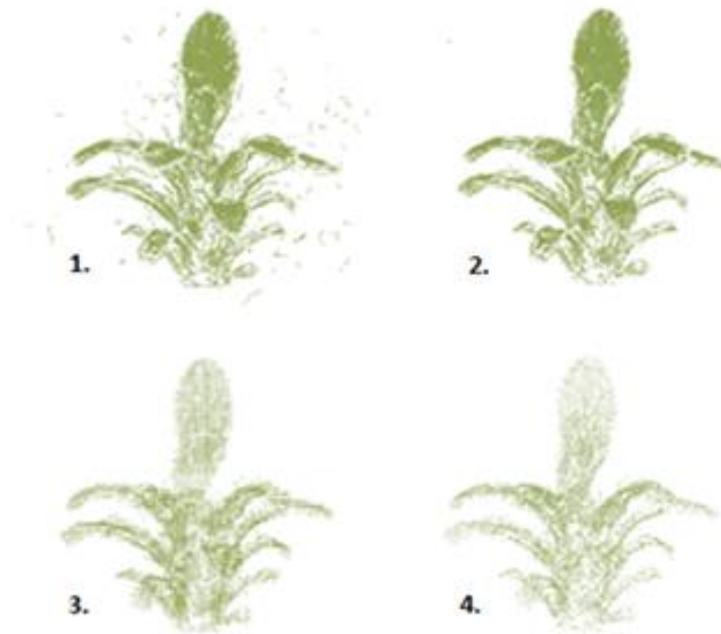


Figure 5.3 Model merging results on a Bromeliad (*Vriesea sp.*). From left to right; 1. The original data, 2. The all within representation, 3. The centres representation and 4. The average representation

Experiment One

	μ	σX	$\sigma 2$	P
Original (1)	0.212	0.962	0.925	265,782
All (2)	0.103	0.325	0.106	235,254
Centres (3)	0.153	0.392	0.154	11,545
Average (4)	0.133	0.387	0.149	11,545

Table 5.1 Comparison of merging results for Experiment One: a Bromeliad (*Vriesea sp.*)

From the first experiment, it can be seen than for all merging approaches, (2), (3) and (4), there is a reduction in the mean distance of the points to the ground truth in comparison to the original data, (1). Representation (2) reduces the mean more than any other method, though still contains a significantly large number of points. While unclear from 2D images, the representation from method (3) (Figure 5.3) increases the size of the object as points are placed centrally. The

final method (4), significantly reduces the number of points and the mean in comparison to the original data (1).

Experiment Two

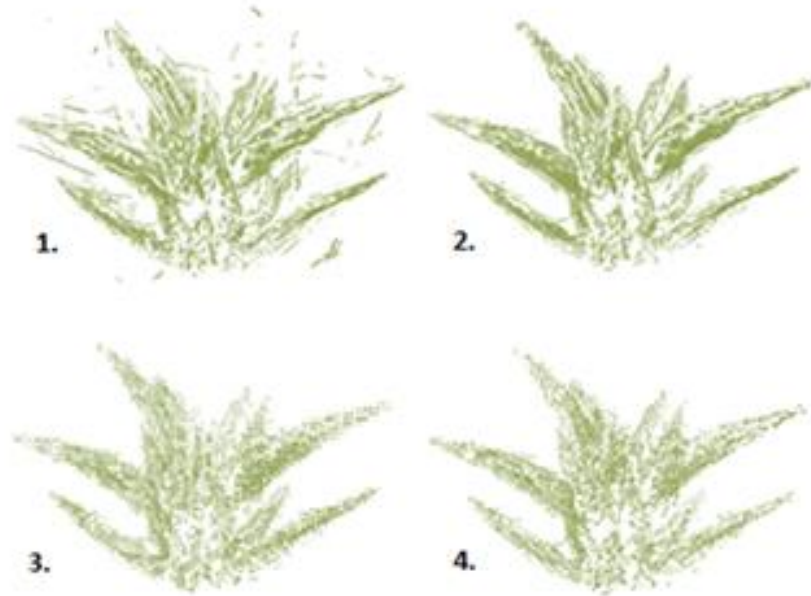


Figure 5.4 Model merging results for *Aloe vera*. From left to right; 1. The original data, 2. The all within representation, 3. The centres representation and 4. The average representation

Experiment Two

	μ	σ_X	σ^2	P
Original (1)	1.437	4.107	16.869	217,379
All (2)	0.571	1.119	1.251	181,249
Centres (3)	0.772	1.324	1.754	11,094
Average (4)	0.716	1.306	1.705	11,094

Table 5.2 Comparison of merging results for Experiment Two: *Aloe vera*

The second experiment shows a similar result to the first. The original point set (1) consists of a highly noisy set of points. This can be seen in the variance between (1) and the other methods (2-4). In the model merging approaches proposed here, the noise around the original point set (1) is discarded and the mean distance for all merging representations is reduced by 50% or more. Like the previous experiment, the average method (4) significantly reduces the

number of points while maintaining plant structure and reducing the mean. The ‘all’ method (2) has the lowest mean, but, as expected, has a significantly large number of points. Once more, the averages method (4) reduces the mean and significantly reduces the size of the dataset.

Experiment Three



Figure 5.5 Model merging results on a Cordyline (*Cordyline sp.*). From left to right; 1. The original data, 2. The all within representation, 3. The centres representation and 4. The average representation

Experiment Three

	μ	σ_X	σ^2	P
Original (1)	0.738	2.069	4.281	143,050
All (2)	0.392	0.860	0.740	93,453
Centres (3)	0.527	1.091	1.191	4,560
Average (4)	0.472	0.979	0.958	4,560

Table 5.3 Comparison of merging results for Experiment Three: a Cordyline (*Cordyline sp.*)

Experiment three shows how powerful the model merging algorithm proposed here can be. Matching features for this plant was incredibly challenging due to the similarities between leaves and lack of texture. Consequently, as seen in Figure 5.5, there is a large amount of noise and the plant components are incredibly difficult to identify. This illustrates how merging the models allows erroneous points to be removed and, in this instance, produces a significantly clearer representation. Like the previous experiments, method (2) has the lowest mean, but still produces a large dataset. Method (4) significantly reduces the size of dataset (i.e. the number of points) and reduces the mean and standard deviation in comparison to method (1). The results show that the model merging algorithm improves the accuracy of points in all instances and can deal with challenging and largely erroneous data.

There are numerous advantages to model merging, as opposed to using either the point set or PPR independently. Namely; the ability to acquire connectivity information from the PPR (which is particularly useful in the surface reconstruction – Chapter 6) where such information is not available in the point cloud, along with normals, and more accurate colour estimates which can be obtained from the point cloud. Though the biggest advantage is the ability to significantly reduce the size of the point set whilst maintaining the integrity and complexity of the object.

Method (2), all points within, does have slightly higher accuracy in comparison to method (4), however, the small difference in accuracy between the two methods comes with significantly more computational requirements. The all within method (2) has at least ten times more points than either of the other merging methods (3 and 4). Therefore, in this work, the average method (4) is used as the point set for surface reconstruction (Chapter 6). Given this new point set representation, a method to efficiently cluster the data needs to be applied before surface reconstruction can take place. Figure 5.6 illustrates the results when applied to a wheat plant, however, as no ground truth is available due to the size of plant (it can not fit in the CT scanner) no further results are presented for this model.

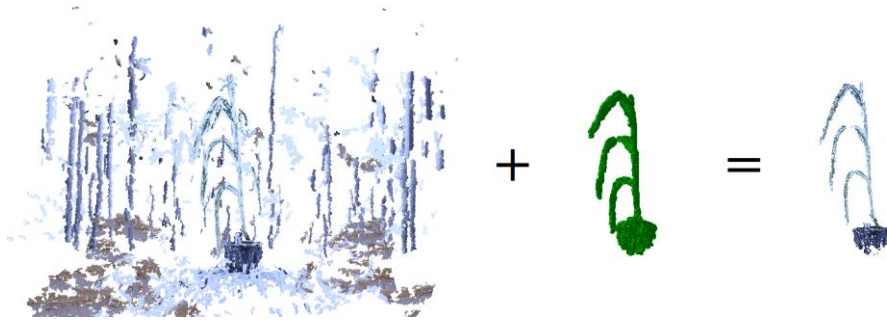


Figure 5.6 Illustration of how a noisy point set of a wheat plant (left) can be merged with a volumetric model (middle) to produce a more accurate point set (right)

5.3 Phase 2: Cluster Analysis

For surface reconstruction, Delaunay triangulation is performed in two-dimensions. Consequently, it is imperative that points are clustered such that they have minimal deviation from the orthogonal plane to prevent long triangles, the loss of curvature and the incorrect triangulation of separate components i.e. two leaves. Numerous constraints need to be considered when clustering that directly impact the quality of 3D modelling. These are; **1.** The *normal constraint*, which specifies that a cluster must have minimum deviation from the plane. If this constraint is not satisfied, large triangles form across the curve of the model. **2.** The *separation constraint*, enforces the rule that no single cluster can contain two distinct groups of point i.e. a cluster that consists of two distinct objects **3.** The *outlier and incorrect patch constraint* specifies that all points must be closer to the centroid of the cluster they are assigned to, than any other cluster in the set. These constraints are illustrated in Figure 5.7.

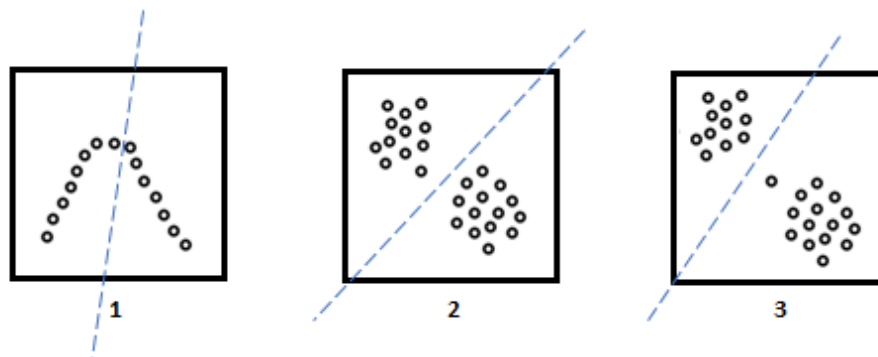


Figure 5.7 Illustration of the main constraints that need to be satisfied; from left to right, 1. the normal constraint, 2. the separation constraint, and 3. the outlier constraint.

Many existing approaches were implemented and tested for means of clustering in this work, however they were not suitable for thin complex objects. For example, abstracting the skeleton of the model using a force field approach was tested (Bucksch and Lindenberg, 2008). However, as some leaves are exceptionally thin, they contain areas in which no internal voxels were present, even at a much greater volumetric depth (i.e. much smaller voxels), and as a result the force field pushes in an incorrect direction, reducing the size of the object incorrectly. Volumetric shrinking was also applied, though, the same problem occurred, producing inaccurate results. Iterative Principal Direction Divisive Partitioning (Tasoulis and Tasoulis, 2008) and Principal Direction Divisive Partitioning (Boley, 1998) methods produced small clusters, with minimal deviation from the plane, however they were unable to overcome difficulties where two branch segments connect, forming a v like shape. Spectral clustering (Reddy, 2016) was able to successfully cluster the point set in most cases, though large computational requirements meant that this was not a feasible option. Note: success in this instance is defined such that the clusters are semantically meaningful. Simple methods, such as K-Means, produced small clusters but were invalid as the clusters were meaningless and potentially contained errors.

Histograms obtained through PCA are a possible solution to split a series of points into two distinct clusters. However, there are numerous difficulties with this approach. For example, methods that perform splitting in this manner typically require that there is some zero-value point (or a clear minimum) index of the histogram, surrounded by non-zero values, illustrating a clear partition of points. If there is no zero-value, then the difference between index i and $i + / - 1$ must exceed some predefined threshold. In complex 3D models it is unlikely that a zero-valued index will occur and defining a threshold can be difficult and often results in incorrectly clustered points.

To overcome these constraints, a hybrid clustering algorithm has been devised, manipulating PCA and 1D data representations. The algorithm proposed here involves three key steps to satisfy the constraints:

1. Partition clusters based on the normal, using PCA to determine deviation from the plane – satisfying the *normal constraint*.
2. Partition clusters if there are two distinct groups, using a modified DIVFRP to determine distinction – satisfying the *separation constraint*.

3. Reclassify clusters using the nearest centroid and points in the proximity – overcoming the *outlier or incorrect patch constraint*.

These three steps significantly influence the outcome of the number and quality of clusters. Moreover, a poor termination criterion results in either excessive clusters or too few. In this work, only when all the constraints have been satisfied, is the termination criterion met. The clustering algorithm is summarised below (Algorithm 5.2) and discussed in more detail for the remainder of the chapter.

The clustering algorithm (Algorithm 5.2) takes a set of points, P , as input and outputs a series of clusters grouping points that are similar based on the stated constraints. The clustering process repeats, Algorithm 5.2 Line 2, until the constraints have been satisfied. Principal component analysis (PCA) is performed for each cluster, Algorithm 5.2 Line 3, to determine orthogonal regression planes. The normal constraint for the cluster is then evaluated and if it has not been satisfied, cluster splitting occurs, Algorithm 5.2 Line 4-5, the algorithm then returns to the start, Algorithm 5.2 Line 2. The same process repeats, although this time, if the normal constraint is satisfied, process to the separation constraint, Algorithm 5.2 Line 6-7, and repeat. Finally, if the reclassification constraint is not satisfied then reclassify the cluster, Algorithm 5.2 Line 8-9. Once the constraints have been satisfied, Algorithm 5.2 Line 11, the clustering algorithm is terminated, Algorithm 5.2 Line 12.

Algorithm 5.2 Cluster Analysis

Input : *A set of points, P*

Output : *A list of clusters*

- 1: *Intialise points as a single cluster*
 - 2: *while the constraints have not been met*
 - 3: *Perform PCA on each cluster*
 - 4: *if the normal constraint has not been met*
 - 5: *split the clusters using based on their normal*
 - 6: *else if the separation constraint has not been satisfied*
 - 7: *split the cluster according the separation algorithm*
 - 8: *else if the reclassification constraint has not been satisfied*
 - 9: *split the cluster according the reclassification algorithm*
 - 10: *else*
 - 11: *all constraints have been met*
 - 12: *end*
-

Clustering is performed in a top-down fashion, starting with the initial point set, which is categorised as the single starting cluster. The problem is defined such that given a set of distinct points, originating in 3D space, where P lies on or close to an unknown surface, we cluster such points within each cluster are closer (Euclidean distance) and more similar to each other, than in any other cluster.

Clusters are subdivided into a series of smaller more precise clusters until there are no improving clusters. A non-improving cluster is a cluster that meets the necessary criterion in terms of *curvature*: ensuring the clusters deviation from the plane is minimal; *distinct partitions*: ensuring that the cluster does not contain two distinct object parts; *outliers*: ensuring that the model contains minimal outliers. The three stages of the algorithm are now discussed in more detail.

Step 1: Normal

PCA is utilised to establish orthogonal regression planes for each cluster from which the Euclidean distance between the original points in the cluster and the orthogonal plane can be calculated. From this, the variance in data can be determined. PCA is performed to obtain the eigenvalues for each cluster where the covariance matrix, CO , is defined in (Eq. 5.3)

$$CO = \frac{1}{1 - n} (M^T M) \quad (\text{Eq. 5.3})$$

where the $3 \times c_i^n$ matrix, M , is

$$M = \begin{pmatrix} x_1 - \bar{x} & y_1 - \bar{y} & z_1 - \bar{z} \\ \vdots & \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} & z_n - \bar{z} \end{pmatrix} \quad (\text{Eq. 5.4})$$

and centre the centre is calculated as

$$(\bar{x}, \bar{y}, \bar{z}) = \left(\frac{1}{n} \sum_i^{c_i^n} x_i, \frac{1}{n} \sum_i^{c_i^n} y_i, \frac{1}{n} \sum_i^{c_i^n} z_i \right)$$

(Eq. 5.5)

CO , (Eq. 5.3), can be factorised as the SVD, $CO = U\Sigma V^T$, where U is an $m \times m$ unitary matrix, where the columns are known as the left singular values of CO ; Σ is a diagonal matrix where the diagonals are singular values, namely eigenvalues $\lambda_{max}, \lambda_{mid}, \lambda_{min}$ when ordered by decreasing value and; V^T is the transpose of V , where columns correspond to the right singular values of CO . The eigenvalues can be used to determine the size of the plane where λ_{max} is the width; λ_{mid} the length and; λ_{min} the deviation of points from the normal of the plane, also known as the height (Figure 5.8).

The deviation of points from the plane, obtained from PCA, (with respect to the Euclidean distance) determines whether the point set is flat, or close to flat, by evaluating λ_{mid} . As the size of the voxel in real world measurements is known, it is possible to determine the maximum physical distance that lies between two points in neighbouring voxels; as an accepted voxel contains exactly one point. Based on this, an evaluation function is used that marks a cluster as acceptable, that is one with little or no curvature. A λ_{mid} value closer to zero shows that points are nearly coplanar; if $\lambda_{mid} < VoxelSize^2$ then the cluster is accepted; if $\lambda_{mid} > VoxelSize^2$ then the cluster must be split. The voxel size is used as a threshold, as the resolution of the octree increases or decreases with the size of the object, changing the size of the corresponding voxels. As a result, the evaluation function is variable, adapting to the size of the model.

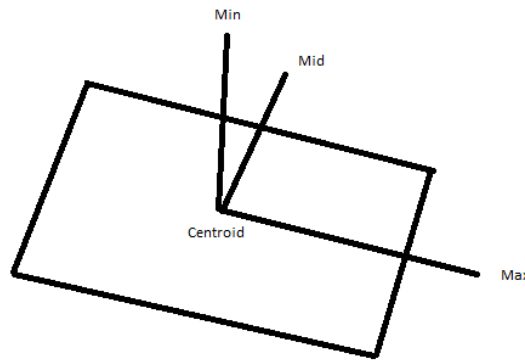


Figure 5.8 PCA eigenvalue decomposition illustrating the *min*, *mid* and *max* eigenvalues

Eigenvector approaches to clustering typically split through the middle of the plane or the largest gap between the points when represented as a histogram, i.e.

where there is a clear distinction between points (Wold, Esbensen and Geladi, 1987; Boley, 1998; Tasoulis and Tasoulis, 2008). While the latter of these methods works when there are two distinct clusters, it does not work well on 3D objects in which gaps between points are uncommon, as they form fully connected objects. The algorithm proposed here is novel; determining the position to split based on the differences of angles in 1D space.

PCA is reduced to a 1D array by multiplying the position of the point with the eigenvectors: that is the point is projected onto the eigenvector with the largest eigenvalue, creating an index for each point along λ_{max} , where each index may contain multiple points which are averaged to produce a single value. This is defined in (Eq. 5.6).

$$i = \lambda_{max} \cdot \left(\left(\frac{\sum_{k=1}^{|C_j|} C_j^k}{|C_j|} \right) - p_i \right)$$

(Eq. 5.6)

where i corresponds to the position of the point in the 1D array of objects and the object has a list of points, a normal and an ID. Each index has a normal defined as the average normal of each point for that index. The normal of the object with index i is defined as the average normal of all points within objected referenced by index.

To determine a potential position to split the point set, an angle is calculated for each element in the array. The angle is the deviation between the current normal and the average sum of normals of values to the left and right of the current index. Figure 5.9 illustrates this process.

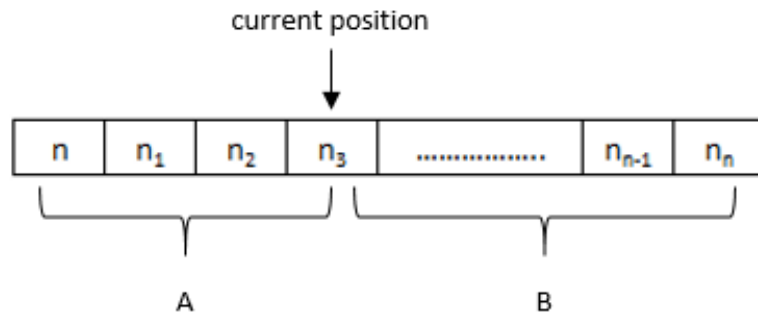


Figure 5.9 Illustration of novel clustering algorithm. As you move along the array, the difference between the sum of angles on either side is calculated.

When the minimum difference is found, the position to split the points is assigned.

The position in which the point set is split is obtained as follows (Eq. 5.7)

$$splitIndex = Min(abs(A - B)) \quad (\text{Eq. 5.7})$$

where

$$A = \forall n \in Arr: n_i < I, \left(\sum_{i=1}^I \arccos(n_I \cdot n_i) \right)$$

$$B = \forall n \in Arr: n_i > I, \left(\sum_{i=I}^{n_{max}} \arccos(n_I \cdot n_i) \right) \quad (\text{Eq. 5.8})$$

where n is the normal, and A and B are the angles between the normal of the current position, $index$, and those of either side of it, $n_i > index$ and $n_i < index$. When the minimum difference is found, all patches in A are classified as cluster one and those in B are classified as cluster two. This approach works particularly well on curved leaves, ensuring that they are split at the peak of their curve; as shown in Figure 5.10.

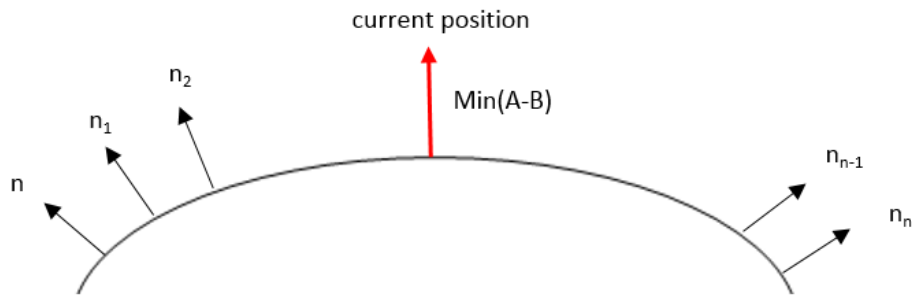


Figure 5.10 An example of the proposed clustering algorithm (Algorithm 5.2) and how it works on a simple curve where n represents the normal for that position and $Min(A - B)$ is the minimum deviation between angles and is where the cluster is split

For more complex curves, the process will occur over a number of iterations, starting with the largest curve. This process is shown in Figure 5.11, where the splits are illustrated in the order in which they occur (left to right).

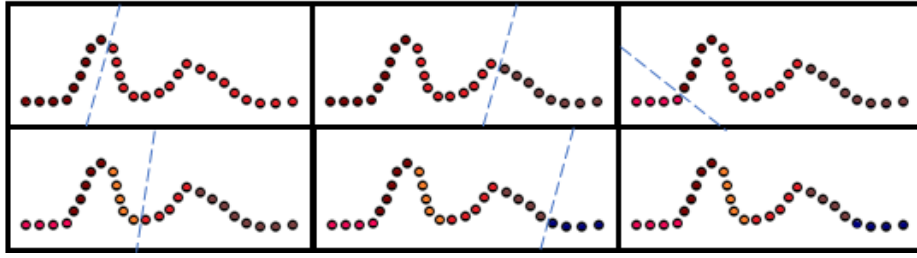


Figure 5.11 An example of the proposed normal clustering algorithm for a complex curve. From left to right and top to bottom: the designated order in which the algorithm splits the point set and the final cluster set (bottom right).

Step 2: Separation

By this point, Step 1 has dealt with many of the constraints, however one remaining difficulty lies within partitioning points that are exceptionally close and with very similar point normals. When PCA (Step 1) is unable to improve the current point set, a modified DIVFRP algorithm is applied. A dissimilarity measure based on the furthest reference points is used to overcome difficulties of clustering areas containing small distinct groups, which often occurs in complex scenes. DIVFRP aims to maximise the dissimilarity function to split points into distinct clusters.

Algorithm 5.3 represents the modified algorithm. Algorithm 5.3 takes a series of clusters as input and outputs and updates the cluster set. The algorithm is as follows: for each cluster, Algorithm 5.3 Line 1, determine the mean, Algorithm 5.3 Line 2. Calculate the furthest point from the mean of the cluster, i.e. the furthest reference point, Algorithm 5.3 Line 3, and calculate an ordered list of the distance from all other points in the cluster to furthest reference point, Algorithm 5.3 Line 4. Find the maximum distance between any two neighbouring points in the ordered list and obtain an index, Algorithm 5.3 Line 5, any points on the left of the index are classified as cluster 1 and any on the right are classified as cluster 2, Algorithm 5.3 Line 6. The final adjustment made to this algorithm is to traverse bordering points in cluster 1 and cluster 2; i.e. the distance between them is determined and points are shifted if necessary, Algorithm 5.3 Line 7. Once each cluster has been evaluated the process terminates, Algorithm 5.3 Line 8.

Algorithm 5.3 The modified DIVFRP algorithm**Input :** A list of clusters**Output :** A list of clusters

-
- ```

1: foreach cluster
2: Calculate the mean of the cluster
3: Determine the further reference point
4: Calculate ordered list of furthest reference points
5: Find maximum distance between two points
6: Split into two clusters based on reference point
7: Traverse each of the points in each cluster to correct
8: end

```
- 

Figure 5.12 illustrates the improvement of the modified DIVFRP algorithm. From the figure, two clear separations of points can be seen; where the points on the left should be split from points on the right due to the gap between them. These points are obtained from a rice (*Oryza sativa*) plant, where the left and right denser areas represent two distinct tillers and the gap (i.e. the larger triangles) in the middle, represent spacing between two distinct tillers (i.e. stems). The left image indicates the original algorithm, where the cluster is clearly incorrectly partitioned, including points from two distinct tillers. The modified version, on the right, shows a correctly partitioned cluster which is obtained by adjusting points.

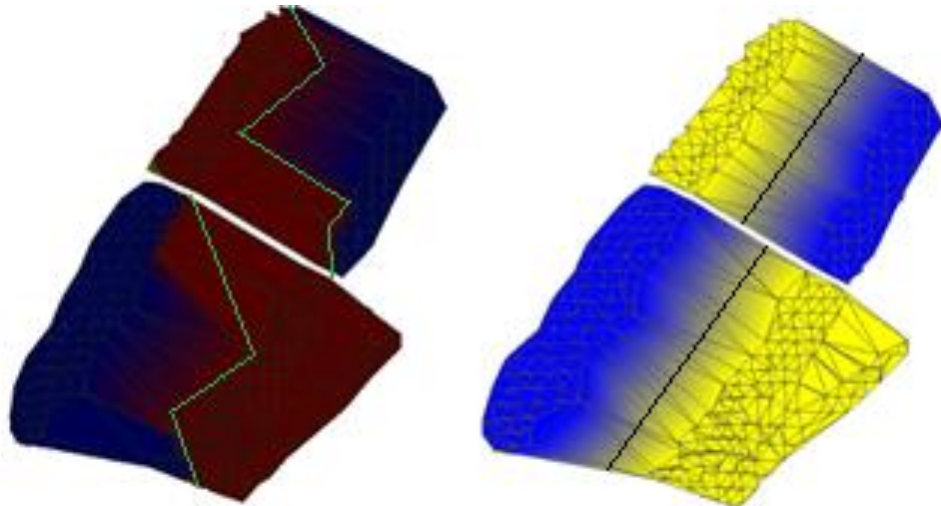


**Figure 5.12** Comparison of the original (Left) versus the modified (Right) DIVFRP algorithm. Use of the original DIVFRP algorithm leads to red connection across each side whereas the modified DIVFRP algorithm correctly partitions both datasets.

### Step 3: Reclassification

In some instances, single points may be incorrectly placed. These are typically located at the borders of clusters (the outer most points of a cluster). This can be due to noise or incorrectly estimated normals, due to the point proximity when estimating normals.

The reclassification step aims to shift border points (i.e. those furthest away from the cluster centroid) to another neighbouring cluster (i.e. one in close proximity), should the point be closer to the neighbouring cluster than the cluster it is currently assigned. This is a two-step process:

The **first step** uses a KD search tree constructed using the centroids of the clusters. For each boundary point in each cluster, the nearest neighbour is found using a search tree of cluster centroids. If the nearest neighbour cluster differs from the one in which the point resides, it is moved, otherwise the point remains. This simple process reassigns points that are closer to another cluster than the one they are originally classified.

The **second step** again uses a search tree, this time constructed using all points in the model. For each boundary point in each cluster, 26 of the nearest neighbours to the point are selected. This number was selected as 26 ensures that all points that surround the voxel are found if they exist (Figure 5.13). This works successfully as the point cloud is produced using the model merging process (Section 5.2), thus the maximum distance is known and therefore points that are further away can be excluded.

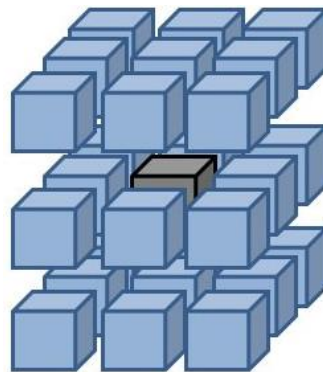


**Figure 5.13** The nearest neighbour voxels, illustrating the surrounding 26 neighbours (blue) that a target voxel (grey) can have.

The Reclassification step ‘smooths’ the points in clusters and ensures that no points have been incorrectly moved in the previous step (Separation). The mode of the clusters reference ID, from the 26 neighbours, is calculated and, if it differs from the cluster the current points resides in, the point is moved. Put simply; if the neighbouring points appear in another cluster, move the point to the same cluster.

## 5.4 Results

Evaluation of the clustering algorithm (Algorithm 5.2) is conducted on real data obtained from the rice plant model by segmenting the target plant along the z axis. The segments are illustrated in Figure 5.14 and the number of clusters are easily identifiable through visual inspection. Evaluating clustering algorithms on complex 3D points from scanned data is particularly challenging; the number of clusters is not known, and the opinion of a ‘correctly’ clustered 3D point set is subject to personal opinion, and thus human error. In this work a 3D point set using the AVC is captured and segmented into levels of varying difficulty (Figure 5.14). The clustering algorithm is evaluated against existing algorithms on these segments, where the true number of clusters can clearly be seen visually.

The evaluation is performed for four algorithms; 1. The method proposed here, 2. K-Means clustering, 3. IPDDP algorithm and 4. The spectral clustering algorithm. Each algorithm is compared with respect to the time taken to cluster the point set, the number of clusters found and the percentage of points that lie correctly within a cluster; i.e. the accuracy.

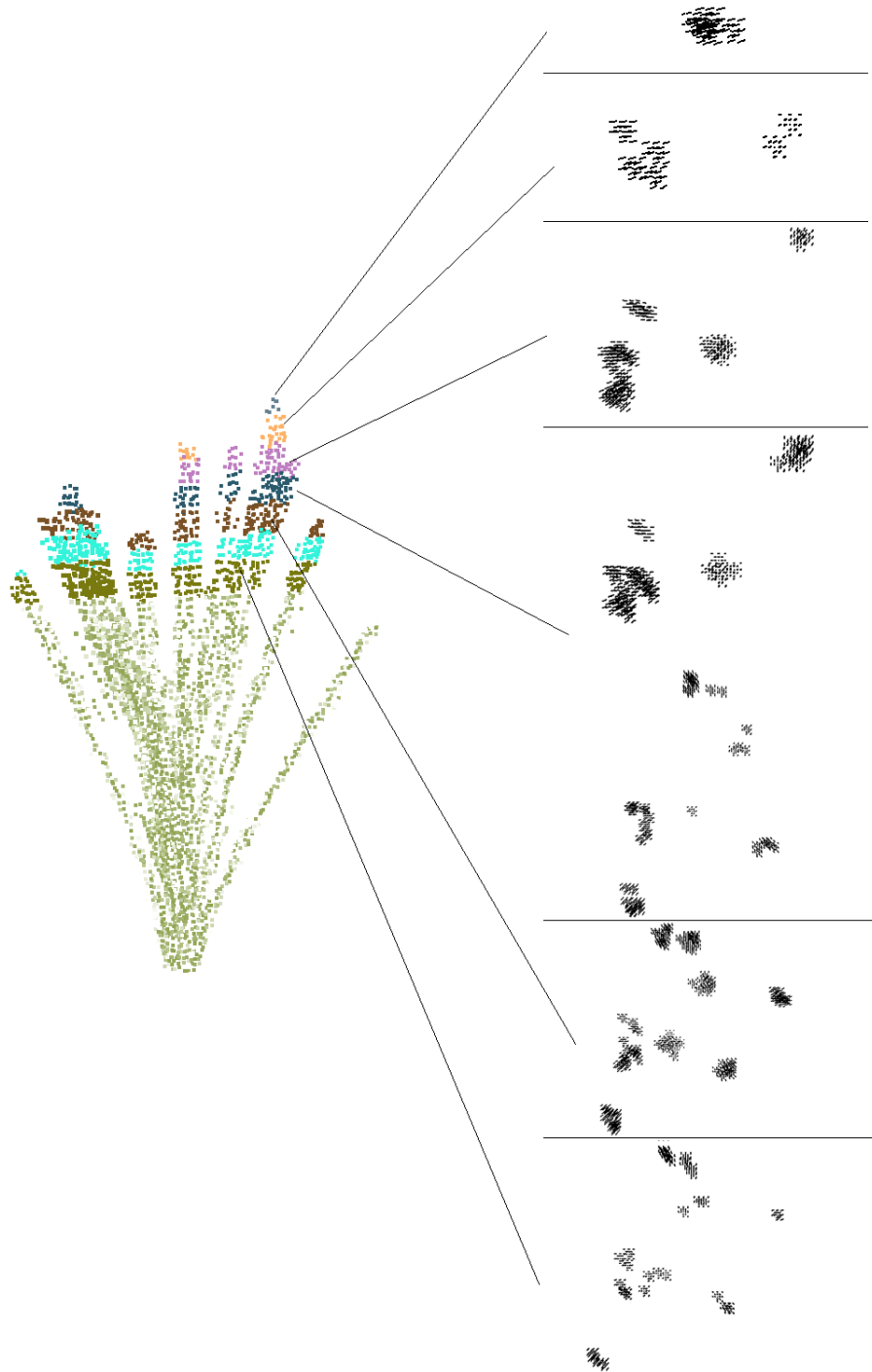


**Figure 5.14** Example of cluster Segments. A rice plant (*Oryza sativa*) was segmented and 6 segments of varying complexity have been selected. The number of clusters and points are known, which is used as the basis for ground truth.

**Segment One**


Clusters: 1

Number of points: 365

**Original Data**


|                     | Time (s) | Clusters | Correct Points | % Correct |
|---------------------|----------|----------|----------------|-----------|
| Proposed Clustering | 0.183    | 1        | 365            | 100%      |
| K-means             | 0.005    | 1        | 365            | 100%      |
| IPDDP               | 0.113    | 1        | 365            | 100%      |
| Spectral            | 0.392    | 1        | 365            | 100%      |

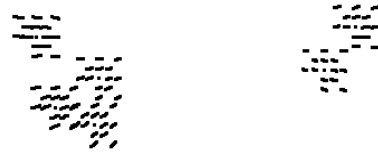
**Table 5.4** Comparison of the proposed clustering algorithm relative to existing clustering algorithms for Segment One.

While it may seem illogical to evaluate clustering methods on a single cluster, some simple algorithms partition to create multiple clusters. From the results (Table 5.4) it is possible to see each of the four methods correctly partitions the dataset into a single cluster. Even on a small number of points, spectral clustering takes double the amount of time in comparison to the method proposed here.

**Segment Two**

Clusters: 3

Number of points: 426

**Original Data**

|                     | Time (s) | Clusters | Correct Points | % Correct |
|---------------------|----------|----------|----------------|-----------|
| Proposed Clustering | 0.374    | 3        | 426            | 100%      |
| K-means             | 0.001    | 2        | 352            | 82%       |
| IPDDP               | 0.121    | 4        | 195            | 45%       |
| Spectral            | 0.513    | 3        | 426            | 100%      |

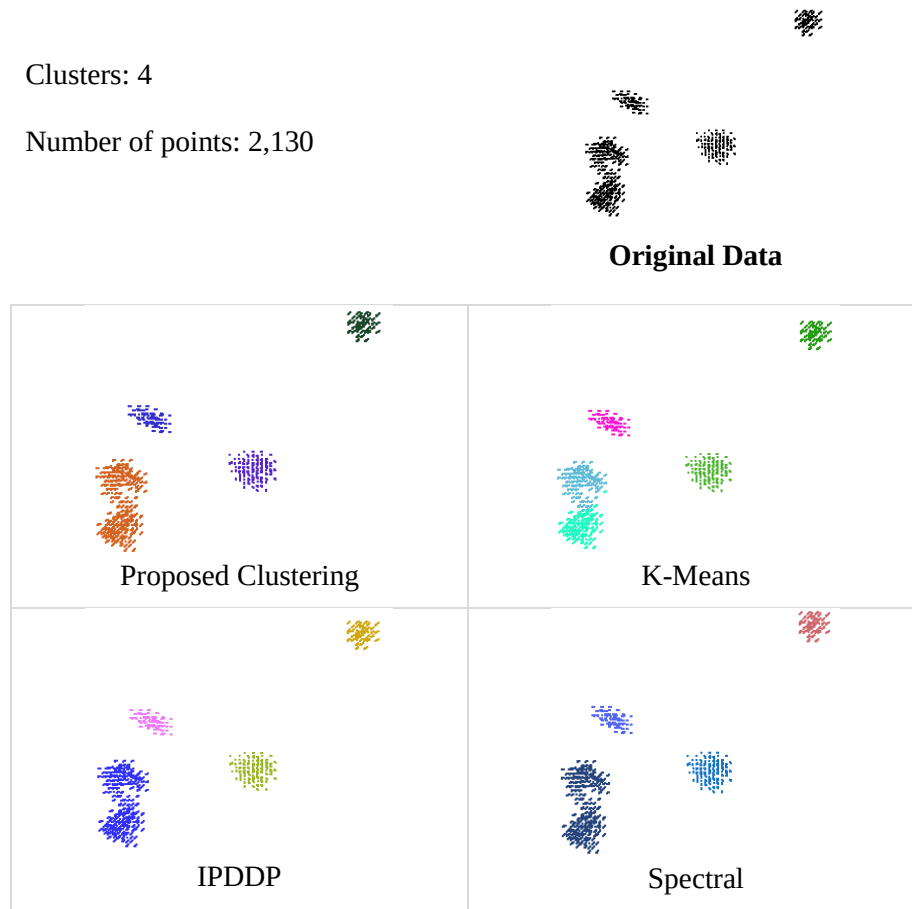
**Table 5.5** Comparison of the proposed clustering algorithm relative to existing clustering algorithms for Segment Two.

Three distinct clusters are visible in Segment two. Both the method proposed here, and spectral clustering, correctly clustered the data with 100% accuracy. K-Means and IPDDP under- and over-estimated cluster number, respectively. The K-means algorithm is significantly quicker than other approaches but with reduced accuracy of 82%. Spectral clustering is, again, the algorithm with the largest computational requirements.

### Segment Three

Clusters: 4

Number of points: 2,130



|                     | Time (s) | Clusters | Correct Points | % Correct |
|---------------------|----------|----------|----------------|-----------|
| Proposed Clustering | 2.512    | 4        | 2,130          | 100%      |
| K-means             | 0.005    | 5        | 1,619          | 76%       |
| IPDDP               | 2.260    | 4        | 2,130          | 100%      |
| Spectral            | 21.90    | 4        | 2,130          | 100%      |

**Table 5.6** Comparison of the proposed clustering algorithm relative to existing clustering algorithms for Segment Three.

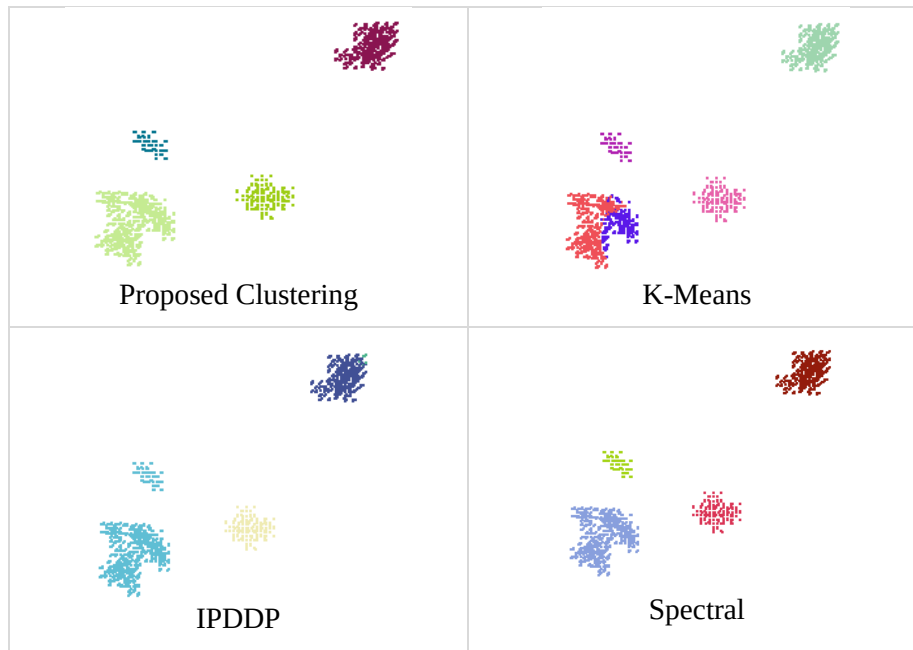
In this segment, K-Means fails to correctly partition the data, subdividing the points with a loop (bottom left corner). IPDDP, spectral clustering and the proposed method correctly partitioned the data. It is worth highlighting the increase in runtime according to complexity (i.e. number of points) when using spectral clustering; with 0.5 s taken for < 500 points (Segment two) relative to

21.9 s with 2130 points (Segment three). K-means again performs faster but with only 76% accuracy.

#### Segment Four

Clusters: 4

Number of points: 2,414



|                     | Time (s) | Clusters | Correct Points | % Correct |
|---------------------|----------|----------|----------------|-----------|
| Proposed Clustering | 2.988    | 4        | 2,414          | 100%      |
| K-means             | 0.008    | 5        | 2,002          | 82%       |
| IPDDP               | 1.928    | 4        | 2,414          | 100%      |
| Spectral            | 29.91    | 4        | 2,414          | 100%      |

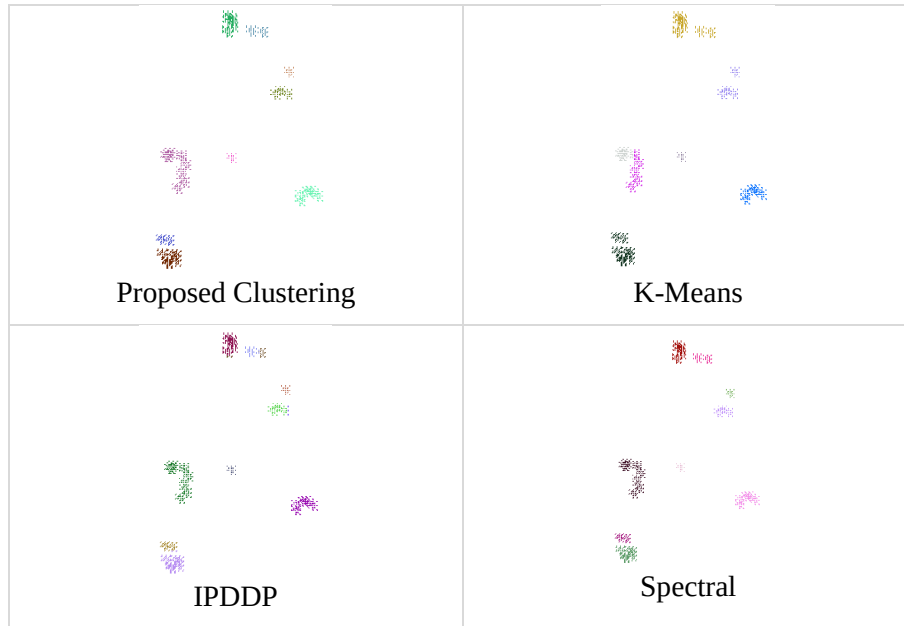
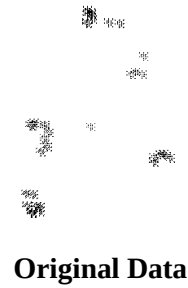
**Table 5.7** Comparison of the proposed clustering algorithm relative to existing clustering algorithms for Segment Four.

For Segment four, K-means is unable to correctly cluster, achieving 82% accuracy, however, the other 3 algorithms successfully cluster the data with 100% accuracy. Note, as previously mentioned, the large computational requirements of spectral clustering, which takes 10 times as long to run as the proposed clustering algorithm.

### Segment Five

Clusters: 9

Number of points: 2,840



|                     | Time (s) | Clusters | Correct Points | % Correct |
|---------------------|----------|----------|----------------|-----------|
| Proposed Clustering | 7.008    | 9        | 2,840          | 100%      |
| K-means             | 0.007    | 7        | 2,117          | 74%       |
| IPDDP               | 3.106    | 10       | 2,031          | 71%       |
| Spectral            | 45.13    | 9        | 2,840          | 100%      |

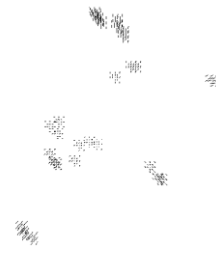
**Table 5.8** Comparison of the proposed clustering algorithm relative to existing clustering algorithms for Segment Five.

Segment five is a more complex clustering problem, with a varying range of cluster shapes and sizes. Only the proposed clustering algorithm and spectral clustering achieve 100% accuracy. However, spectral clustering has significant computational requirements, taking 45 s to compute compared to 7 s for the novel algorithm. K-means remains significantly faster than any other method but achieves only 76% accuracy.

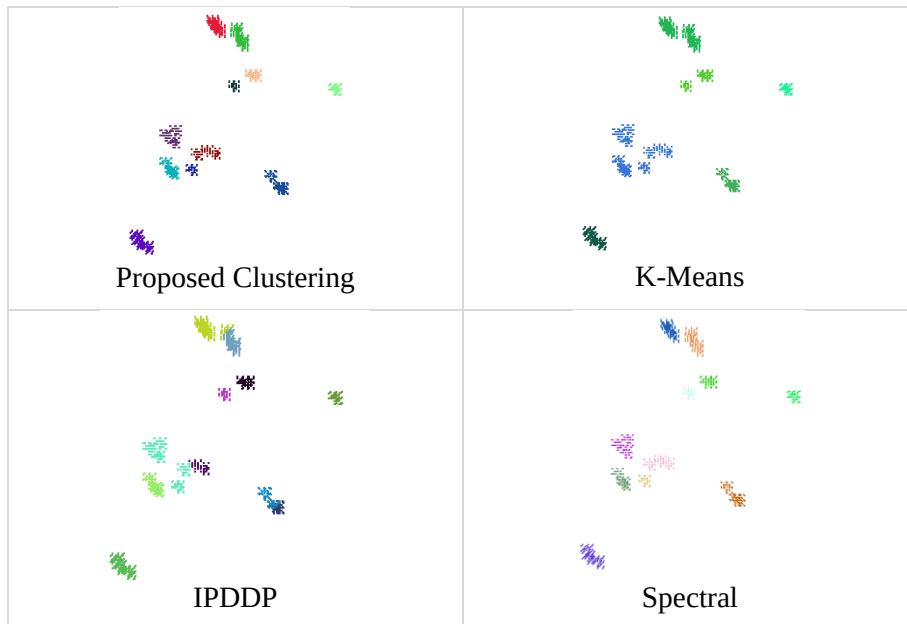
### Segment Six

Clusters: 11

Number of points: 1,988



Original Data



|                     | Time (s) | Clusters | Correct Points | % Correct |
|---------------------|----------|----------|----------------|-----------|
| Proposed Clustering | 2.762    | 11       | 1,988          | 100%      |
| K-means             | 0.006    | 6        | 977            | 49%       |
| IPDDP               | 1.564    | 12       | 1,811          | 91%       |
| Spectral            | 18.63    | 11       | 1,988          | 100%      |

**Table 5.9** Comparison of the proposed clustering algorithm relative to existing clustering algorithms for Segment Six.

In segment six, the search space is reduced to less than 2,000 points. K-Means significantly under-clusters, finding only 6 of a possible 11. Spectral and proposed algorithm again successfully cluster the data set with 100% accuracy.

From the evaluations on all six segments (Tables 5.4-5.9), only the proposed clustering algorithm and spectral clustering achieve 100%. However, spectral clustering has the drawback of long computational requirements. The clustering algorithm proposed here is therefore used to cluster the point cloud in the remainder of this thesis.

## 5.6 Discussion

Two novel contributions have been made in this chapter. The first is the generation and application of the model merging algorithm (Algorithm 5.1) which is shown to not only improve the accuracy in comparison to raw PMVS data, but also significantly reduce the size of the dataset while maintaining object integrity. This is later shown to improve the accuracy of 3D modelling by removing the outliers which often prove to be problematic when performing surface reconstruction (Chapter 6).

Secondly, the devised clustering algorithm (Algorithm 5.2) can satisfy the constraints and statistically perform equally as well as the existing spectral clustering method, but with significantly lower computational requirements. In comparison to K-means and IPDDP, the proposed clustering algorithm can at least equal, but often increases, the accuracy of clustering in all instances. Whilst processing time is reduced in comparison to spectral clustering, it is increased relative to the K-means and IPDDP algorithms.

The clustering of points directly impacts the quality of the reconstructed model. Consequently, a correctly partitioned dataset is of utmost importance and it is necessary to prioritise accuracy over speed. The proposed clustering algorithm (Algorithm 5.2) is therefore used for surface reconstruction (Chapter 6). Finally, Figure 5.15 illustrates the proposed clustering algorithm applied to an entire rice plant obtained using the AVC. Though visually hard to inspect through a 2D image, it is possible to see the correctly clustered points, particularly where leaf segments join together.



**Figure 5.15** An example of the proposed clustering algorithm applied to a rice point set obtained using the AVC

## Chapter 6

# Surface Reconstruction

Chapter 6 presents an algorithm for fitting a surface over the clustered points provided by Chapter 5. The chapter opens with an introduction to surface reconstruction before moving on to discuss the four phases of the algorithm. The first phase defines how the triangulation of point clusters is performed to produce a set of surfaces. Phase two applies level sets to these surfaces to expand to the boundaries of the object in 2D images. In the third phase, mesh modification methods are discussed, and in the final phase the merging of triangulated point clusters is performed. The final section of the chapter presents the results and highlights that using the point and volumetric models obtained using the active vision cell (AVC) can improve the accuracy of reconstruction, when compared to existing methods. The chapter is concluded with a discussion of the work conducted here.

## 6.1 Introduction

The final stage of the proposed 3D modelling process produces a surface representation of the object. Starting with the initial point clustering obtained using methods presented in Chapters 4 and 5, the points in the clusters are projected to an orthogonal regression plane and are triangulated to produce a series of triangulated point clusters, known from this point forward as cells. Level sets are run on each cell to expand boundary points and maintain sharp features and edges. The surfaces are smoothed using a two-step Laplacian smoothing algorithm and long triangles are removed. A merging algorithm is applied to the cells based on connectivity information obtained from the volumetric model (Chapter 4) and the Euclidean distance.

Algorithm 6.1 presents the surface reconstruction algorithm discussed in this chapter. A series of point clusters and an image set obtained using the AVC are used. For each of the clusters, triangulation is performed, Algorithm 6.1 Line 2, to produce a series of cells, followed by level sets to expand the boundaries, Algorithm 6.1 Line 3. The resulting cells are smoothed; first internal points and then boundary points, Algorithm 6.1 Line 4. The second stage creates a fully connected surface. The boundary for each cell is detected, Algorithm 6.1 Line 10, cells are then evaluated to determine if they are connected, and if so they are merged, Algorithm 6.1 Line 13.

---

**Algorithm 6.1** Surface reconstruction algorithm

---

**Input :** *A list of clusters, An image set*

**Output :** *A surface representation*

```

1: for each cluster
2: Perform delaunay triangulation
3: Perform level sets on triangulated cluster
4: Smooth the surface of the cluster
5: end
6: Intialise surface as all clusters
7: While the number of clusters is > 1
8: Assign surface as clusters
9: Intialise a new empty surface
10: Detect boundaries of clusters
11: for each cluster
12: Check if cluster is already merged on all boundaries
13: If not merge cell based on algorithm proposed here
14: end
15: end
16: Apply final surface smoothing step

```

---

Figure 6.1 visually illustrates the process of the proposed reconstruction method and consists of six steps:

1. The original clustering is obtained from the hybrid clustering algorithm (Algorithm 5.2, Chapter 5).
2. Delaunay triangulation of clusters to form a series of cells.
3. Level sets are applied to the cells; the green indicates the expansion by projecting the original points to 2D images and expanding these until an appropriate front (boundary) is found.
4. Smoothing of the internal and external points is carried out
5. The cell merging process adds additional points between two cells being merged
6. The final smoothing stage applied to the new triangles generated from cell merging.

The novel contributions made and discussed in this chapter are: a hybrid algorithm for surface reconstruction (Algorithm 6.1) and; a cell merging method (Section 6.5, Phase 4), which uses connectivity information obtained from the volumetric model and is otherwise unavailable when using only a point set.

Throughout this Chapter, the term *projection* refers to the transformation of some 3D point in world space to a 2D coordinate on the orthogonal regression plane. The term *back-projection* refers to the transformation of some 2D point to a 3D world coordinate.

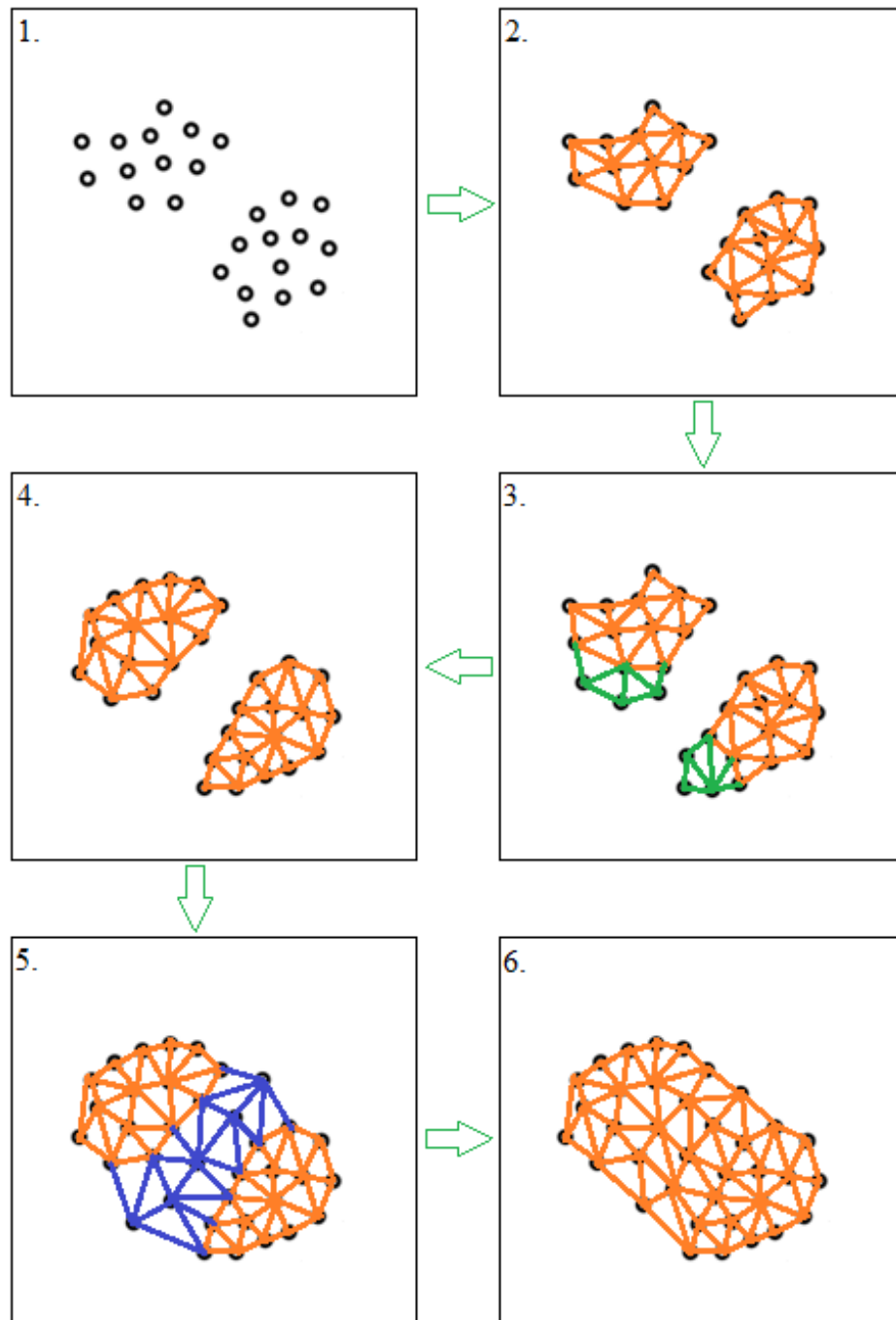


**Figure 6.1** The surface reconstruction process; 1. The starting clusters, 2. The triangulated clusters (cells), 3. Level sets applied to cells, 4. Smoothing of the cells, 5. Cell merging, 6. Final smoothing step

## 6.2 Phase 1: Triangulation

Initial surface estimates are achieved through the process of triangulation; the partition of a surface into a set of triangles, using Alpha shapes ( $\alpha$ -shapes).  $\alpha$ -shapes are a subset of the Delaunay triangulation, which groups points with specific properties such that each *circumcircle*, a circle that passes through each of a triangles vertices, is empty. That is, it does not contain any other points. Figure 6.2 portrays Delaunay Triangulation (left) and non-Delaunay Triangulation (right).

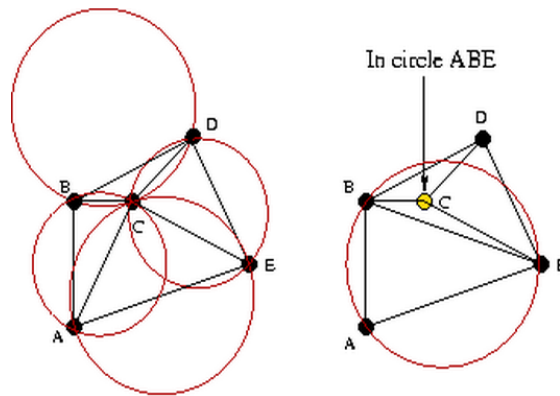


**Figure 6.2** Delaunay Triangulation - representing correct (left) and incorrect (right) triangulation, (Peterson, 1998).

The Voronoi diagram (Figure 6.3) is the result of partitioning some point set into polygons, such that each polygon encompasses exactly one point. From Delaunay Triangulation, a Voronoi diagram can be acquired by drawing edges perpendicular to the boundaries, or vice versa, to reverse the transformation. Figure 6.3 depicts this with perpendicular lines creating the Voronoi diagram.

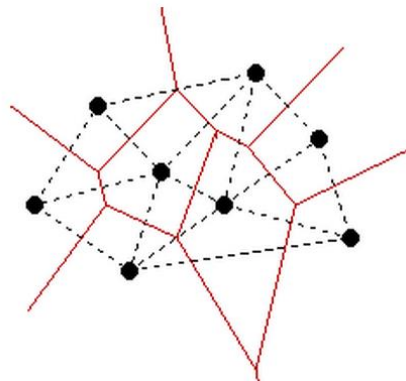


**Figure 6.3** Delaunay Triangulation and the Voronoi Diagram, (Peterson, 1998).

$\alpha$ -shapes use Delaunay Triangulation and Voronoi Diagrams to find boundary points and edges, which is a vital part of cell merging; the process of taking two clusters that have had surfaces fitted, known as cells, and merging them together to create a single cell. Note: clusters of points become cells once they have been triangulated.  $\alpha$ -shapes are used to generate surfaces by connecting point pairs using an empty disc of radius  $\alpha$ ; where a high  $\alpha$  value reduces the level of detail, and a low value increases it. Thus, a trade-off between accuracy and computational speed must be made in most applications. In this work, the radius used is calculated using the points in which a surface needs to be fitted (Eq. 6.1).

$$\alpha = \frac{1}{|P|} \cdot \left[ \sum_{i=0}^{|P|} \sum_{j=0}^{|P|} \sqrt{2 \cdot (\vec{p}_i - \vec{p}_j)} \right] \quad (\text{Eq. 6.1})$$

where  $P$  is the set of all points and  $p$  is the 3D position vector. Triangulation is performed in 2D, which requires that the 3D points are projected to the orthogonal regression plane of their respective cluster, obtained using PCA. The 2D points are then triangulated using a divide and conquer process, starting with a single super triangle and subdividing. A super triangle is one which encloses all of the 2D points on the orthogonal regression plane. The edges of the super triangle are determined, plus the centre of the circumcircle and radius of the triangle are calculated. If a point lies within the circumcircle of the triangle, edges are added to split the triangle into two smaller, sub-triangles. The process of determining whether a point lies within the circumcircle and splitting the triangle if it does is repeated until each circle is empty. At the end of the algorithm all super triangle edges are removed.

Triangulation of this form maximises all angles of the triangles, thus avoiding, or minimising, the occurrence of sliver triangles; those that are thin and long (Berg, 2008). After triangulation, however, the triangles may not faithfully represent the object and may consist of a series of gaps and holes as the clusters are not connected, or may contain long, uneven triangles; the consequence of triangulation in 2D and back-projecting to 3D. These issues are addressed in Phase 3 (Section 6.4). Triangulation is performed in 2D as the efficiency is improved over traditional 3D methods. Moreover, triangulating occluded, crowded scenes in 3D is incredibly challenging and often results in a poor

surface representation where distinct objects, or segments of objects, are triangulated together.

In turn, each cluster is projected to its corresponding orthogonal regression plane and triangulated in 2D, producing a series of cells. Level sets (Phase 2, Section 6.3) requires that data is in 2D, consequently, points are back-projected to 3D following this phase.

### 6.3 Phase 2: Level Sets

The initial surface estimate obtained from triangulation needs reshaping to expand to the edges of the object, without overlaps, to create an accurate representation. Level set methods have been chosen to grow the initial surface. Introduced by (Osher and Sethian, 1988), they are used for tracking evolving interfaces in a wide variety of applications (Sethian and Smereka, 2003). Directly applicable to this work, (Pound *et al.*, 2014) uses levels sets during a surface reconstruction process in order to expand boundaries of clusters. However in (Pound *et al.*, 2014), the resulting boundaries are not adjusted to faithfully represent curvature of the object as they are expanded on planar surfaces. In this work, this is overcome by back-projecting to 3D coordinates, once level sets have been performed, based on the surrounding point transformation matrices. This is discussed further in Phase 3 (Section 6.4).

Before level sets are applied to the cells, the visibility of each cell in each image is determined and the cells are assigned an image set ordered by those that can ‘best’ see the cell. This is based on the percentage of and angle at which the triangle is seen, where a higher seen percentage and an angle close to perpendicular are ranked higher. This information is obtained through the ray tracing method (Section 4.5), in which rays are projected from the known camera positions into the volumetric model. From this, it is possible to calculate the angle between the projected ray (i.e. the camera position) and the normal of the cell, plus the percentage of the cell which can be seen from each camera (i.e. the total number of rays that hit a cell from a given camera).

Each cell is used to initialise a level set method in which the hyper-surface, a 2D surface in a 3D setting, moves and deforms under the control of a speed function until an appropriate front is found; the speed function defined in (Pound *et al.*, 2014) is used here. An appropriate front is one that most accurately represents the boundaries of the object in the images that have the ‘best’

visibility of the cell; that is, they best fit the 2D image confined within the visible boundaries, evaluated by projecting to the 2D in which the cell is best seen. The cells are assigned a z-buffer to manage depth during growth.

The novel contribution achieved through this surface reconstruction process involves the back-projection of level set points from their plane to 3D world coordinates, using the connectivity information of neighbouring points. This is achieved by projecting the level set boundaries for each cell from the 2D image plane to their 3D positions, and the original points, those not estimated by level sets, are back-projected using their corresponding projection matrices. To project the new points obtained from level sets, the projection matrices (consisting of rotation and translation) of connected vertices between the new level set point and the existing points are used. The new point is projected using the average of the projection matrices of existing points which are connected via edges. This ensures relative consistency between existing and new points by adding the difference between the planar coordinates ( $x$  and  $y$ ) of the original points and the new boundary point estimated from level sets to ensure the level set points are positioned correctly in 3D space. As a result, a series of 3D world coordinates for both original points and boundary points estimated using level sets make up each of the cells. This more faithfully represents the object by maintaining the curvature of the object.

## 6.4 Phase 3: Mesh modifications

It is common for a surface to consist of holes due to missing data; shape irregularities from level sets and triangulating in 2D; and the appearance of long triangles, a consequence of missing data and 2D triangulation. Therefore, before proceeding to merge the surfaces, cells are individually repaired. The repair process involves three key steps to address these issues; 1. Hole filling, 2. The smoothing of internal and external (i.e. boundary) points, and 3. The removal of long triangles. Before any of the repair steps can be applied, the boundary points for each cluster need to be identified. The boundary is defined as any edge that has no references to any other edge in the cluster. Boundary detection is performed as an iterative process, evaluating each of the edges. From this, the outer boundary (Figure 6.4A) and any holes, i.e. the internal boundary (Figure 6.4B), can be determined. The internal and external boundaries can be detected

in the same fashion though the distinguishable feature is that an internal boundary will have no internal edges or nodes.

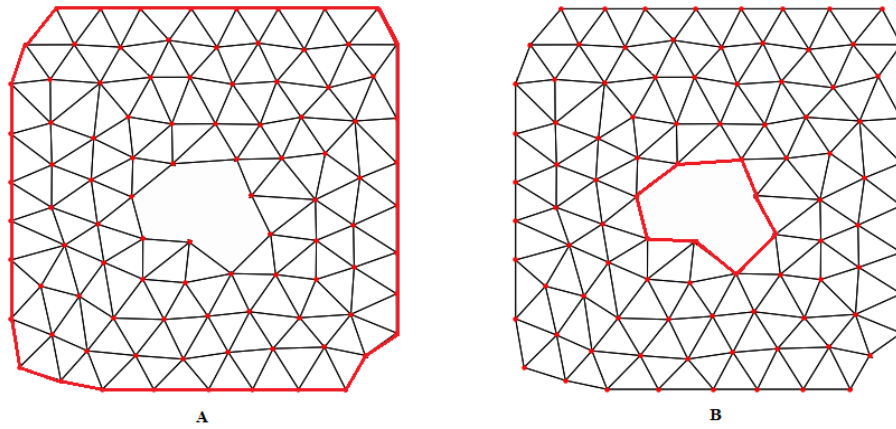


**Figure 6.4** Boundary detection; **A.** The external boundary, **B.** The internal boundary which indicates the presence of any holes

### Hole Filling

A simple hole filling process is applied in 3D space and takes the internal boundaries (Figure 6.5A); calculates the centroid of the hole; and inserts a single point (Figure 6.5B). If the distance between the inserted centroid and internal boundary points exceeds the distance of the  $\alpha$ -shape radius, defined previously, additional points are inserted using the average between the centroid and boundary point in each direction. This is repeated until the distance between all points that are being fitted in the hole are less than or equal to the  $\alpha$ -radius. Once the new point set is complete, the points are triangulated in 2D (Figure 6.5C), as previously discussed, and inserted into the original cluster with the corresponding triangles to accurately fill the hole (Figure 6.5D).

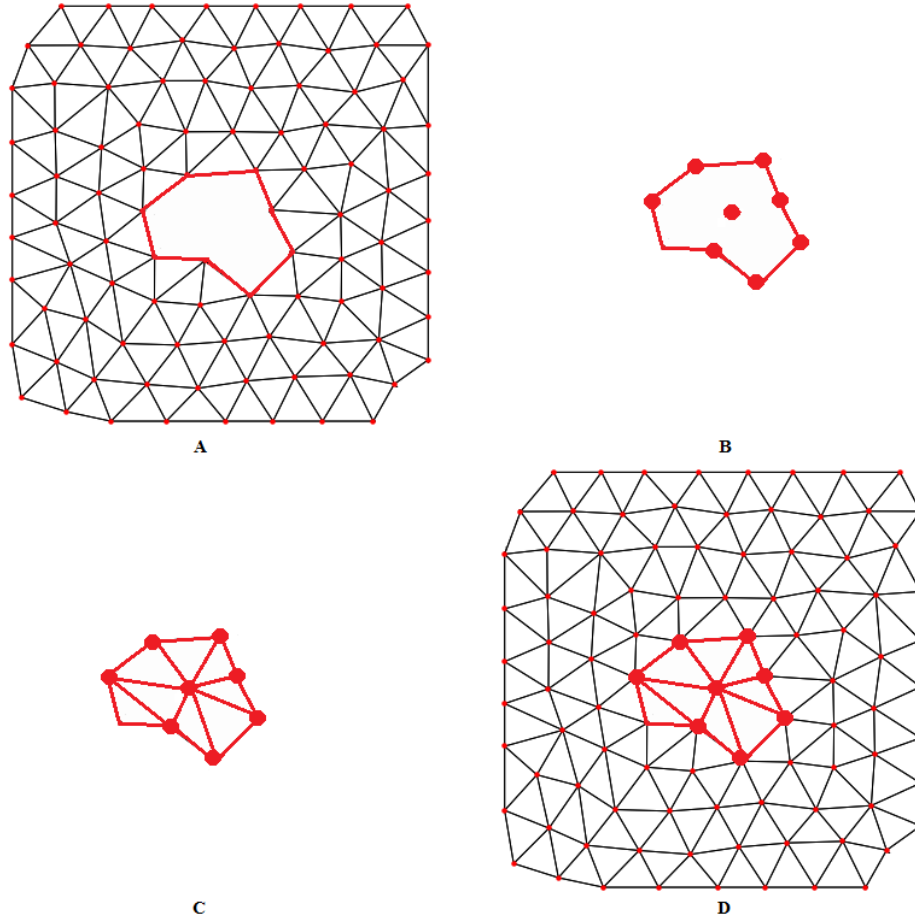


**Figure 6.5** The Hole Filling Process. A. The original triangulation, B. The extracted internal boundary points and centroid, C. The triangulation of the hole, D. The insertion of the new triangulation into the original surface

### Smoothing

After hole filling, smoothing is applied to counter the undesirable irregularities of back-projecting to 3D. An unsmooth surface is often generated during hole filling (above) as the projection of the new 2D triangulated surface to 3D world coordinates often results in an irregular shape. The smoothing process has two key steps; the first is to smooth all the points of the cell that are not part of the boundary, in which the new position of some internal point  $\vec{p}$  is calculated as (Eq. 6.2). The pointset  $P$  is organised such that all non-boundary points have indices between 0 and  $Y$  and that boundary points have indices between  $Y$  and  $|P|$

$$\vec{p} = \alpha \cdot \left[ \left( \frac{1}{Y} \right) \sum_{j=0}^Y \vec{p}_j \right] + \beta \cdot \vec{p}$$

(Eq. 6.2)

where  $\alpha + \beta = 1$  and determines the impact the other points in the cell have on the new position and a small  $\alpha$  means the original point is adjusted only slightly.  $Y$  is the total number of boundary points. Finally,  $\vec{p}$  is a 3D vector representation of the point position in world space.

The second stage of the smoothing process is the smoothing of the boundary points. Unsmooth boundary points may have been generated by the level set method, depending on whether the boundary was expanded, and, as a result, may not fully correspond to the cell shape and structure. By smoothing the boundary points generated by level sets with the existing points it is possible to improve the accuracy of level set points with respect to the original curvature of the object. This is done as follows (Eq. 6.3):

$$\vec{p} = \alpha \cdot \left[ \left( \frac{1}{|P| - Y} \right) \sum_{j=Y}^{|P|} \vec{p}_j \right] + \beta \cdot \vec{p}$$

(Eq. 6.3)

### Long Triangles

Once smoothing has been performed, any ‘long’ triangles are removed. External long triangles are defined as any that contain an edge that has a length larger than three times the radius and is classified as a boundary edge. Internal long triangles, those that do not consist of a boundary edge, are divided by employing the process used for hole filling (discussed above). The radius is used as a threshold as it is calculated automatically from the data. Consequently, the radius is variable and adjusts across varying object shapes, sizes and density; unlike previous approaches that are constrained to fixed values or user input. This allows the process to be applied automatically to any target object.

## 6.5 Phase 4: Cell Merging

A series of smooth cells now represent the object. Nevertheless, to further increase the accuracy, it is necessary that these cells are merged to produce a fully connected mesh.

Initially a small search space constructed using the centroids of the cells, is produced; the centroid of each cell, a single point  $\vec{p} = (x, y, z)$ , is used to initialise a search tree. Cells are ranked against each other with respect to the

distance between them, based on the distance between the centroids. For each of the cells, the boundary points, detected previously, are evaluated against the boundary points of those closest to the cell. Should any of the boundary points in differing cells be connected, that is they are neighbouring voxels in the volumetric model (see Chapter 4), they are scored (Eq. 6.4).

$$score = \frac{1}{r^2} \cdot (f(c_i, c_x) + f(c_i^v, c_x^v))$$

**(Eq. 6.4)**

where  $r$  is the radius and  $f(x, y)$  is a distance function that returns the Euclidian distance between  $x$  and  $y$ . Once cells have been evaluated and scored, those with a score of less than the radius to the power of 2 are connected. In this instance no further evaluation is necessary as the voxels are known to be small.

For those cells with higher scores, the cell merging process is applied as follows: at each iteration two cells, those with the lowest score, are selected and merged together (Figure 6.6A), in the instance that multiple cells have the same score then a random selection between the lowest scoring cells is made. The boundaries are taken and projected to 2D planar coordinates and a centroid is added (Figure 6.6B and 6.6C, respectively). The 2D planar coordinates are triangulated (Figure 6.6D and 6.6 E), and the triangles that have vertices from two cells are re-projected to the original surface (Figure 6.6F). Finally, the position in the original 3D model is again obtained by averaging the connected vertices re-projection matrices, as discussed previously. The process repeats until all clusters have been merged together, at this point a final smoothing stage is applied.

In instances where data is not available, i.e. missing points that hole filling was unable to estimate, cells are evaluated in two ways to determine whether they can be merged. First, the centremost position of the gap between two cells is selected and a ray is traced into the carved volumetric model. The carved volumetric model is one where the voxels have been verified as part of the object; that is the voxels contain a point from the point cloud. This is performed during the process described in Chapter 4. The ray is traversed in the direction of the normal of the two cells. If it passes through the centre most point without collision, then the two cells are not merged as a gap between cells clearly exists. If a collision occurs the second evaluation step occurs.

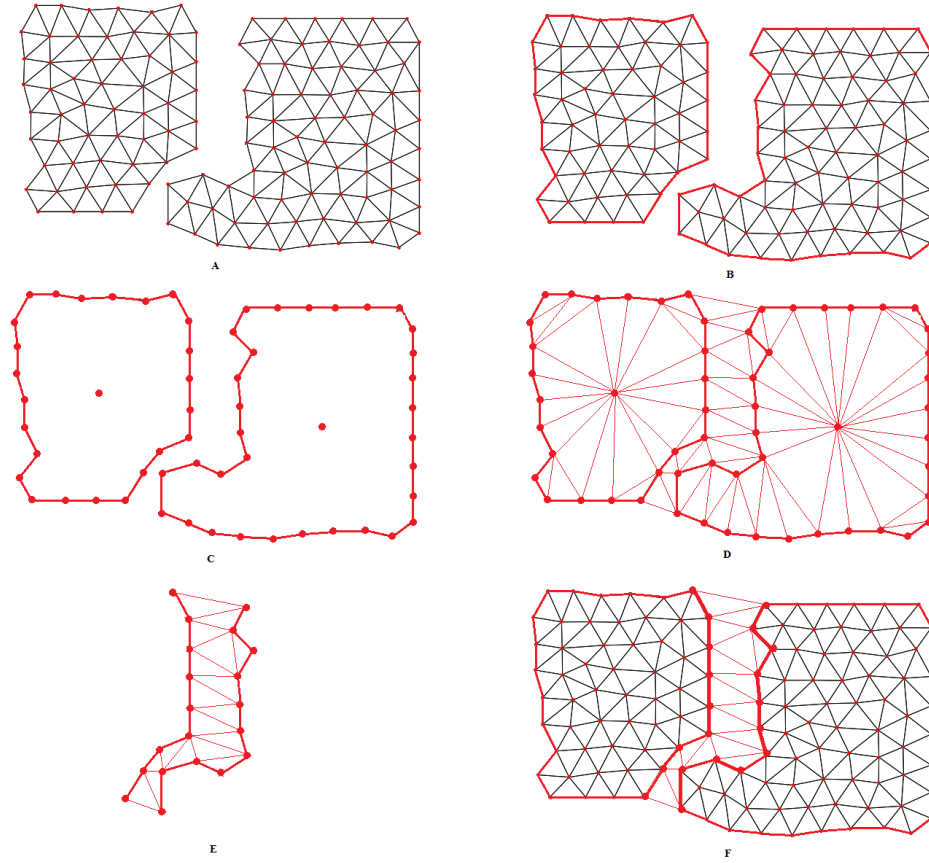


**Figure 6.6** The merging method. A. The original triangulated clusters, B. External boundary detection, C. The extraction of boundaries and insertion of centroid, D. The triangulation of the boundary points, E. Extraction of triangles that contain a boundary point from both clusters, F. The merging of the original clusters and the new triangulated surface

The two cells are projected to multiple 2D images; images that are ranked as the ones that ‘best see’ the cell, where the normal is close to perpendicular and the percentage of the cell visible in the image is highest. Each image is given a score between 1 and 0 which acts as a weight for that image (Eq. 6.5).

$$w = \frac{1}{2} \cdot [vis + f(ray, norm)]$$

(Eq. 6.5)

where  $vis$  is the visibility; if an image that has a full perpendicular view of the cell is given a score of 1, whereas no visibility is assigned a score of 0.  $f(ray, norm)$  determines the angle between the ray (camera) and the cell normal to determine whether the angles are perpendicular and reduce the score the further away it becomes, resulting in a score of between 1 and 0.

The generated centre position is projected to the selected 2D images to determine whether it lies on the object. Projection onto a higher-ranking image constitutes to a higher likelihood of merging acceptance as it has a higher weight. The cell is projected to all images and should it have a score of more than 20% of the image count, the cell is merged. 20% is used to ensure that the cell has been sufficiently identified in images and the sum of scores across all images allows partial views to contribute towards acceptance.

## 6.6 Results

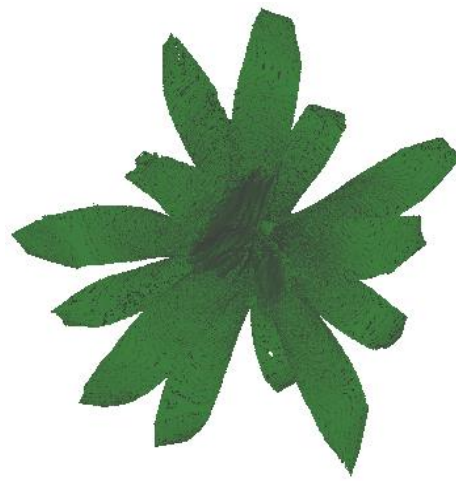
The proposed surface reconstruction algorithm (Algorithm 6.1) is evaluated against the canopy reconstruction algorithm of Pound *et al.* (2014). The X-ray CT-obtained ground truth, which was discussed previously, is again used during evaluation. Images were captured using the AVC proposed in this work and the same image set was provided to each reconstruction method. The total surface area, the number of points, the standard deviation ( $\sigma^2$ ) and the mean distance, in cm, ( $\mu$ ) of the points to ground truth surface are used to evaluate the surface reconstruction.

### Experiment One

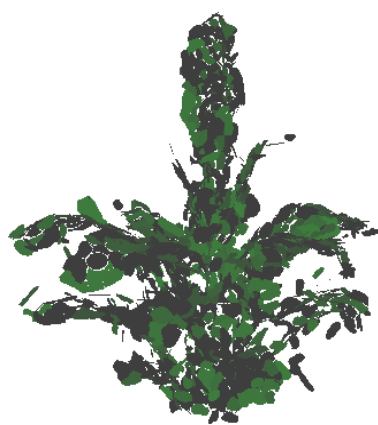
Experiment one is conducted on a Bromeliad (*Vriesea sp.*). The surface area of the X-ray CT scan is 186,476mm<sup>2</sup>.



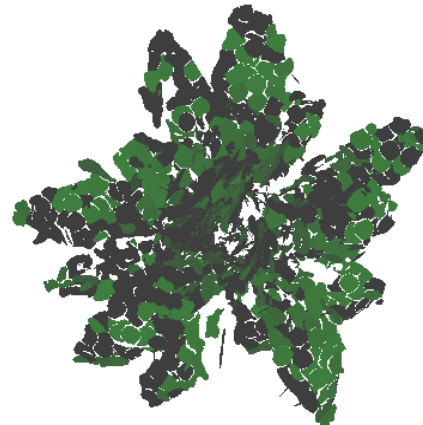
CT Scan – Side



CT Scan - Top



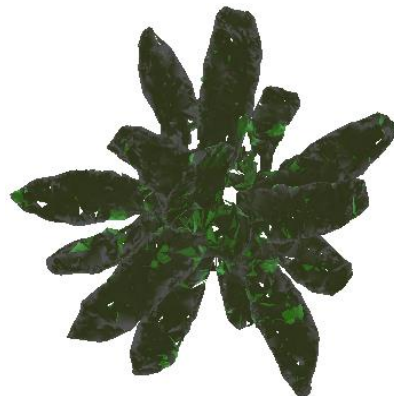
Canopy Reconstruction - Side



Canopy Reconstruction - Top



Surface Reconstruction – Side



Surface Reconstruction - Top

| PMVS to ground truth |            |            |         |        |         |
|----------------------|------------|------------|---------|--------|---------|
|                      | $\mu$ (cm) | $\sigma^2$ | Points  | Images | Area    |
| Canopy               | 3.37       | 3.37       | 509,473 | 40     | 62,747  |
| Surface R            | 1.18       | 1.19       | 237,810 | 36     | 128,834 |

**Table 6.1** Experiment One: evaluation of the proposed surface reconstruction algorithm applied to the AVC image set versus the canopy reconstruction algorithm (Pound et al. 2014) on a Bromeliad (*Vriesea sp.*)

The results indicate that the proposed Surface Reconstruction algorithm significantly improves the existing Canopy Reconstruction algorithm. More than double the surface area is obtained, 128,834 compared to 62,747, and with a greater accuracy; a mean of 1.18 compared to 3.37. Moreover, the proposed Surface Reconstruction algorithm achieves this greater accuracy with a significantly smaller number of points, thus reducing the computational requirements yet ensuring the object is faithfully represented.

The advantage of estimating the  $\alpha$ - radius (Section 6.3) is that it is automatically adjusted to each model. The Canopy Reconstruction algorithm requires that the values are entered prior to investigating the data. Consequently, the Canopy Reconstruction algorithm has a high potential to over or under estimate boundaries, which is a consequence of manually specifying parameters. This is not a problem for the surface reconstruction algorithm due to the automatic calculation of  $\alpha$  combined with the smoothing of level set points into the original data.

## Experiment Two

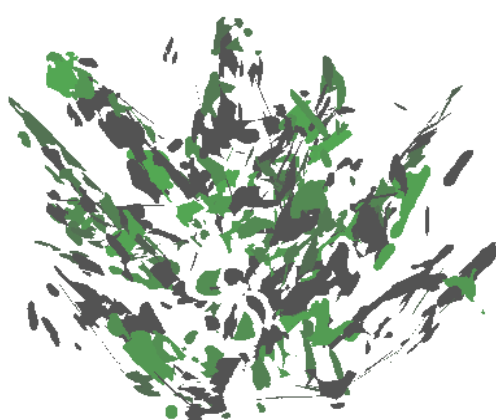
Experiment two is conducted on *Aloe vera*. The surface area of the X-ray CT scan is 117,067mm<sup>2</sup>.



CT Scan – Side



CT Scan - Top



Canopy Reconstruction - Side



Canopy Reconstruction - Top



Surface Reconstruction – Side



Surface Reconstruction - Top

| PMVS to ground truth |            |            |         |        |        |
|----------------------|------------|------------|---------|--------|--------|
|                      | $\mu$ (cm) | $\sigma^2$ | Points  | Images | Area   |
| Canopy               | 5.54       | 4.74       | 234,310 | 40     | 23,645 |
| Surface R            | 1.91       | 1.70       | 125,538 | 31     | 88,542 |

**Table 6.2** Experiment Two: evaluation of the proposed surface reconstruction algorithm applied to the AVC image set versus the canopy reconstruction algorithm (Pound et al. 2014) on *Aloe vera*

The results again show that the proposed Surface Reconstruction algorithm significantly improves the accuracy and surface area relative to the existing algorithm. Though the Surface Reconstruction algorithm may still result in some holes, related to the complexity of the target structure, and may lack some smoothness, due to the thickness of the plant, it still provides a more detailed representation of the plant than the previous method, as seen above. In comparison, the Canopy Reconstruction algorithm leads to particularly large mean and standard deviation due to the outliers and flat surfaces which do not faithfully represent the curvature of the plant. Moreover, the Surface Reconstruction algorithm uses significantly less points 125,538 as opposed to 234,310, reducing the computation requirements and thus improving the efficiency of performing modifications to the mesh.

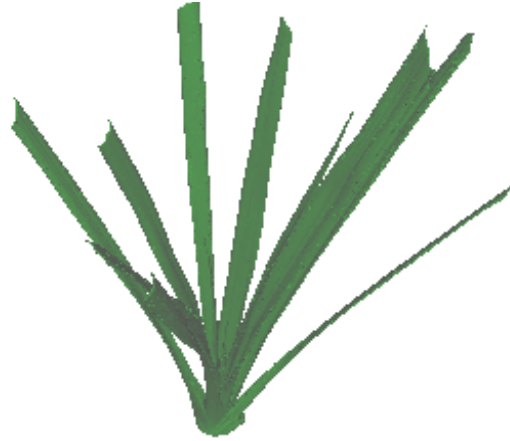
The Canopy Reconstruction algorithm requires parameters that determine the size of the clusters; in this instance a smaller size was chosen as a larger radius resulted in a more erroneous reconstruction with respect to mean and standard deviation. Manually specifying parameters can be problematic particularly when dealing with objects that vary in structure and complexity, while some parameters are essential, others should be calculated automatically by inspecting the dataset prior to processing. The proposed Surface Reconstruction algorithm automatically calculates parameters by estimating a radius and defining a clear set of cluster constraints that need to be satisfied.

### Experiment Three

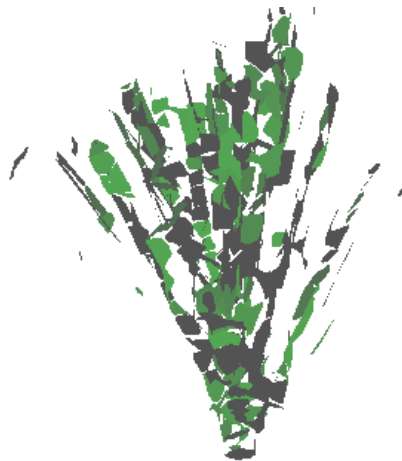
Experiment three is conducted on a Cordyline (*Corydline sp.*). The surface area of the X-ray CT scan is 30,572mm<sup>2</sup>.



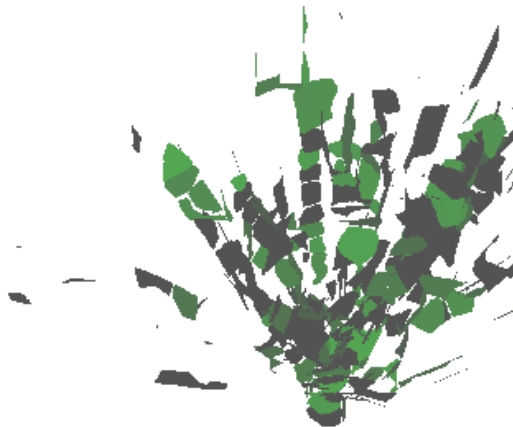
CT Scan – Side



CT Scan - Top



Canopy Reconstruction - Side



Canopy Reconstruction - Top



Surface Reconstruction – Side

Surface Reconstruction - Top

**PMVS to ground truth**

|           | $\mu$ (cm) | $\sigma^2$ | Points  | Images | Area   |
|-----------|------------|------------|---------|--------|--------|
| Canopy    | 6.85       | 5.32       | 166,903 | 40     | 23,882 |
| Surface R | 1.78       | 1.24       | 43,254  | 26     | 25,752 |

**Table 6.3** Experiment Three: evaluation of the proposed surface reconstruction algorithm applied to the AVC image set versus the canopy reconstruction algorithm (Pound et al. 2014) on a Cordyline (*Cordyline sp.*)

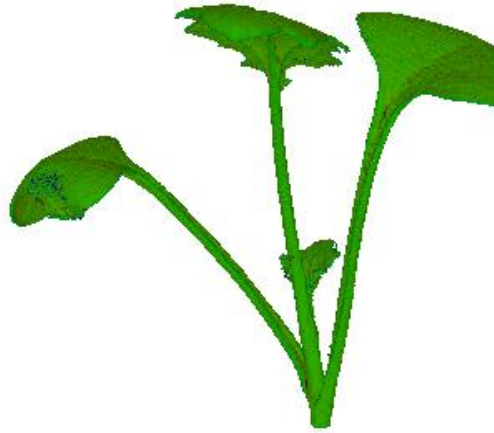
The Cordyline is particularly challenging due to the thin leaf areas. Moreover, the plant becomes increasingly dense towards the base, making the identification of single leaves exceptionally difficult. Level set processing becomes more challenging in situations like this due to a high level of occlusion as they rely on the projection of 3D points to the 2D images in which the cell is best seen. The AVC aids the level set method by obtaining a set of images that help overcome occlusion by intelligently selecting the views, ensuring that the object is adequately scanned.

Both algorithms struggle to produce a realistic model due to the complexity of the object, the lack of differences in shape and texture and the variation in density from top to bottom; ranging from relatively sparse to a dense connected region where all the components are connected. The Canopy Reconstruction and the Surface Reconstruction algorithms both have similar overall surface areas; however, the Canopy Reconstruction algorithm generates significantly more

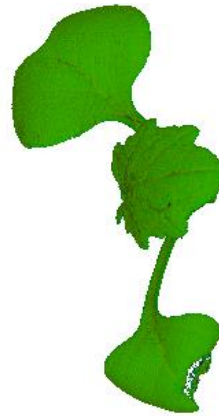
points with a much larger mean and standard deviation. The proposed Surface Reconstruction algorithm can produce a more accurate representation, having a lower mean and higher surface area with significantly less points (43,254 in comparison to 166,903).

#### Experiment Four

Experiment four is conducted on a Brassica (*Brassica napus*). The surface area of the X-ray CT scan is 2,007mm<sup>2</sup>.



CT Scan – Side



CT Scan - Top



Canopy Reconstruction - Side



Canopy Reconstruction - Top



Surface Reconstruction – Side



Surface Reconstruction – Top

| PMVS to ground truth |            |            |        |        |       |
|----------------------|------------|------------|--------|--------|-------|
|                      | $\mu$ (cm) | $\sigma^2$ | Points | Images | Area  |
| Canopy               | 0.48       | 0.39       | 90,132 | 40     | 1,610 |
| Surface R            | 0.32       | 0.24       | 76,246 | 21     | 1,556 |

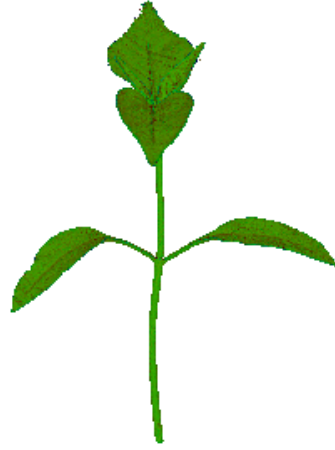
**Table 6.4** Experiment Four: evaluation of the proposed surface reconstruction algorithm applied to the AVC image set versus the canopy reconstruction algorithm (Pound et al. 2014) on a Brassica (*Brassica napus*)

The smaller mean and standard deviation via both methods in comparison to the previous experiments is the result of using a small plant. As shown in this example, the Canopy Reconstruction leads to a larger surface area, though the increase in mean and standard deviation, when compared to the proposed Surface Reconstruction algorithm, shows that it consists of incorrect surface areas. These areas can be seen in the images and are a result of the Canopy Reconstruction algorithm failing to remove the outliers.

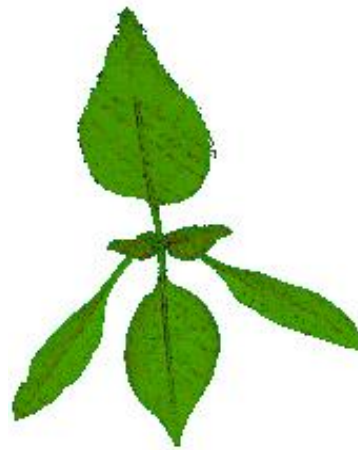
The Surface Reconstruction algorithm significantly improves the quality and accuracy of the plant model, producing a connected mesh, as opposed to a series of patches; the output of the Canopy Reconstruction algorithm. Consequently, it provides a more visually complete model and one that faithfully represents the original object. With respect to plant phenotyping, fully connected models are much more desirable due to their more realistic representations. Furthermore, connectivity of surfaces allows for more accurate modelling of light and photosynthesis dynamics or of canopy movement (Burgess et al. 2015; Burgess et al. 2017; Burgess et al. 2016; Townsend et al. 2018; Song et al. 2013).

### Experiment Five

Experiment five is conducted on a chill (*Capsicum sp.*). The surface area of the X-ray CT scan is 4,956mm<sup>2</sup>.



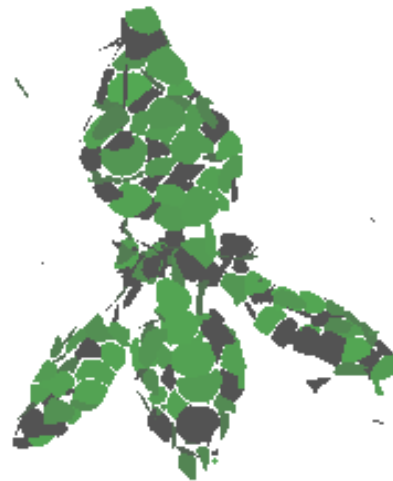
CT Scan – Side



CT Scan - Top



Canopy Reconstruction - Side



Canopy Reconstruction - Top



Surface Reconstruction – Side



Surface Reconstruction - Top

| PMVS to ground truth |            |            |        |        |       |
|----------------------|------------|------------|--------|--------|-------|
|                      | $\mu$ (cm) | $\sigma^2$ | Points | Images | Area  |
| Canopy               | 0.55       | 0.37       | 83,010 | 40     | 4,055 |
| Surface R            | 0.53       | 0.34       | 24,000 | 28     | 4,616 |

**Table 6.5** Experiment Five: evaluation of the proposed surface reconstruction algorithm applied to the AVC image set versus the canopy reconstruction algorithm (Pound et al. 2014) on a chilli (*Capsicum sp.*)

The Canopy Reconstruction algorithm again produces numerous outlying points, resulting in inaccurate and incorrect surface areas that are not part of the plant. The general model still clearly represents the target, but in comparison to the Surface Reconstruction algorithm, which again uses significantly less points, it provides a much less accurate representation.

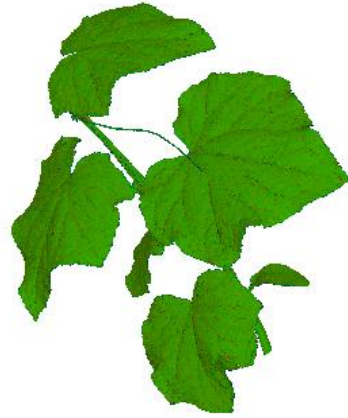
The Surface Reconstruction algorithm results in a larger and more accurate surface area, along with producing a fully connected mesh. By performing cell merging, it is possible to cover a greater proportion of the target object's surface, while the Canopy Reconstruction algorithm results in a series of triangulated clusters, each with a surrounding gap, consequently reducing the surface area and not faithfully representing the object. Moreover, the Surface Reconstruction algorithm uses approximately a quarter of the number of points that Canopy Reconstruction does, which significantly reduces computational requirements.

### Experiment Six

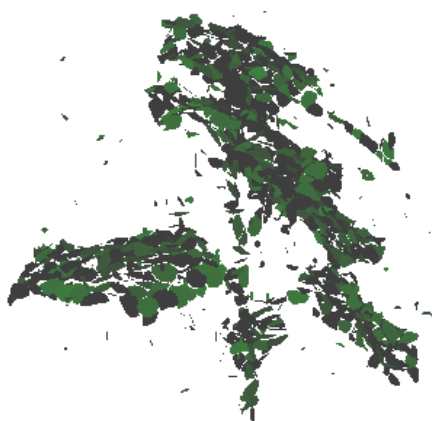
Experiment six is conducted on a pumpkin (*Cucurbita pepo*). The surface area of the X-ray CT scan is 124,677mm<sup>2</sup>.



CT Scan – Side



CT Scan - Top



Canopy Reconstruction - Side



Canopy Reconstruction - Top



Surface Reconstruction – Side



Surface Reconstruction - Top

| PMVS to ground truth |            |            |         |        |         |
|----------------------|------------|------------|---------|--------|---------|
|                      | $\mu$ (cm) | $\sigma^2$ | Points  | Images | Area    |
| Canopy               | 1.50       | 1.123      | 383,805 | 40     | 76,805  |
| Surface R            | 1.35       | 0.88       | 163,428 | 30     | 127,057 |

**Table 6.6** Experiment Six: evaluation of the proposed surface reconstruction algorithm applied to the AVC image set versus the canopy reconstruction algorithm (Pound et al. 2014) on a pumpkin plant (*Cucurbita pepo*)

The Canopy Reconstruction algorithm again is unable to deal with outliers and noise in the data. Despite this, the mean and standard deviation for each method are very similar. However, the Surface Reconstruction algorithm was able to estimate a much large proportion of the surface area.

In this instance, the Surface Reconstruction algorithm overestimates the total surface area of the plant. From the images of the reconstruction and from further visual inspection it can be seen that this is due to the occurrence of triangulations over leaf and base joints; i.e. the junctions between leaf and stem material. This is due to the rapid difference in size and structure, changing from wide flat services, the leaves, to narrow stems.

### Experiment Seven

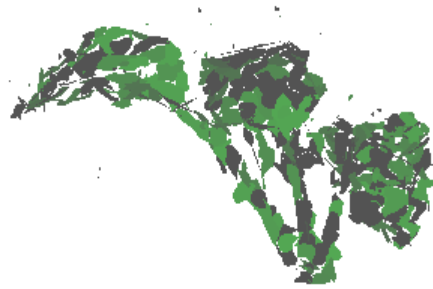
Experiment seven is conducted on a sugar beet (*Beta vulgaris*). The surface area of the X-ray CT scan is 39,625mm<sup>2</sup>.



CT Scan – Side



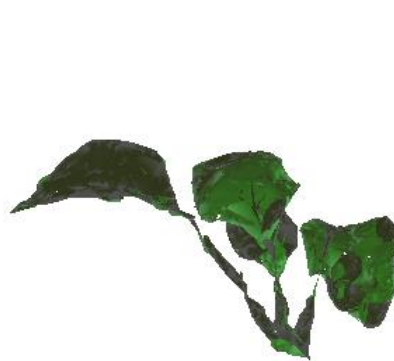
CT Scan - Top



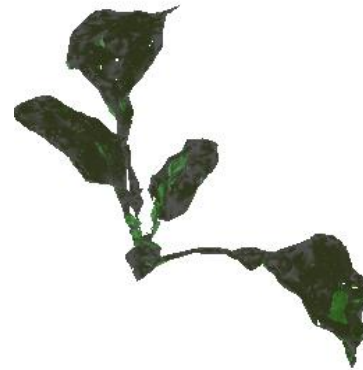
Canopy Reconstruction - Side



Canopy Reconstruction - Top



Surface Reconstruction – Side



Surface Reconstruction - Top

| PMVS to ground truth |            |            |         |        |        |
|----------------------|------------|------------|---------|--------|--------|
|                      | $\mu$ (cm) | $\sigma^2$ | Points  | Images | Area   |
| Canopy               | 2.58       | 2.47       | 211,438 | 40     | 25,699 |
| Surface R            | 0.70       | 0.70       | 24,642  | 30     | 20,580 |

**Table 6.7** Experiment Seven: evaluation of the proposed surface reconstruction algorithm applied to the AVC image set versus the canopy reconstruction algorithm (Pound et al. 2014) on Sugar beet (*Beta vulgaris*)

The Canopy Reconstruction algorithm leads to a higher area, yet a much larger mean and standard deviation; almost four times that of the Surface Reconstruction algorithm, which indicates a noisier and less accurate model representation. Canopy Reconstruction uses a green filter to determine which points to remove from the point set, as opposed to the Surface Reconstruction algorithm, which combines a volumetric model and point cloud to reduce outliers. The method proposed here, of merging models, as opposed to using a colour filter or background subtraction, makes it applicable to a much wider range of objects. Due to the complexity of the objects used in this work, numerous outliers in the point clouds are green due to erroneous matching in PMVS. The canopy reconstruction is unable to differentiate between correct and incorrect points, whereas the Surface Reconstruction algorithm can eliminate a larger number of outliers, thus reducing the mean and standard deviation. Consequently, the Canopy Reconstruction algorithm leads to a higher but less accurate surface area. The Surface Reconstruction algorithm can produce a fully connected mesh with few holes.

Error in each experiment typically occurs due to over estimation of area, as discussed in 4.7, and due to connectivity between close plant components. For example, when two stems overlap such that they touch it becomes challenging to determine which set of points belongs to which stem. Consequently, in some instances, additional triangles (surface) is placed along the two connecting stems, further increasing area estimates.

## 6.7 Discussion

In this chapter, the novel Surface Reconstruction algorithm is evaluated against the state-of-the-art. It is clear from the results that merging and adjusting

boundary points to truly reflect shape and curvature improves the accuracy of reconstructions. In all cases, the novel algorithm improves the reconstruction in comparison to existing algorithms in terms of accurate surface area. Moreover, in each of the cases, the number of images used is at least 10% less due to the removal of redundant images in the AVC - Surface Reconstruction method.

## **Chapter 7**

# **Conclusions and Future Work**

Chapter 7 provides a summary of the work presented in this thesis. The chapter begins with an overview of main contributions made in this thesis. Section two discusses the research outcomes compared to the original research questions in Chapter 1. Section three identifies targets for future improvement and applications for the proposed phenotyping pipeline. The chapter concludes with a short summary.

## 7.1 Main Contributions

This thesis has provided four important contributions:

- An active vision cell (AVC) designed to support recovery of 3D models of complex, crowded objects, such as plants. The cell integrates a fully calibrated robot arm, turntable and camera with a new and more efficient next best view (NBV) algorithm (Algorithm 4.1) that reduces the search space by using clusters of voxels combined with an incremental view sphere and utility function designed to support image capture for multi view stereo (MVS).
- A hybrid point clustering algorithm (Algorithm 5.2) comprised of an improved furthest reference point algorithm; a novel splitting algorithm based on differences of angles in 1D data from principal component analysis and the binning algorithm. An evaluation was performed against existing clustering algorithms.
- A hybrid surface reconstruction algorithm (Algorithm 6.1) combining volumetric and point-based representations and extending the work of (Pound *et al.*, 2014) to improve and more faithfully represent complex objects. The improvement in accuracy primarily results from the application of the two novel merging algorithms. The first is the merging of data performed between the volumetric model, 2D images and point cloud data producing more accurate data cells. The second is the merging of cells, clusters that have been triangulated, to produce a fully connected mesh. Moreover, the need to specify parameters is eliminated, instead parameters are estimated by automatically inspecting the dataset prior to processing.
- Each stage of the process is evaluated against a true ground truth model: generated from an X-ray CT scanner.

The combined output of the thesis is a powerful automated framework for capturing 3D representations of a target object from a set of 2D images. Noise and outliers in raw data are reduced using voxels and improved surface reconstruction is performed to produce a fully connected surface.

## 7.2 Research Outcomes

In Chapter 1, three research questions were proposed. To recap:

1. Can we improve the process, and output, of 3D reconstruction by manipulating 2D images, 3D data and interacting with the scene to obtain knowledge without human intervention? As opposed to relying solely on estimates obtained from mesh and point cloud data.
2. Can we significantly decrease the amount of data produced, reducing computational requirements, while maintaining the structural parameters of the object during the reconstruction process while maintaining the structural integrity of the object?
3. Can we automated methods based on active image acquisition, surface reconstruction and mesh-repairing to support robust and accurate plant phenotyping with respect to fast and accurate 3D models?

The rest of this section (Section 7.2) discusses how the work and algorithms proposed here address these questions.

In Chapter 3, the cell (robot, turntable and camera) is defined, calibrated and a simple interface is presented that allows for the manipulation and communication of all hardware components. Chapter 4 builds upon the work reported in Chapter 3 and the AVC is proposed. The AVC manipulates the position of the robot in world space by visual inspection of the cameras live feed and obtains initial object knowledge by constructing a volumetric model and evaluating this at voxel cluster level. Evaluation is performed via ray tracing from a view sphere to identify inadequately scanned object areas. From this evaluation, intelligent views of the object can be obtained, improving the quality of data and the output, with human interaction minimised to simply placing the target object on the turntable.

It is shown in Chapter 4 that the AVC produces more accurate data from interacting with the scene to intelligently determine camera positions and that this is a robust framework, able to adapt to different object types (in this case plant species), sizes and shapes.

In Chapter 5, model merging takes place. The volumetric model previously obtained is merged with the point cloud (Algorithm 5.1) to reduce the data requirements, remove noise and erroneous data whilst maintaining, and in some instances, improving, recovery of the structural properties of the object. This is as shown in the results of Chapter 5, where the mean distance to the ground truth is reduced relative to other methods. The remainder of Chapter 5 presents

clustering of the new dataset and compares the proposed clustering algorithm (Algorithm 5.2) to the state-of-the-art methods. The proposed clustering method performs at least equally as well as spectral clustering, but with much lower computational requirements. These improved clusters, in turn, support a more accurate reconstruction which more faithfully represents the object and its structure.

In Chapter 6, the triangulation of clusters to generate a series of cells is conducted. These cells are then repaired, smoothed and merged to produce a fully connected mesh. The results indicate that the proposed hybrid surface reconstruction algorithm (Algorithm 6.1) improves the accuracy of reconstructions while reducing the size of the dataset, the points, relative to the existing Canopy Reconstruction algorithm (Pound *et al.*, 2014). The improvement of 3D models helps support phenotyping, and the AVC that can adapt to plant shape and species, including those highly occluded, supports the robustness.

The research questions below have each been successfully addressed;

1. Can we improve the process, and output, of 3D reconstruction by manipulating 2D images, 3D data and interacting with the scene to obtain knowledge without human intervention?

The process has been improved, particularly with respect to plant phenotyping, by employing an AVC (Chapter 3 and 4). The output is more accurate as a result of merging (Chapter 5), clustering (Chapter 5) and the hybrid surface reconstruction algorithm (Chapter 6).

2. Can we significantly decrease the amount of data produced, reducing computational requirements, while maintaining an accurate representation of the structural parameters of the object during the reconstruction process?

By merging models (Chapter 4), data is significantly reduced while maintaining and, in some instances, improving the completeness and accuracy of the representation produced.

3. Can we automated methods based on active image acquisition, surface reconstruction and mesh-repairing to support robust and accurate plant phenotyping with respect to fast and accurate 3D models?

The full process, from gathering images (Chapter 4) to the reconstruction of a fully connected mesh (Chapter 6), is fully automated improving the output, and throughput, of surface reconstruction. The whole process is repeatable ensuring that if a plant was scanned multiple times, without moving it, the results would be the same; that is the same images would be captured and provided that the point cloud generated by PMVS is the same, the surface reconstruction will be too. This has been tested on multiple plants used in this thesis and in each instance the quantity and position in which the image was captured was consistent and the reconstructions produced areas and points within 99% of each other.

### 7.3 Future Work

From the work presented in this thesis, there are a number of both immediate- and long-term targets for improvement, plus extended applications of the proposed phenotyping module. These will be discussed in the following section.

#### **Calibration**

Calibration of the robot arm, turntable and camera is a challenging computer vision problem and the quality of reconstruction is highly dependent on the accuracy with which it is achieved. The resulting quality is particularly reliant on accurate calibration in this instance, where error is propagated through multiple hardware components. The algorithm proposed in this thesis (Section 3.3) produces accurate calibration matrices for each component. However, this accuracy is, in part, reliant on a ‘good’ set of images to solve the initial camera calibration, which relies on views of the checkerboard. Good views are those that see the checkerboard at distinct views, with sufficient variation in the angle, but not too much. At present, these views are defined manually by a user and although calibration only needs to be performed when the hardware setup has been physically moved, it is a long, and often challenging process to find views that minimise the error function.

Automatically determining the best viewpoints of the checkerboard, which could easily come as an extension from the AVC proposed here, would greatly increase the accuracy of calibration and reduce setup time, making the cell much easier to deploy.

#### **Technological Advances**

Volumetric models and surface reconstructions are both memory and processor intensive operations. Although a high-level volumetric model is not essential for this work, it could have proved to offer additional accuracy to the proposed model merging algorithm (Algorithm 5.1). With respect to the volumetric model, memory requirements were often a limitation to the depth and resolution. Access to larger memory allocations coupled with a more efficient design of octrees and the corresponding voxels would reduce memory requirements allowing for production of a higher-level volumetric model.

Furthermore, the reconstruction was performed on a machine with a standard CPU. With GPUs becoming more commonly available, the proposed reconstruction method, particularly the level set methods and projection to and from 2D images, would see a much-improved efficiency if ran on a GPU as opposed to a CPU. Consequently, an increased dataset of both images and points could be supported, improving the knowledge of the object and reducing erroneous data and producing a more faithful representation.

### **Generation of a high throughput phenotyping module**

The imaging and reconstruction pipeline presented in this thesis could be adapted for use in a larger scale phenotyping system. Image-based systems are highly desirable as a method of plant phenotyping; providing the information to calculate a number of key plant traits (Houle, Govindaraju and Omholt, 2010; White, Andrade-Sanchez, M. A. Gore, *et al.*, 2012; SANTOS and OLIVEIRA, 2013). Furthermore, 3D reconstruction methods have previously been shown to accurately represent key physiological measurements and accurately represent shoot traits such as leaf curling, shape and area or root traits such as morphology, geometry and topology (Lobet *et al.* 2011; Burgess *et al.* 2015; Burgess *et al.* 2017). This is because image-based reconstruction works by digitising existing plants, any structural differences found within the field- or pot- grown plants will be preserved in the final 3D model, resulting in higher accuracy relative to rule-based systems (Chapter 2). Image-based systems are becoming increasingly accessible due to the improvement and reduction in the cost of hardware, and the generation of improved software for processing data. Consequently, digitising plants for 3D modelling will become increasingly accessible. Furthermore, compared to other systems required for capturing plant structure (e.g. LiDAR; Chapter 2), cameras are easily portable and can be used in multiple different settings.

Parameters such as area, leaf length, plant height, leaf angle and yield could all be extracted automatically from the models obtained in this thesis. Deep neural networks can segment plant components, count leaves and ears and estimate yield. Full plant area is already an output of this thesis, however single plant component area and dimensions can be obtained by correctly partitioning the surface.

The AVC and associated pipeline presented in this thesis could form a reconstruction module within a higher throughput system. Multiple different sensors are employed within phenotyping systems to gather information on plant growth and function. These include: hyperspectral cameras, fluorescence and chlorophyll fluorescence cameras, near-infrared cameras and laser scanners. Different modules containing the different sensors can be combined together within one system to maximise the amount of information gathered.

Currently, the majority of the process is automated, but user interaction is required in order to place the plant on the turntable and remove it following imaging. Multiple large-scale phenotyping systems have been created in which plants are supplied to the sensors/ phenotyping modules via conveyor belts (Photon Systems Instruments, Drasov, Czech Republic; LemnaTec, Aachen, Germany; Qubit Phenomics, Kingston Ontario, Canada). Alternatively, within the University of Nottingham's Hounsfield Facility, plants are supplied to the X-ray CT scanner via an automated laser guided vehicle and automatic FANUC robotic arm.

Design of a high throughput phenotyping system remains a challenging problem with many systems still reliant on static or manual image capture. The approach here provides a strong framework upon which future systems can be built. Such a system will be invaluable as the demand for increased productivity of crops continues to increase over the coming years.

## 7.4 Summary

In recent years, plant phenotyping has received increased interest among computer scientists, as recognition of the research challenges it presents and its potential value to society spreads. As the need for accurate and robust high throughput plant phenotyping increases, so too will the demand for a robust reconstruction method capable of keeping up with the recent advances made in genotyping (i.e. the analysis of the genetic make-up of an organism).

This thesis proposed an AVC and surface reconstruction algorithm that provides a robust, flexible and accurate approach to automating 3D reconstruction of plants. The reconstruction algorithm can eliminate noise and provides a promising and extendable framework, improving current state-of-the-art, for high throughput phenotyping. Furthermore, the application of an active vision framework (Chapter 4) combined with the automatic detection of key parameters for surface reconstruction (Chapter 6), mean that the pipeline can be applied to any plant species or structure. The 3D models and 2D images generated via the proposed method are already being used for plant phenotyping with respect to feature tracking in two and three-dimensions.

# Bibliography

A. Hornung, B. Z. and L. K. (2008) 'Image selection for improved Multi-View Stereo', in *Computer Vision and Pattern Recognition*, pp. 1–8.

Adeloye, A. (2010) 'Global Warming Impact: Flood Events, Wet-Dry Conditions and Changing Scene in World Food Security', *Journal of Agricultural Research and Development*. Faculty of Agriculture, University of Ilorin, 9(1). doi: 10.4314/jard.v9i1.56128.

Aloimonos, J., Weiss, I. and Bandyopadhyay, A. (1988) 'Active vision', *International Journal of Computer Vision*. Kluwer Academic Publishers, 1(4), pp. 333–356. doi: 10.1007/BF00133571.

Anastacio, F. *et al.* (2006) 'Modeling plant structures using concept sketches', in *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering - NPAR '06*. New York, New York, USA: ACM Press, p. 105. doi: 10.1145/1124728.1124746.

Andersen, H. J., Reng, L. and Kirk, K. (2005) 'Geometric plant properties by relaxed stereo vision using simulated annealing', *Computers and Electronics in Agriculture*, 49(2), pp. 219–232. doi: 10.1016/j.compag.2005.02.015.

Arend, D. *et al.* (2016) 'Quantitative monitoring of Arabidopsis thaliana growth and development using high-throughput plant phenotyping', *Scientific Data*. Nature Publishing Group, 3, p. 160055. doi: 10.1038/sdata.2016.55.

Arvidsson, S., Pérez-Rodríguez, P. and Mueller-Roeber, B. (2011) 'Methods A growth phenotyping pipeline for Arabidopsis thaliana integrating image analysis and rosette area modeling for robust quantification of genotype effects', *New Phytologist*, 191, pp. 895–907. doi: 10.1111/j.1469-8137.2011.03756.x.

Baratoff, G., Neubeck, A. and Regenbrecht, H. (2002) 'Interactive Multi-Marker Calibration for Augmented Reality Applications'. IEEE Computer Society, p. 107. Available at: <http://dl.acm.org/citation.cfm?id=850976.854954> (Accessed: 11 May 2015).

Barnard, S. T. and Fischler, M. A. (1982) 'Computational Stereo', *ACM Computing Surveys*. ACM, 14(4), pp. 553–572. doi: 10.1145/356893.356896.

Berg, M. de. (2008) *Computational geometry: algorithms and applications*. Springer. Available at: <http://www.cs.uu.nl/geobook/> (Accessed: 16 August 2018).

Bingul, S. K. and Z. (2006) *Industrial Robotics: Theory, Modelling and Control*. Edited by S. Cubero. Pro Literatur Verlag, Germany / ARS, Austria. doi: 10.5772/44.

Biskup, B. *et al.* (2007) 'A stereo imaging system for measuring structural parameters of plant canopies.', *Plant, cell & environment*,

- 30(10), pp. 1299–308. doi: 10.1111/j.1365-3040.2007.01702.x.
- Blais, F. (2004) 'Review of 20 Years of Range Sensor Development', *Journal of Electronic Imaging*, 13(1), pp. 231–240.
- Boley, D. (1998) 'Principal Direction Divisive Partitioning', *Data mining and knowledge discovery*, 2(4), pp. 325–344. Available at: <http://www-users.cs.umn.edu/~boley/publications/papers/PDDP.pdf> (Accessed: 24 January 2018).
- Boudon, F. et al. (2003) 'Interactive Design of Bonsai Tree Models', *Computer Graphics Forum*, 22(3), pp. 591–599. doi: 10.1111/1467-8659.t01-2-00707.
- Boudon, F. et al. (2012) 'L-py: an L-system simulation framework for modeling plant architecture development based on a dynamic language.', *Frontiers in plant science*, 3, p. 76. doi: 10.3389/fpls.2012.00076.
- Brown, M. Z. et al. (2003) 'Advances in Computational Stereo', 25(8), pp. 993–1008.
- Bucksch, A. and Lindenbergh, R. (2008) 'CAMPINO — A skeletonization method for point cloud processing', *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(1), pp. 115–127. doi: 10.1016/j.isprsjprs.2007.10.004.
- Burgess, A. J. et al. (2015) 'High-Resolution Three-Dimensional Structural Data Quantify the Impact of Photoinhibition on Long-Term Carbon Gain in Wheat Canopies in the Field.', *Plant physiology*. American Society of Plant Biologists, 169(2), pp. 1192–204. doi: 10.1104/pp.15.00722.
- Burgess, A. J. et al. (2015) 'High-Resolution Three-Dimensional Structural Data Quantify the Impact of Photoinhibition on Long-Term Carbon Gain in Wheat Canopies in the Field', *Plant Physiology*, 169(2), pp. 1192–1204. doi: 10.1104/pp.15.00722.
- Burgess, A. J. et al. (2016) 'The 4-Dimensional Plant: Effects of Wind-Induced Canopy Movement on Light Fluctuations and Photosynthesis.', *Frontiers in plant science*. Frontiers Media SA, 7, p. 1392. doi: 10.3389/fpls.2016.01392.
- Burgess, A. J. et al. (2017) 'Exploring Relationships between Canopy Architecture, Light Distribution, and Photosynthesis in Contrasting Rice Genotypes Using 3D Canopy Reconstruction', *Frontiers in Plant Science*. Frontiers, 8, p. 734. doi: 10.3389/fpls.2017.00734.
- Burgess, A. J. et al. (no date) 'Image-based 3D canopy reconstruction to determine potential productivity in complex multi-species crop systems'. doi: 10.1093/aob/mcw242.
- Calin, G. and Roda, V. O. (2007) 'Real-time disparity map extraction in a dual head stereo vision system', *Latin American applied research*. Universidad Nacional del Sur, 37(1), pp. 21–24. Available at: [http://www.scielo.org.ar/scielo.php?script=sci\\_arttext&pid=S0327-07932007000100005&lng=es&nrm=iso&tlng=en](http://www.scielo.org.ar/scielo.php?script=sci_arttext&pid=S0327-07932007000100005&lng=es&nrm=iso&tlng=en) (Accessed: 18 November 2014).

Calin, G. and Roda, V. O. (2007) *REAL-TIME DISPARITY MAP EXTRACTION IN A DUAL HEAD STEREO VISION SYSTEM*, *Latin American Applied Research*. Available at: <https://pdfs.semanticscholar.org/eba0/31def4a4fe392f7d19eb2b77ecdc86a9ddc6.pdf> (Accessed: 16 August 2018).

Challinor, A. *et al.* (2014) 'A meta-analysis of crop yield under climate change and adaptation', *Nature Climate Change*. Nature Publishing Group. Available at: [http://eprints.whiterose.ac.uk/78340/13/Challinor-et-al-AR5-RevisionsFinal\\_with\\_coversheet.pdf](http://eprints.whiterose.ac.uk/78340/13/Challinor-et-al-AR5-RevisionsFinal_with_coversheet.pdf) (Accessed: 12 December 2014).

Connolly, C. (1985) 'The determination of next best views', in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers, pp. 432–435. doi: 10.1109/ROBOT.1985.1087372.

Cryer, J. E. and Shah, M. (1999) 'Shape-from-shading: a survey', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8), pp. 690–706. doi: 10.1109/34.784284.

DeJong, T. M. *et al.* (2011) 'Using functional-structural plant models to study, understand and integrate plant development and ecophysiology', *Annals of Botany*. Oxford University Press, 108(6), pp. 987–989. doi: 10.1093/aob/mcr257.

Denavit, J. (1955) 'A kinematic notation for lower-pair mechanisms based on matrices.', *Journal of Applied Mechanics*, 20, pp. 215–222.

Denzler, J., Zobel, M. and Niemann, H. (2003) 'Information theoretic focal length selection for real-time active 3D object tracking', in *Proceedings Ninth IEEE International Conference on Computer Vision*. IEEE, pp. 400–407 vol.1. doi: 10.1109/ICCV.2003.1238372.

Deussen, O. and Lintermann, B. (1997) 'A modelling method and user interface for creating plants', *Graphics interface*, 97, pp. 189–198.

Dhond, U. R., Aggarwal, J. K. (1989) 'Structure from Stereo - A Review', *IEEE Transactions on Systems, Man and Cybernetics*. IEEE Transactions on Systems, Man and Cybernetics, 19(6), pp. 1489–1510.

Dornaika, F. and Horaud, R. (1998) 'Simultaneous robot-world and hand-eye calibration', *IEEE Transactions on Robotics and Automation*, 14(4), pp. 617–622. doi: 10.1109/70.704233.

Evers, J. B. *et al.* (2005) 'Towards a generic architectural model of tillering in Gramineae, as exemplified by spring wheat (*Triticum aestivum*)', *New Phytologist*. Wiley/Blackwell (10.1111), 166(3), pp. 801–812. doi: 10.1111/j.1469-8137.2005.01337.x.

Faaij, A. (2008) *Bioenergy and global food security*. Available at: [http://www.wbgu.de/fileadmin/templates/dateien/veroeffentlichung/en/hauptgutachten/jg2008/wbgu\\_jg2008\\_ex03.pdf](http://www.wbgu.de/fileadmin/templates/dateien/veroeffentlichung/en/hauptgutachten/jg2008/wbgu_jg2008_ex03.pdf) (Accessed: 24 November 2014).

Fourcaud, T. *et al.* (2007) 'Plant Growth Modelling and Applications: The Increasing Importance of Plant Architecture in Growth Models',

*Annals of Botany*. Oxford University Press, 101(8), pp. 1053–1063. doi: 10.1093/aob/mcn050.

Fournier, C. *et al.* (2002) 'ADEL-Wheat: a 3D Architectural Model of wheat development'. Springer Verlag, pp. 54–63. Available at: <https://hal.archives-ouvertes.fr/hal-00909184/> (Accessed: 20 August 2018).

Furbank, R. T., & Tester, M. (2011) 'Phenomics – technologies to relieve the phenotyping bottleneck', *Trends in plant science*, 16(12), pp. 635–644. Available at: <https://www.plantphenomics.org.au/publications/2011TrendsFurbankTester.pdf> (Accessed: 1 November 2014).

Furbank, R. T. *et al.* (2009) 'C 4 rice: a challenge for plant phenomics', *Functional Plant Biology*. CSIRO PUBLISHING, 36(11), p. 845. doi: 10.1071/FP09185.

Furukawa, Y. and Ponce, J. (2010) 'Accurate, dense, and robust multiview stereopsis.', *IEEE transactions on pattern analysis and machine intelligence*, 32(8), pp. 1362–76. doi: 10.1109/TPAMI.2009.161.

Hartmann, A. *et al.* (2011) 'HTPheno: an image analysis pipeline for high-throughput plant phenotyping.', *BMC bioinformatics*, 12(1), p. 148. doi: 10.1186/1471-2105-12-148.

Horaud, R. and Dornaika, F. (1995) 'Hand-Eye Calibration', *The International Journal of Robotics Research*, 14(3), pp. 195–210. doi: 10.1177/027836499501400301.

Horn, B. K. P. and Brooks, M. J. (1989) 'Shape from shading'. MIT Press. Available at: <http://dl.acm.org/citation.cfm?id=93871> (Accessed: 8 January 2015).

Hosoi, F. and Omasa, K. (2006) 'Voxel-Based 3-D Modeling of Individual Trees for Estimating Leaf Area Density Using High-Resolution Portable Scanning Lidar', *IEEE Transactions on Geoscience and Remote Sensing*, 44(12), pp. 3610–3618. doi: 10.1109/TGRS.2006.881743.

Houle, D., Govindaraju, D. R. and Omholt, S. (2010) 'Phenomics: the next challenge', *Nature Reviews Genetics*. Nature Publishing Group, 11(12), pp. 855–866. doi: 10.1038/nrg2897.

Ivanov, N. *et al.* (1995) 'Computer stereo plotting for 3-D reconstruction of a maize canopy', *Agricultural and Forest Meteorology*, 75(1–3), pp. 85–102. doi: 10.1016/0168-1923(94)02204-W.

J. I. Vasquez-Gomez, L. E. S. and R. M.-C. (2014) 'View planning for 3D object reconstruction with a mobile manipulator robot', in *Intelligent Robots and Systems (IROS 2014)*, pp. 4227–4233.

Jackson, A. S. *et al.* (2017) 'Large Pose 3D Face Reconstruction from a Single Image via Direct Volumetric CNN Regression'. Available at: <http://arxiv.org/abs/1703.07834> (Accessed: 18 October 2017).

- Jin, J. and Tang, L. (2009) 'Corn plant sensing using real-time stereo vision', *Journal of Field Robotics*, 26(6-7), pp. 591-608. doi: 10.1002/rob.20293.
- Ju, T. (2009) 'Fixing Geometric Errors on Polygonal Models: A Survey', *Journal of Computer Science and Technology*. Institute of Computing Technology, 24(1), pp. 19-29. doi: 10.1007/s11390-009-9206-7.
- Juan Irving Vásquez, L. E. S. (2011) 'Next-Best-View Planning for 3D Object Reconstruction under Positioning Error', *Advances in Artificial Intelligence*, 7094, pp. 429-442.
- Junker, A. et al. (2015) 'Optimizing experimental procedures for quantitative evaluation of crop plant performance in high throughput phenotyping systems', *Frontiers in Plant Science*. Frontiers, 5, p. 770. doi: 10.3389/fpls.2014.00770.
- Karwowski, R. and Prusinkiewicz, P. (2003) 'Design and Implementation of the L+C Modeling Language', *Electronic Notes in Theoretical Computer Science*, 86(2), pp. 134-152. doi: 10.1016/S1571-0661(04)80680-7.
- Kender, J. R. (1981) 'Shape from texture'. Carnegie Mellon University. Available at: <http://dl.acm.org/citation.cfm?id=909911> (Accessed: 8 January 2015).
- Klasing, K., Wollherr, D. and Buss, M. (2008) 'A clustering method for efficient segmentation of 3D laser data', in *2008 IEEE International Conference on Robotics and Automation*. Ieee, pp. 4043-4048. doi: 10.1109/ROBOT.2008.4543832.
- Laurentini, A. (1994) 'The visual hull concept for silhouette-based image understanding', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2), pp. 150-162. doi: 10.1109/34.273735.
- Van Leeuwen, M., Coops, N. C. and Wulder, M. A. (2010) 'Canopy surface reconstruction from a LiDAR point cloud using Hough transform', *Remote Sensing Letters*. Taylor & Francis, 1(3), pp. 125-132. doi: 10.1080/01431161003649339.
- Li, A., Wang, L. and Wu, D. (2010) 'Simultaneous robot-world and hand-eye calibration using dual-quaternions and Kronecker product', *International Journal of Physical Sciences*. Academic Journals, pp. 1530-1536. Available at: <http://www.academicjournals.org/journal/IJPS/article-abstract/20DFAEA30999> (Accessed: 26 January 2015).
- Li, L. and Xiao, N. (2015) 'Volumetric view planning for 3D reconstruction with multiple manipulators', *Industrial Robot: An International Journal*. Emerald Group Publishing Limited, 42(6), pp. 533-543. doi: 10.1108/IR-05-2015-0110.
- Lindenmayer, A. (1968) 'Mathematical models for cellular interactions in development I. Filaments with one-sided inputs', *Journal of Theoretical Biology*, 18(3), pp. 280-299. doi: 10.1016/0022-5193(68)90079-9.
- Lintermann, B. and Deussen, O. (1996) *Interactive modelling of*

*branching structures.*

Livny, Y. *et al.* (2010) 'Automatic reconstruction of tree skeletal structures from point clouds', in *ACM SIGGRAPH Asia 2010 papers on - SIGGRAPH ASIA '10*. New York, New York, USA: ACM Press, p. 1. doi: 10.1145/1866158.1866177.

Lobet, G., Pagès, L. and Draye, X. (2011) 'A novel image-analysis toolbox enabling quantitative analysis of root system architecture.', *Plant physiology*. American Society of Plant Biologists, 157(1), pp. 29–39. doi: 10.1104/pp.111.179895.

Lowe, D. G. (1999) 'Object recognition from local scale-invariant features', in *Proceedings of the Seventh IEEE International Conference on Computer Vision*. IEEE, pp. 1150–1157 vol.2. doi: 10.1109/ICCV.1999.790410.

Masry, M. and Lipson, H. (2007) 'A sketch-based interface for iterative design and analysis of 3D objects', in *ACM SIGGRAPH 2007 courses on - SIGGRAPH '07*. New York, New York, USA: ACM Press, p. 31. doi: 10.1145/1281500.1281542.

Massios, Nikolaos A., and R. B. F. (1998) 'A best next view selection algorithm incorporating a quality criterion', *BMVC*.

Minguzzi, E. (2013) 'A geometrical introduction to screw theory', *European Journal of Physics*, 34(3), pp. 613–632. doi: 10.1088/0143-0807/34/3/613.

Mulayim, A. Y., Yilmaz, U. and Atalay, V. (2003) 'Silhouette-based 3-D model reconstruction from multiple images.', *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics: a publication of the IEEE Systems, Man, and Cybernetics Society*, 33(4), pp. 582–91. doi: 10.1109/TSMCB.2003.814303.

Neubert, B., Franken, T. and Deussen, O. (2007) 'Approximate image-based tree-modeling using particle flows', in *ACM SIGGRAPH 2007 papers on - SIGGRAPH '07*. New York, New York, USA: ACM Press, p. 88. doi: 10.1145/1275808.1276487.

Northend, C. A. (1967) 'Lidar, a laser radar for meteorological studies', *Die Naturwissenschaften*, 54(4), pp. 77–80. doi: 10.1007/BF00608760.

Ole, K. and Winfried, K. (2008) 'The Modelling Platform GroIMP and the Programming Language XL', *Applications of Graph Transformations with Industrial Relevance*, 5088, pp. 570–572.

Olsen, L. *et al.* (2009) 'Sketch-based modeling: A survey', *Computers & Graphics*, 33(1), pp. 85–103. doi: 10.1016/j.cag.2008.09.013.

Omasa, K., Hosoi, F. and Konishi, A. (2007) '3D lidar imaging for detecting and understanding plant responses and canopy structure.', *Journal of experimental botany*, 58(4), pp. 881–98. doi: 10.1093/jxb/erl142.

Osher, S. and Sethian, J. a (1988) 'Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi

formulations', *Journal of Computational Physics*, 79(1), pp. 12–49. doi: 10.1016/0021-9991(88)90002-2.

Paproki, A. et al. (2012) 'A novel mesh processing based technique for 3D plant analysis', *BMC plant biology*, 12(1), pp. 1–13.

Peterson, S. (1998) *Computing Constrained Delunay Triangulations*.

Phattaralerphong, J. and Sinoquet, H. (2005) 'A method for 3D reconstruction of tree crown volume from photographs: assessment with 3D-digitized plants', (Laurentini 1999), pp. 1229–1242.

Pound, M. P. et al. (2014) 'Automated recovery of three-dimensional models of plant shoots from multiple color images.', *Plant physiology*, 166(4), pp. 1688–98. doi: 10.1104/pp.114.248971.

Preuksakarn, C. et al. (2010) 'Reconstructing Plant Architecture from 3D Laser scanner data', in *Proceedings of the 6th International Workshop on Functional-Structural Plant Models*, pp. 12–17.

Prusinkiewicz, P. (2003) *Introduction to Modeling with L-systems*. University of Calgary. Available at: <http://algorithmicbotany.org/papers/sigcourse.2003/1-9-L-fundamentals.pdf> (Accessed: 10 November 2014).

Prusinkiewicz, P., Hanan, J. and Měch, R. (2000) 'An L-System-Based Plant Modeling Language', *Applications of Graph Transformations with Industrial Relevance*, 1779, pp. 395–410.

Quan, L. et al. (2006) 'Image-based plant modeling', *ACM Transactions on Graphics*. ACM, 25(3), p. 599. doi: 10.1145/1141911.1141929.

Rakocevic, M. (2000) 'Assessing the Geometric Structure of a White Clover (*Trifolium repens* L.) Canopy using 3-D Digitising', *Annals of Botany*, 86(3), pp. 519–526. doi: 10.1006/anbo.2000.1209.

Rao, R. P. N. and Ballard, D. H. (1995) 'An active vision architecture based on iconic representations', *Artificial Intelligence*, 78(1–2), pp. 461–505. doi: 10.1016/0004-3702(95)00026-7.

Reche-Martinez, A., Martin, I. and Drettakis, G. (2004) 'Volumetric Reconstruction and Interactive Rendering of Trees from Photographs', in *ACM Transactions on Graphics (ToG)*, pp. 720–727.

Reddy, S. (2016) 'Image Segmentation by Using Linear Spectral Clustering', *Journal of Telecommunications System & Management*. OMICS International, 05(03), pp. 1–5. doi: 10.4172/2167-0919.1000143.

Remondino, F. and El-Hakim, S. (2006) 'Image-based 3D Modelling: A Review', *The Photogrammetric Record*, 21(115), pp. 269–291. doi: 10.1111/j.1477-9730.2006.00383.x.

Rocha, C. R., Tonetto, C. P. and Dias, A. (2011) 'A comparison between the Denavit–Hartenberg and the screw-based methods used in kinematic modeling of robot manipulators', *Robotics and Computer-Integrated Manufacturing*, 27(4), pp. 723–728. doi:

10.1016/j.rcim.2010.12.009.

Rutledge, A. M. and Popescu, S. C. (2006) 'Using LiDAR in Determining Forst Canopy Parameters', in *ASPRS 2006 Annual Conference*. Available at: <http://www.asprs.org/a/publications/proceedings/reno2006/0130.pdf> (Accessed: 24 November 2014).

Sakaguchi, T. (1998) 'Botanical tree structure modeling based on real image set', in *ACM SIGGRAPH 98 Conference abstracts and applications on - SIGGRAPH '98*. New York, New York, USA: ACM Press, p. 272. doi: 10.1145/280953.282241.

SANTOS, T. T. and OLIVEIRA, A. A. de (2013) 'Image-based 3D digitizing for plant architecture analysis and phenotyping.' In: *CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES, 25., 2012, Ouro Preto. Workshop on industry applications*. [S.l.]: Conference Publishing Series, 2012. Available at: <https://www.alice.cnptia.embrapa.br/handle/doc/948383> (Accessed: 20 August 2018).

Sanz-Cortiella, R. *et al.* (2011) 'Innovative LIDAR 3D Dynamic Measurement System to estimate fruit-tree leaf area.', *Sensors (Basel, Switzerland)*. Molecular Diversity Preservation International, 11(6), pp. 5769–91. doi: 10.3390/s110605769.

Sarlikioti, V., de Visser, P. H. B. and Marcelis, L. F. M. (2011) 'Exploring the spatial distribution of light interception and photosynthesis of canopies by means of a functional-structural plant model', *Annals of Botany*. Oxford University Press, 107(5), pp. 875–883. doi: 10.1093/aob/mcr006.

Seitz, S. M. (2000) 'A Theory of Shape by Space Carving', *International Journal of Computer Vision*, 38(3), pp. 199–218.

Seitz, S. M. *et al.* (2006) 'A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms', in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)*. IEEE, pp. 519–528. doi: 10.1109/CVPR.2006.19.

Sethian, J. a. and Smereka, P. (2003) 'Level Set Methods for Fluid Interfaces', *Annual Review of Fluid Mechanics*, 35(1), pp. 341–372. doi: 10.1146/annurev.fluid.35.101101.161105.

Shah, M. (2013) 'Solving the Robot-World/Hand-Eye Calibration Problem Using the Kronecker Product', *Journal of Mechanisms and Robotics*. American Society of Mechanical Engineers, 5(3), p. 031007. doi: 10.1115/1.4024473.

Shah, M., Eastman, R. D. and Hong, T. (2012) 'An overview of robot-sensor calibration methods for evaluation of perception systems', in *Proceedings of the Workshop on Performance Metrics for Intelligent Systems - PerMIS '12*. New York, New York, USA: ACM Press, p. 15. doi: 10.1145/2393091.2393095.

Shiu, Y. and Ahmad, S. (1987) 'Finding the mounting position of a sensor by solving a homogeneous transform equation of the form  $AX = XB'$ ', in *Proceedings. 1987 IEEE International Conference on*

*Robotics and Automation*. Institute of Electrical and Electronics Engineers, pp. 1666–1671. doi: 10.1109/ROBOT.1987.1087758.

Shlyakhter I. et al. (2001) 'Reconstructing 3D Tree Models from Instrumented Photographs', *IEEE Computer Graphics and Applications*, 21(3), pp. 53–61.

Shum, H.-Y. and Kang, S. B. (2000) 'A Review of Image-based Rendering Techniques', in Ngan, K. N., Sikora, T., and Sun, M.-T. (eds) *Visual Communications and Image Processing 2000*. International Society for Optics and Photonics, pp. 2–13. doi: 10.1117/12.386541.

Sinoquet, H. and Rivet, P. (1997) 'Measurement and visualization of the architecture of an adult tree based on a three-dimensional digitising device', *Trees*, 11(5), p. 265. doi: 10.1007/s004680050084.

Song, Q., Zhang, G. and Zhu, X.-G. (2013) 'Optimal crop canopy architecture to maximise canopy photosynthetic CO<sub>2</sub> uptake under elevated CO<sub>2</sub>: a theoretical study using a mechanistic model of canopy photosynthesis', *Functional Plant Biology*. CSIRO PUBLISHING, 40(2), p. 108. doi: 10.1071/FP12056.

Sonohat, G. et al. (2002) 'Leaf dispersion and light partitioning in three-dimensionally digitized tall fescue-white clover mixtures', *Plant, Cell and Environment*. Wiley/Blackwell (10.1111), 25(4), pp. 529–538. doi: 10.1046/j.1365-3040.2002.00833.x.

Sticklen, M. B. (2007) 'Feedstock Crop Genetic Engineering for Alcohol Fuels', *Crop Science*. Crop Science Society of America, 47(6), p. 2238. doi: 10.2135/cropsci2007.04.0212.

Sutton, M. A. et al. (2011) *The European Nitrogen Assessment: Sources, Effects and Policy Perspectives*. Cambridge University Press. Available at: <http://books.google.com/books?hl=en&lr=&id=oX3oFqAM9GkC&pgis=1> (Accessed: 8 January 2015).

Tan, P. et al. (2007) 'Image-based tree modeling', *ACM Transactions on Graphics*. ACM, 26(3), p. 87. doi: 10.1145/1276377.1276486.

Tan, P., Yuan, L. and Wang, J. (2003) 'Image-based Plant Modeling Overview of Plant Modeling System', pp. 599–604.

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

Tasoulis, S. K. and Tasoulis, D. K. (2008) 'Improving Principal Direction Divisive Clustering', in *International Conference on Knowledge Discovery and Data Mining*. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.323.5881&rep=rep1&type=pdf> (Accessed: 24 January 2018).

Townsend, A. J. et al. (2018) 'Suboptimal Acclimation of Photosynthesis to Light in Wheat Canopies.', *Plant physiology*.

- American Society of Plant Biologists, 176(2), pp. 1233–1246. doi: 10.1104/pp.17.01213.
- Tsai, R. Y. and Lenz, R. K. (1989) 'A new technique for fully autonomous and efficient 3D robotics hand/eye calibration', *IEEE Transactions on Robotics and Automation*, 5(3), pp. 345–358. doi: 10.1109/70.34770.
- Vasquez-Gomez, J. I., Lopez-Damian, E. and Sucar, L. E. (2009) 'View planning for 3D object reconstruction', in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 4015–4020. doi: 10.1109/IROS.2009.5354383.
- Vasquez-Gomez, J. I., Sucar, L. E. and Murrieta-Cid, R. (2013) 'Hierarchical Ray Tracing for Fast Volumetric Next-Best-View Planning', in *2013 International Conference on Computer and Robot Vision*. IEEE, pp. 181–187. doi: 10.1109/CRV.2013.42.
- Vos, J. *et al.* (2010) 'Functional-structural plant modelling: a new versatile tool in crop science.', *Journal of experimental botany*, 61(8), pp. 2101–15. doi: 10.1093/jxb/erp345.
- Wahl, S. W. and F. M. (2001) 'Shape from 2D Edge Gradients', *Pattern recognition*, pp. 377–384. Available at: [http://download.springer.com/static/pdf/870/chp%253A10.1007%252F3-540-45404-7\\_50.pdf?auth66=1420725056\\_808c3f84dfd30d270e7b5f2ab0425b2d&ext=.pdf](http://download.springer.com/static/pdf/870/chp%253A10.1007%252F3-540-45404-7_50.pdf?auth66=1420725056_808c3f84dfd30d270e7b5f2ab0425b2d&ext=.pdf) (Accessed: 8 January 2015).
- Wang, H. *et al.* (2009) 'Image-based 3D corn reconstruction for retrieval of geometrical structural parameters', *International Journal of Remote Sensing*. Taylor & Francis, 30(20), pp. 5505–5513. doi: 10.1080/01431160903130952.
- Wang, L. L. and Tsai, W. H. (1991) 'Camera calibration by vanishing lines for 3-D computer vision', *IEEE Transactions on Pattern Analysis and Machine Intelligence*. IEEE Computer Society, 13(4), pp. 370–376. doi: 10.1109/34.88572.
- Watanabe, T. *et al.* (2005) 'Rice morphogenesis and plant architecture: measurement, specification and the reconstruction of structural development by 3D architectural modelling.', *Annals of botany*, 95(7), pp. 1131–43. doi: 10.1093/aob/mci136.
- Weber, J. and Penn, J. (1995) 'Creation and Rendering of Realistic Trees', in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, pp. 119–128.
- White, J. W., Andrade-Sanchez, P., Gore, M. a., *et al.* (2012) 'Field-based phenomics for plant genetics research', *Field Crops Research*. Elsevier B.V., 133, pp. 101–112. doi: 10.1016/j.fcr.2012.04.003.
- White, J. W., Andrade-Sanchez, P., Gore, M. A., *et al.* (2012) 'Field-based phenomics for plant genetics research', *Field Crops Research*. Elsevier, 133, pp. 101–112. doi: 10.1016/J.FCR.2012.04.003.
- Wold, S., Esbensen, K. and Geladi, P. (1987) 'Principal component analysis', *Chemometrics and Intelligent Laboratory Systems*. Elsevier,

2(1), pp. 37–52. doi: 10.1016/0169-7439(87)80084-9.

Wong, L. M., Dumont, C. and Abidi, M. A. (1998) 'Next-best-view algorithm for object reconstruction', in Schenker, P. S. and McKee, G. T. (eds) *Photonics East (ISAM, VVDC, IEMB)*. International Society for Optics and Photonics, pp. 191–200. doi: 10.1117/12.327001.

Wong, L. M., Dumont, C. and Abidi, M. A. (1999) 'Next best view system in a 3D object modeling task', in *Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation. CIRA'99 (Cat. No.99EX375)*. IEEE, pp. 306–311. doi: 10.1109/CIRA.1999.810066.

Zhang, Z. (2000) 'A Flexible New Technique for Camera Calibration', *IEEE Pattern Analysis and Machine Intelligence*, 22(11), pp. 1330–1334.

Zhang, Z. (2003) 'Camera Calibration', in. Available at: <https://pdfs.semanticscholar.org/6e20/c43a0077d6580975625c44411e8c3fcf3ffe.pdf> (Accessed: 16 August 2018).

Zhang, Z. (2004) 'Camera Calibration', in *Emerging Topics in Computer Vision*. Prentice Hall PTR, pp. 4–43. Available at: <http://dl.acm.org/citation.cfm?id=993884> (Accessed: 5 December 2014).

Zheng, Y. F. (1995) 'Passive and active stereo vision for smooth surface detection of deformed plates', *IEEE Transactions on Industrial Electronics*, 42(3), pp. 300–306. doi: 10.1109/41.382141.

Zolfaghari, H. and Khalili, K. (2008) 'On-Line 3D Geometric Model Reconstruction', in, pp. 146–152. doi: 10.1007/978-3-540-69387-1\_16.