Western Graduate&PostdoctoralStudies

Western University
Scholarship@Western

Electronic Thesis and Dissertation Repository

August 2017

Computer Vision Problems in 3D Plant Phenotyping

Ayan Chaudhury
The University of Western Ontario

Supervisor
Prof. John Barron
The University of Western Ontario

Graduate Program in Computer Science

A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy

© Ayan Chaudhury 2017

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Part of the Artificial Intelligence and Robotics Commons, and the Other Computer Sciences Commons

Recommended Citation

Chaudhury, Ayan, "Computer Vision Problems in 3D Plant Phenotyping" (2017). *Electronic Thesis and Dissertation Repository*. 4789.
https://ir.lib.uwo.ca/etd/4789

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact tadam@uwo.ca.

Abstract

In recent years, there has been significant progress in Computer Vision based plant *phenotyping* (quantitative analysis of biological properties of plants) technologies. Traditional methods of plant *phenotyping* are destructive, manual and error prone. Due to non-invasiveness and non-contact properties as well as increased accuracy, imaging techniques are becoming state-of-the-art in plant *phenotyping*. Among several parameters of plant *phenotyping*, growth analysis is very important for biological inference. Automating the growth analysis can result in accelerating the throughput in crop production. This thesis contributes to the automation of plant growth analysis.

First, we present a novel system for automated and non-invasive/non-contact plant growth measurement. We exploit the recent advancements of sophisticated robotic technologies and near infrared laser scanners to build a 3D imaging system and use state-of-the-art Computer Vision algorithms to fully automate growth measurement. We have set up a gantry robot system having 7 degrees of freedom hanging from the roof of a growth chamber. The payload is a range scanner, which can measure dense depth maps (raw 3D coordinate points in mm) on the surface of an object (the plant). The scanner can be moved around the plant to scan from different viewpoints by programming the robot with a specific trajectory. The sequence of overlapping images can be aligned to obtain a full 3D structure of the plant in raw point cloud format, which can be triangulated to obtain a smooth surface (triangular mesh), enclosing the original plant. We show the capability of the system to capture the well known diurnal pattern of plant growth computed from the surface area and volume of the plant meshes for a number of plant species.

Second, we propose a technique to detect branch junctions in plant point cloud data. We demonstrate that using these junctions as feature points, the correspondence estimation can be formulated as a subgraph matching problem, and better matching results than state-of-the-art can be achieved. Also, this idea removes the requirement of a priori knowledge about rotational angles between adjacent scanning viewpoints imposed by the original registration algorithm for complex plant data. Before, this angle information had to be approximately known.

Third, we present an algorithm to classify partially occluded leaves by their contours. In general, partial contour matching is a NP-hard problem. We propose a suboptimal matching solution and show that our method outperforms state-of-the-art on 3 public leaf datasets. We anticipate using this algorithm to track growing segmented leaves in our plant range data, even when a leaf becomes partially occluded by other plant matter over time.

Finally, we perform some experiments to demonstrate the capability and limitations of the system and highlight the future research directions for Computer Vision based plant *phenotyping*.

Keywords: Robotic Imaging, 3D Plant Growth, Multiview Reconstruction, Diurnal Plant Growth, *phenotyping*, Junction Point Detection, Correspondence Estimation, Leaf Classification, Partial Occlusion, Subgraph Matching, Beta Spline, Energy Optimization

Co-authorship statement

Chapter 1 and **Chapter 2** are my own work on the introduction to the thesis and the summary of the background/related literature.

Chapter 3 is a joint work with Mark Brophy and our common supervisor John Barron. Mark Brophy proposed the main idea of combining Coherent Point Drift (CPD) & Mutual Nearest Neighbour (MNN) algorithms to perfrom registration of range plant scans. I explored state-of-the-art algorithms for point set registration and performed a quantitative comparison on plant data. He is the primary author of the relevant paper.

Chapter 4 is a collaborative work with different people. I closely collaborated with Christopher Ward for the robot operations. He handled the robot kinematics part and we integrated the system together. Ali Talasaz worked on the project at the very beginning and made the initial trajectory. Alex Ivanov helped in the issues with the Biochamber and in getting the plants. Norm Hüner, Rajni Patel and John Barron supervised the overall project. I performed all the analysis and generated all the results. I am the primary author of the conference and journal version of the relevant papers.

Chapter 5 is a joint work with my supervisor John Barron. I proposed the idea of using junctions as feature points. We worked on the technical ideas together. I wrote the code, performed the experiments and (mostly) wrote the paper.

Chapter 6 is a joint work with my supervisor John Barron. He suggested that partially occluded leaf matching can be a good problem to work on as leaf tracking over time is an important plant *phenotyping*. I explored the background of contour based shape matching, experimented with the state-of-the-art algorithms on leaf database and identified their limitations. We developed the technical ideas together, I implemented the code and did most of the paper writing.

Chapter 7 is my work on summarizing the whole thesis and pointing out the possible future research directions.

Dedicated to my super amazing parents for everything...

Acknowledgements

First of all I would like to thank my supervisor Prof. John Barron for supervising me. He has been a great mentor to support & believe in me through a tough journey. The way he has listened seriously to my problems & ideas in our weekly meetings, provided constructive feedback, engaged in arguments, & most importantly treated me as a equal friend, it's been simply amazing. I should also mention that I have hardly seen a professor like him who really cares about his students. Thanks John for being so nice and giving me so much freedom!

I am grateful to Sharcnet computing facilities to run the massive datasets. I would like to thank my thesis examiners, Michel Jenkin, Graham Denham, Dan Lizotte & Mike Bauer for their comments & suggestions. Thanks to Steven Beauchemin for offering some wonderful courses. Thanks my collaborators: Mark Brophy, Christopher Ward, Ali Talasaz, Alex Ivanov, Bernard Grodzinski, Norm Hüner and Rajni Patel. Special thanks to Chris, without his collaboration, things would have never been easy. Thanks to Leon & Golnaz for the plants. Thanks to all the CS office staffs: Janice, Cheryl & Dianne for all the help.

I had a great time here surrounded by many friends in the department. Special mention to Manasi & Roopa. I will never forget my surprise birthday celebration with them and all the hangouts that I will miss. Last few years I have been surrounded by so many Persians, now I've learnt quite a few Persian phrases! ;) Thanks to the Iranian club: Ali, Besat, Mahboubeh, Nasim, Taraneh & Parisa among many others. MerC. Thanks to my recent lab-mates, Linda, Duff & Seereen. We really talked a lot in the short span! Rallis, I will never forget the "4th September", "3^3 day", the "Masonville ice-cream" & our other ventures! BTW, is there any Japanese restaurant left in London where we haven't been? ;)

I am grateful to Prof. Amlan Chakrabarti of University of Calcutta, whose guidance and positive encouragement helped me to do whatever I have done today. I am also grateful to Prof. Ujjwal Bhattacharya of Indian Statistical Institute. Thanks to Mr. Amitava Roy of Sarangsoft for being a great teacher and a motivator.

I am also indebted to some of my friends. Subhajit, hope to continue our "quality discussion" over Skype. Arijit, although are we no longer in contact, but I still consider you as a good friend. Sukanta, thanks for being there 24x7, hope to continue our discussion on 0.1% ;)

Grad school has been an amazing experience. I've almost become a full time traveller and a long distance swimmer! These years have been some of the BEST days of my life. However, things have been very tough for me in graduate school. There have been times, when I became alcoholic and isolated due to repeated turmoil in my academic & personal life. But all these years helped me to get matured as a person. I think while doing the 'doctor of philosophy', I have realized the philosophy of life at the deepest level (& I mean it).

I miss my grandfather Mrinal Kanti Choudhury, he would have been the happiest person today. Finally I've got a chance to talk about the most important persons of my life, my parents: Supriya Chaudhury and Sipra Chaudhury. Whatever I've done & whatever I will do in future, are because of them. I could never figure out how come they can be so perfect in supporting me through every stage of life. Especially my mother, I have no idea how come someone can be so perfect in every aspect! If there is a textbook definition of ideal parents, then I would say that they have been beyond that. I feel so proud to be their son. If there is any such thing called reincarnation, then I would want them to be my parents over & over again. I dedicate all of my works and everything to my parents.

Contents

Abstract **i**

Co-authorship Statement **ii**

Acknowledgements **iv**

List of Figures **viii**

List of Tables **xi**

1 Introduction **1**
- 1.1 Role of Computer Vision in Plant *phenotyping* 1
- 1.2 Thesis Contributions 2
  - 1.2.1 Machine Vision System 3
  - 1.2.2 Multiview Alignment 3
  - 1.2.3 Rough Initial Alignment 3
  - 1.2.4 Partially Occluded Leaf Classification 4
- 1.3 Relevant Publications 5
- 1.4 Thesis Outline 5

2 Literature Survey **7**
- 2.1 Computer Vision Approaches for Plant *phenotyping* 7
- 2.2 MSc Theses from Barron's Lab 9
- 2.3 Multi-View Alignment 11
- 2.4 Correspondence Estimation 12
- 2.5 Contour based Shape Matching 13
  - 2.5.1 Contour Based Leaf Matching 15

3 Multi-View Alignment **16**
- 3.1 Introduction 16
- 3.2 Gaussian Mixture Model based registration 17
  - 3.2.1 Quantitative Comparison of State-Of-The-Art 19
- 3.3 Proposed Method 20
  - 3.3.1 Approximate Alignment 22
  - 3.3.2 Global Non-Rigid Registration via MNN 22
    - Global Registration 22
- 3.4 Results 24

3.4.1 Plant Data 24
3.4.2 Synthetic Vascular Data 24

4 Machine Vision System for 3D Plant Growth Analysis 27
4.1 Introduction 28
4.2 System Description 29
4.2.1 Gantry Robot 29
4.2.2 Scanner 32
4.2.3 Growth Chamber 32
4.3 Alignment of multi-view scans 32
4.3.1 Feature Clustering 33
4.4 System Integration 34
4.5 Experimental Results 36
4.5.1 Merging of Multi-View Plant Point Cloud 37
4.5.2 Polygonal Mesh Formation 37
α-Shape Triangulation 37
4.5.3 Biological Relevance 39
4.6 Limitations 42
4.6.1 Conifer Experiment 43
4.6.2 Bean plant experiment 46

5 Junction Based Correspondence 48
5.1 Introduction 48
5.2 Coordinate Transformation 49
5.2.1 Coordinate Transformation: 3D to 2D 50
Rotate $\hat{u}$ into the xz-plane 51
Align $\hat{u}_{xz}$ along z-axis 53
5.3 DIP Test for Multi-Modality 54
5.4 RANdom SAmple Consensus (RANSAC) fitting and Total Least Squares (TLS) approximation 54
5.5 Correspondence Matching 57
5.5.1 Graph Formation 57
5.5.2 Sub-graph Matching Formulation 57
5.5.3 Matching of Nodes 58
5.6 Experimental Results 58

6 Partially Occluded Leaf Matching 62
6.1 Introduction 62
6.2 Algorithm Overview 63
6.3 Our Approach 63
6.3.1 β-Spline Based Representation 64
6.4 Approximate Curve Section From Full Leaf 65
6.4.1 Subgraph Matching 66
6.4.2 Inverse Affine Transform 69
6.4.3 Curve matching by *Fréchet Distance* 71

6.4.4 Objective Function 72
6.5 GNCCP optimization 76
6.6 Experimental Results 77
6.6.1 Swedish dataset 79
6.6.2 Flavia dataset 79
6.6.3 Leafsnap dataset 80
6.6.4 Discussion 80

7 Conclusion and Future Work **84**
7.1 Conclusion 84
7.2 Future Work 84

Bibliography **86**

Curriculum Vitae **98**

List of Figures

2.1 The experimental setup used for scanning the Arabidopsis plants. 9

3.1 Pairwise registration results for two scans of the Stanford Bunny. Top row: original two views, Bottom row: results obtained using Rusu et al. (FPFH) [118], Fitzgibbon (LMICP) [52], Jian and Vemuri (GMMReg) [69] and Myronenko and Song (CPD) [101]. 19
3.2 Pairwise registration results two scans of the real Arabidopsis plant data. Top row: the original two views. Bottom row: results obtained using Rusu et al. [118], Fitzgibbon [52] and Myronenko and Song [101]. 20
3.3 12 scans of the Arabidopsis plant, prior to registration, but with rotation and and translation pre-applied. Different colours indicate different scans. 22
3.4 12 scans captured in 30° increments about the plant and then merged into a single point cloud using MNN. Shown from two viewpoints, front facing on the left and from above on the right. Different colours indicate different scans. . 23
3.5 MNN versus sequential pairwise registration. 24
3.6 20 synthetic scans of vascular data merged into a single point cloud using MNN, as seen from two viewpoints, front and back. 25
3.7 MNN versus sequential pairwise registration on vascular data, registering the first scan to each of the subsequent scans. 25

4.1 Schematic diagram of the gantry robot system 28
4.2 Robot arm 30
4.3 Robot room where the experiment was performed 30
4.4 Different scanning positions of the robot arm around the plant 31
4.5 scan head distance from the plant 32
4.6 Demonstration of feature clustering. Red dots represent junction feature points [29]. Density clustering algorithm [45] detects clusters at true junctions (denoted by blue circle) and treats false feature points as outliers. 34
4.7 High level view of the system 35
4.8 Computing the bounding box to determine the center of the plant 36
4.9 Scanner Field of View (FOV) to enclose the whole plant. When the plant is big, one scan can't capture the whole plant and multiple scans are required . . . 36
4.10 System GUI or user interface 37
4.11 Reconstructed Arabidopsis plant point cloud (different colors indicate different scans). Note that there was no plant jittering in our setup as the wind could be fully controlled, unlike for the setup used in Brophy's work (described in Chapter 3). 38

4.12 Reconstructed Barley plant point cloud (different colors indicate different scans) 39
4.13 Triangulated Arabidopsis plant data 40
4.14 Triangulated Barley plant data 41
4.15 Diurnal growth pattern of mesh surface area and volume for the Arabidopsis plant. The red dots represent night time scans, the blue dots represent day time scans and the four green dots represent missing scan data. A spline is fitted to both surface and volume scan data (shown in different colours). The y-axis in the left and right hand side represents the range of surface area and volume data respectively. 42
4.16 Diurnal growth pattern of mesh surface area and volume for the barley plant. The red dots represent night time scans while the blue dots represent day time scan data. A spline is fitted to both surface and volume scan data (shown in different colours). The y-axis in the left and right hand side represents the range of surface area and volume data respectively. 43
4.17 (a) and (c): the surface growth rate lines and (c) and (d): the volume growth rate lines for the Arabidopsis and barley plants. 44
4.18 Reconstruction of a conifer plant (different colors indicate different scans). . . . 45
4.19 Growth data for the conifer plant. The conifer as scanned at 14:00 in the afternoon (light) and at 2:00 in the night (dark). 45
4.20 Volume growth rate for the conifer plant 46
4.21 Two adjacent views of a bean plant 46
4.22 Reconstruction result of bean plant (different colours indicate different scans) . 47

5.1 Planar vector orientations 50
5.2 Rotate about V 51
5.3 Rotate $\hat{u}$ about xz-plane. Left: Project $\hat{u}$ onto the $y - z$ plane as $\hat{u}'$. Right: $\hat{u}'$ is rotated by α about the x axis onto the $\hat{k}$ axis. 52
5.4 Aligning $\hat{u}$ along z-axis. Left: $\hat{u}$ is rotated by α about the x axis onto the $x - z$ plane as $\hat{u}_{xz}$. Right: $\hat{u}_{xz}$ is rotated by β about the y-axis onto the $\hat{k}$ axis as $\hat{u}_z$. . . 53
5.5 Distribution of data: column 1 are the neighbourhood point clouds under consideration, column 2 are the histograms of the x coordinates distribution and column 3 are the histograms of the y coordinate distribution for a single stem (first row), a leaf (second row) and a stem with 2 branches (last row). The later is potentially a junction point. 55
5.6 Examples of detected junction points (red dots) on the real Arabidopsis plant data 59
5.7 Two examples of correspondence matching (green lines are good, red lines are bad matches which are eliminated by RANSAC) 59
5.8 The point to point correspondence results obtained from (a) CPD and (b) GMM for large rotation angle. Good matches are shown in green lines and bad matches are in red lines. 60
5.9 Effect of rotation angle: The rotation angle (in degrees) is plotted against (normalized) accuracy. 61
5.10 Precision-Recall curve comparing CPD, GMM, SISI and our method. 61

6.1 (Left) original unsmoothed leaf contour and (right) smoothed leaf contour. . . . 66

6.2 Feature points obtained from Discrete Contour Evolution (DCE) method [79]. The contour is decomposed into meaningful parts based on the convexity and the feature points represent the joint of these parts (or the vertices of the polygon represent the shape). 67
6.3 Four examples of graph matching. An occluded curve section is matched with 4 full curves in the database (the left curve is the occluded curve and the right curve is the full curve). In examples (a), (b) and (c), an occluded curve section is matched with full curves from the same species. In example (d), the same occluded curve is matched with a different species. Good matches are shown as solid green lines while ambiguous matches are shown as dotted green lines. . 68
6.4 (a) and (b) the correspondence of some points on 2 curves, (c) and (d) the registration (warping) of the 2 curves and (e) and (f) the 2 curves overlaid via the computed affine parameters (rotation, translation and scale). 70
6.5 The idea of "StringCut". Some samples of neighbourhood around a point is shown. A straight line is drawn between the first and the last point of the neighbourhood and the distribution of the points around the straight line are used as StringCut features. 74
6.6 Fréchet errors, F, and energy optimization values, E, for a number of curve sets. Curves (a) and (b) have both low F and E values, curves (c) and (d) have low F and high E values (showing the two are not always i well correlated) and curves (e) and (f) have both high F and E values. 75
6.7 Some typical occluded leaves with varying contour occlusion levels (specified below each image as a percentage). The part of each leaf enclosed by a red contour is cut away. The partial leaf contour remaining (not including the contour the cut away leaf part makes with the leaf) is the open occluded curve. 78
6.8 Leaves with similar contours for different species: leaves from (a) Sycamore Maple (acer pseudoplatanus) and Red Maple (acer rubrum), (b) Sweet Birch (betula lenta), Whitebark Himalayan Birch (betula jacquemontii) and Paper Mulberry (broussonetxtia papyrifera), (c) Southern Catalpa (catalpa bignonioide) and Northern Catalpa (catalpa speciosa) and (d) Yellow Birch (betula alleghaniensis), Downy Serviceberry (amelanchier arborea) and European Hornbeam (carpinus betulus). Note that we only classify these leaves based on their contour shapes and not on their texture, even though some of the species are clearly distinguishable by their texture. 79
6.9 Precision-Recall curves for the Swedish dataset with (a) no occlusion, (b) 25% occlusion, (c) 50% occlusion and (d) average performance for random occlusion levels from 20% to 50% respectively. 81
6.10 Precision-Recall curves for the Flavia dataset with (a) no occlusion, (b) 25% occlusion, (c) 50% occlusion and (d) average performance for random occlusion levels from 20% to 50% respectively. 82
6.11 Precision-Recall curves for the Leafsnap dataset with (a) no occlusion, (b) 25% occlusion, (c) 50% occlusion and (d) average performance for random occlusion levels from 20% to 50% respectively. 83

List of Tables

3.1 Quantitative results for different algorithms and datasets. 19

4.1 Robot Specifications 29

Chapter 1

Introduction

With increasing world population, the production of crops for food and energy is more important than ever. There is a great need for automation in analyzing the output of agricultural/crop monitoring systems. Quantitative growth analysis is a very fundamental task in Plant Sciences. Such an analysis might allow a feedback loop, where plant growth might control the plant's treatment (for example, watering and fertilizing and maybe as well, for indoor controlled environments, the temperature and lighting). Accurate measurement and analysis of a plant's growth is a challenging task and is extremely important for real life applications. The naive method used by biologists (by means of measuring weight, area or volume) is invasive and destructive, thus hinders the measurement of this data over time. Manual measurements can be time consuming and erroneous. In recent years, autonomous and accurate real time plant *phenotyping* has been a quintessential part of modern crop monitoring and agricultural technologies. Non-invasive/non-contact analysis of plants is highly desirable in Plant Sciences research because traditional techniques usually require the destruction of the plant, thus hindering the analysis of the plant's growth over its life-cycle. Machine vision systems can allow us to monitor, analyze and produce high throughput in an autonomous manner without any manual intervention.

1.1 Role of Computer Vision in Plant *phenotyping*

Among several parameters of plant *phenotyping*, growth analysis is very important for biological inference. *Functional analysis* of growth curves can reveal many underlying functionalities of the plant [38]. A plant's growth pattern can reveal different biological properties in different environmental conditions. For example, a leaf's elevation angle is impacted by the amount of sunlight [42]. In direct sunlight, leaves exhibit more elongation than when the plant is shaded. Similar kinds of behaviour are exhibited for rosette size, stem height, surface area and volume. Imaging techniques are very effective in terms of non-invasive/non-contact and accurate analysis. While 2D imaging techniques have been used extensively in the literature, they have some inherent limitations. The advantage of 3D over 2D are numerous. For example, consider the area of a leaf. If the leaf is curved, the 3D area will be significantly different from the area computed from its 2D image. Another restriction of 2D is that it is often difficult to measure 3D quantities such as the surface area or the volume of a plant without doing error prone and potentially complex calculations, such as a stereo disparity calculation from stereo images to get 3D depth information that is a necessary precursor to these calculations. Recently, 3D laser scanners are being used in many applications for studying plant *phenotyping*.

This thesis contributes to the recent advancements of Computer Vision based plant *phenotyping* technologies. Broadly, we focus on two problems in this thesis:

1. automated 3D growth analysis of living plants and

2. matching partially occluded leaves with databases of leaves.

Growth analysis of plants is a well studied problem in plant biology. A plant's growth can be measured in different ways. For example, stem height, plant width, leaf area, etc. can be used as growth metrics of a plant. However, none of these parameters can solely be relied on for the growth analysis of any plant. For example, stem diameter is shown to be a good growth metric [55, 128, 56] for certain plants. On the other hand, plant height and stem diameter is shown to have a good correlation in growth analysis for certain plants [139]. There can be many other metrics for growth analysis of different plants. If certain species of plants exhibit growth by their stem diameter, any attempt to quantify growth by their heights will not yield desired results. Similarly, if the plant grows by its width (for example, during the vegetative stage *Arabidopsis thaliana* mostly grows leaves, the stem starts shooting up after few weeks), stem height will not be an appropriate parameter for the growth analysis. However, even if different plants grow in different ways, surface area and/or volume of the plant will change accordingly, and that is why these parameters can be considered more reliable to quantify the plant growth in general. We can use Archimedes' principle to measure the volume of a plant. Dipping a plant into the water and measuring the change of water level has been a standard technique for volume measurement for a long time. But there are some inherent problems with this approach. First, the technique is destructive. One cannot study the growth analysis of a plant during its lifetime. Second, the process is prone to error. When the plant is dipped into the water, there will always remain some water bubbles which will affect the true volume of the plant from the change in water level. Also, there can be manual errors in the precise measurement of the water's level (the water's surface is a bit concave).

One motivation of our work is to address this problem and propose a solution to automate plant growth analysis in a non-invasive/non-contact and yet accurate manner. Recent progress in imaging techniques greatly aid us in the analysis of an object from various image modalities. With the advent of laser scanning technologies, we can measure dense depth (range) maps of an object's surface in point cloud format. Using multi-view geometry, we can accurately build a 3D point cloud model of an object without touching or damaging it. We exploit state-of-the-art algorithms to process this data to solve plant *phenotyping* problems.

1.2 Thesis Contributions

This thesis focuses on 4 related problems in Computer Vision based 3D plant growth measurement. We briefly discuss the contributions in the following subsections.

1.2.1 Machine Vision System

We propose a Machine Vision system that is capable of analysing plant growth throughout its life cycle in a fully automated manner. We used a robotic arm to build a gantry robot system having 7 degrees of freedom. The robot arm is highly flexible and can be moved anywhere around the plant. A laser scanner is attached as the robot arm's payload. The laser scanner can measure accurate dense depth maps of the plant's surface in point cloud format. The robot arm and gantry system is placed inside a controllable growth chamber, in which the temperature, humidity, lighting, wind, etc. can be controlled as needed. The idea is to scan a plant from multiple viewpoints at many times throughput a day. The viewpoints are chosen so that adjacent scans overlap. This process is continued throughout the life cycle of the plant (or until a desired time period has been reached). For every set of overlapping scans at a specific time, we reconstruct the 3D model of the plant from the point cloud data by registering the data sets into a single 3D triangulated polygonal mesh. The volume and surface of this mesh closely approximates surface and volume of the original plant. We show that the volume and area of a plant's meshes over time is often a good growth metric. Our growth results exhibit the well known diurnal growth pattern of the plants. We have automated the whole system in such a way, in theory, that a naive user can start the growth measurement system by a single mouse click and obtain fully processed growth results at the end of the life cycle of the plant. In practice, there can be problems preventing this, including, but not limited to, power, computer and networking problems.

1.2.2 Multiview Alignment

As a part of the automation process, we need to reconstruct multiple views of a plant in an accurate manner. This is a very crucial step of the growth analysis process. We have shown a quantitative comparison of state-of-the-art registration algorithms and demonstrated their limitations on aligning multiple scans of plant data. We extended the state-of-the-art registration algorithm for aligning large data sets from sequential scans. Quantitative analysis show that our alignment outperforms existing methods.

However, there are still some limitations of the proposed method to align multiple views. In order to obtain good results, different scans need to be roughly aligned. If there is a large difference in the estimated and actual rotation angle between adjacent scans, the alignment results rapidly deteriorate. One of our contributions in this thesis is to accurately estimate this rotation angle by a robust registration algorithm using junction points in the plant data. This, the need for prior knowledge of rotations between scans is eliminated.

1.2.3 Rough Initial Alignment

One idea to perform the rough alignment of the scans is to find some common points in two scans, and estimate the transformation parameters between these points. In the Computer Vision literature, these points are often called "feature points" or points of interest. Usually feature points are selected as "meaningful" locations in the image/point cloud. For example, corner points, edgels, intersection points, points of sharp discontinuity in local structure, etc. can be treated as meaningful locations in the image/point cloud. In general, the feature points are implemented using the idea of “descriptors”. For every point in the image, a local neighbourhood is selected, and the local geometry is encoded as a multi-dimensional vector or histogram. This is called a “descriptor”. In other words, a descriptor contains the local semantics of an image/point cloud. After computing the descriptor for every point in both images, these are matched according to their numerical values. A threshold is typically used to control the number of matches. However, this idea does not work well on plant data because of the complex morphology of the plant structures. Also, it is very hard to find features which are repeatable from different viewpoints.

We present a new approach of selecting feature points in plant point cloud data. Instead of using traditional descriptors for features, we propose the idea of using branch junctions as feature points. Junctions are the points where two or more branches meet. These junctions are strong feature points in plant data, and are repeatable from different viewpoints, because the junctions do not alter even if there are deformations in the data.

However, due to occlusion and clutter, detection of junction points can be erroneous in presence of leaves. There can be many false positives, and even the number of outliers can be significantly larger than the number of true junction points. We propose a simple but effective solution to this problem by using a density based clustering technique to filter out the false positives.

After the detection of feature points, we need to match the features between the image pairs. This is known as the *correspondence estimation* problem. Matching of feature points is a fundamental problem in stereo and motion analysis. Usually, features are matched by their descriptor values and spatial information. For each descriptor in one image, a local neighbourhood around the same position in the other image is selected, and the compatibility of the descriptors are matched for correspondence estimation. However, if the image pairs are not roughly aligned, the local neighbourhood search based feature matching fails to work. Sometimes spatial information of the features are also used to handle these cases. Despite much research over the last several decades, correspondence estimation is still an active area of research, and the state-of-the-art algorithms often fail when the assumptions are not valid, especially in real life domains. For plant data, the problem is even more challenging. In our case, the junction features are raw coordinates of the points without any colour information.

We exploit the advantages of graph based structures and formulate the problem as a subgraph matching optimization technique by considering junctions as graph nodes and their geodesic distances as edges. Also, the graph structure automatically makes the solution rotation invariant. By assigning a cost function between two vertices, we demonstrate that the optimal matching of graph nodes produce better results than existing methods.

1.2.4 Partially Occluded Leaf Classification

We address the problem of classifying partially occluded leaves by their contours as our final contribution. This problem arose from the work of MSc student Pu (Paulina) Yang [156]. She manually segmented and tracked individual leaves in her plant range data. Often these leaves became occluded by other plant matter and tracking had to stop. [In the end, full non-occluded leaves at later times were backwards tracked to avoid the occlusion problem completely.] Matching occluded leaves to full leaves is an NP-Hard problem and, as of yet, does not have a good approximate solution. This problem motivates our interest in 2D partial contour matching. 3D partial contour matching is left to future work.

Recognition of plant leaves has been an active area of research in recent years. Plant species identification from leaf images is now possible from mobile phones (Leafsnap app [77]). Leaves play a vital role in plant *phenotyping* and species identification. Crop monitoring systems can analyze the condition of a plant by the leaf texture [154]. Also, tracking and pruning of leaves is an essential part of these systems [5]. However, tracking of leaves over time can be challenging when the plant grows and leaves occlude each other.

We assume that the contour information and occlusion boundary of an occluded leaf are available. The problem is to find the plant species of the occluded leaf from a database of full leaf contours. Basically this is the partial contour matching problem, which is known to be NP-Hard [135]. Large intra class variations of plant leaves make the problem even more challenging. Recognition of occluded leaves has not been studied before. To the best of our knowledge, we are the first to report occluded leaf recognition where the occlusion can be as high as 50%. We demonstrate that the proposed method on partially occluded leaf classification outperforms state-of-the-art algorithms on 3 publicly available leaf databases.

1.3 Relevant Publications

1. A. Chaudhury and J.L. Barron, *“Partially Occluded Leaf Recognition via Subgraph Matching and Energy Optimization”* (in review)
2. A. Chaudhury, C. Ward, A. Talasaz, A.G. Ivanov, M. Brophy, B. Grodzinski, R.V. Patel, N.P.A. Hüner and J.L. Barron, *“Machine Vision System for 3D plant phenotyping”* (in review).
3. A. Chaudhury, M. Brophy, and J.L. Barron, *“Junction Based Correspondence Estimation of Plant Point Cloud Data using Subgraph Matching”*, IEEE Geoscience and Remote Sensing Letters, Vol. 13, No. 8, pp. 1119-1123, August 2016.
4. A. Chaudhury, C. Ward, A. Talasaz, A.G. Ivanov, N.P.A. Hüner, B. Grodzinski, R.V. Patel, and J.L. Barron, *“Computer Vision Based Autonomous Robotic System for 3D Plant Growth Measurement”*, In Proceedings of the 12th Conference on Computer and Robot Vision (CRV) 2015.
5. M. Brophy, A. Chaudhury, S. Beauchemin, and J.L. Barron, *“A method for global non-rigid registration of multiple thin structures”*, In Proceedings of the 12th Conference on Computer and Robot Vision (CRV) 2015.

1.4 Thesis Outline

The rest of the thesis can be summarized as follows:

- **Chapter 2** reviews the background of different techniques that are used in rest of the chapters. More specifically, the literature survey is focused on the following areas: imaging techniques in plant *phenotyping*, point cloud registration algorithms, correspondence estimation techniques and contour based shape matching algorithms.

- **Chapter 3** explains Gaussian Mixture Models, describes a quantitative comparison of state-of-the-art point cloud registration algorithms. Finally the multi-view alignment algorithm is discussed.

  This work resulted in a publication at the *Computer and Robot Vision* (CRV) conference in 2015 [18].

- **Chapter 4** outlines the whole system to automate plant growth analysis. Details of the system components (gantry robot, laser scanner, Biochamber, etc.) and their integration is explained in detail. We briefly discuss the multi-view alignment of plant point cloud data, followed by triangulation, mesh area and volume computation. Finally the growth curves are analyzed.

  This work was published the *Computer and Robot Vision* (CRV) conference also in 2015 [31] and subsequently expanded and submitted to a journal [30].

- **Chapter 5** presents our idea of using junctions as feature points in plant point cloud data and solves the correspondence problem by formulating a subgraph matching algorithm. We also show that the idea can be extended to match occluded leafy plants by using a simple but effective density based clustering technique.

  This work was published in *IEEE Geoscience and Remote Sensing Letters* (GRSL) [29].

- **Chapter 6** describes the algorithm for classification of partially occluded leaves. We show that it is possible to match a leaf contour having at most 50% occlusion in its contour.

  This work is under review of a journal [28].

- **Chapter 7** summarizes the whole thesis and possible future research directions are pointed out.

Chapter 2

Literature Survey

This thesis is composed of different types of problems and techniques related to Computer Vision based plant growth analysis. We discuss the related literature in various sections below.

2.1 Computer Vision Approaches for Plant *phenotyping*

In last decade there has been tremendous progress in automated plant *phenotyping* and plant imaging technologies [133]. Accurate *phenotyping* of plants is crucial in analyzing different properties of plants in different environmental conditions. Traditionally,y biologists have used naïve (manual) methods for plant *phenotyping*. This led to low throughput and sometimes questionable accuracy. Accuracy and throughput are major factors in mass scale analysis. To build a real time plant *phenotyping* system with high accuracy, the system needs to be fully automated. A truly automated system should allow a naïve user to operate the phenotyping process and obtain the growth analysis results as a ready-made end product. As many of the current *phenotyping* technologies focus on software development for processing the data [63, 75], automated collection of data is also becoming state-of-the-art [121, 136] over tedious manual techniques [111] . However, most of the automated systems have their limitations. Either they are dependent on particular types of plant(s), or on their size and geometrical structures. In an attempt to generalize the *phenotyping* process, Computer Vision based system design is becoming very useful in studying plant growth, and a body of work has been reported in that area [82, 54, 51].

Among several components of *phenotyping*, growth analysis is extremely important [53]. A plant's growth is highly affected by the environmental conditions. Accurate measurement of a plant's growth can reveal much information which can be useful for accelerating crop production. In recent years, different aspects of plant *phenotyping* and growth measurement have appeared in the literature.

Detection and tracking of plant organs (e.g. flowers, buds, stems, leaves, fruit) have gained the interest of many researchers. A Machine Vision system for fruit harvesting was proposed by Jimenez *et al.* [70]. They used an infrared laser scanner to collect data and used computer vision algorithms to detect fruit on a plant using their colour and morphological properties. This type of automated system is in great demand for fruit harvesting and agricultural engineering. Automated classification of plant organs can be useful for tracking a specific area of the plant over time. Chattopadhyay *et al.* [2] presented a 3D reconstruction of apple trees. They used a semi-circle fitting technique to model the main trunk and primary branches. This can be helpful in automating the pruning process as a part of intelligent agricultural robotic application. Paulus *et al.* [109, 110] showed a feature based histogram analysis method to classify different organs in wheat, grapevine and barley plants. A segmentation algorithm to monitor grapevine growth was shown in [74]. A similar type of work on plant organ segmentation by unsupervised clustering was proposed in [145]. Paproki *et al.* [108] showed a 3D approach to measure plant growth in the vegetative stage. They generated 3D point cloud from 2D images. A multi-camera set-up is used and a plant's 3D model is reconstructed via projection matrices in [59]. They demonstrated automatic segmentation of leaves and stems to compute geometric properties such as area, length, etc. and validated the result by comparing with ground truth data destructively by hand. Dellen *et al.* [38] demonstrated a system to analyze leaf growth of tobacco plants by tracking the leaves over time.

Other recent work focused on detecting specific patterns in a plant's leaves to determine the particular condition of the plant. Analysis is performed by tracking leaves and detecting colour properties of the leaves. However, segmentation of plant leaves in different imaging conditions is a challenging task [122]. Kelly *et al.* [73] showed an active contour based model to detect lesions in *Zea mays*. This crop is widely used and detecting the lesions can be helpful to detect disease in the early growth stage. Xu *et al.* [154] presented an approach to detect nitrogen and potassium deficient tomatoes from the colour images of the leaves. Recent work on tracking leaves of rosette plants can be found in the work of Aksoy *et al.* [3]. This work can be useful for measuring growth rate of a particular leaf.

Building a 3D model of a plant from multi-views is a challenging task. The complex geometry of plants make the problem of 3D surface reconstruction difficult. Pound *et al.* [113, 112] proposed a level set based approach to reconstruct the 3D model of a plant from multiple views. Their reported results are promising in that they show that their 3D model closely resembles the original plant structure. A feature matching based approach is used to build the 3D model of a plant [119] using a structure from motion algorithm. However, the method is highly dependent on local features. The idea does not work for the cases where the plant surface is featureless. Note that range data does not require texture or features on the plant. Santos *et al.* [120] applied Structure from Motion (SfM) to build 3D model of the plant and accurately computed plant height and leaf length. Simek *et al.* [127] presented a method to estimate the thin branch structures of a plant from monocular images. They modelled the spatial smoothness of the branches by Gaussian Process (GP), that allows attachment of the branches.

Rhythmic patterns of a plant's growth are well studied in the biological literature [55, 128, 56]. A system capable of detecting diurnal growth patterns can be reliably used to monitor the growth pattern of different species in different conditions. Imaging based techniques are becoming more popular in such analysis [132]. A vision based system to study the circadian rhythm of plants was presented in [104]. The automated system captures the diurnal growth pattern using 2D imaging techniques. A laser scanning based 3D approach was reported in [42] which shows the diurnal pattern of leaf angle in different lighting conditions. Tracking and growth analysis of a seedling by imaging technique was studied in [13]. Barron and Liptay [9, 10] and Liptay *et al.* [86] used a front and side view of a young corn seedling imaged by a near-infrared camera to obtain 3D growth for up to 3 days. The growth was shown to be well correlated by root temperature (Pearson coefficient 0.94 [9]).

Godin and Ferraro [58] presented a structural analysis of tree structures which can be useful in plant growth analysis. Augustin *et al.* [7] modelled a mature *Arabidopsis* plant for phenotype analysis. They demonstrated extraction of accurate geometric structures of the plant from 2D images which can be useful for *phenotyping* studies of different parts of the plant. Tracking the event of budding and bifurcation in a plant's life cycle can be very useful for growth analysis. Li *et al.* [83] performed a 4D analysis to robustly segment plant parts to localize buddings and bifurcations accurately, using a backward-forward analysis.

2.2 MSc Theses from Barron's Lab

This thesis partly used some of the results found in 3 previous MSc theses on Computer Vision based plant growth analysis by Chen Zhao [163], Pu (Paulina) Yang [156] and Zereen Akter [4]. Mark Brophy (PhD 2015) is described elsewhere.

Zhao [163] presented a *proof of concept* technique that the 3D area and 3D volume could be used as measures of Arabidopsis plant growth. Six ping-pong ball mounted on sticks were placed about a growing plant. Figure 2.1 shows the scanning setup used. The plant was on a circular table and the range sensor was fixed on a tripod to give a good view. Twelve scans of the plant at 30° rotation increments were recovered and the scans were merged using the Geomagic Studio 12 software. Geomagic Studio required the ping-pong balls to do the registration. Since each scan view of the plant only has semi-sphere data (the visible half of the ping pong balls), each scan was manually manipulated, with the semi-spheres being cut (edited) from the scans and synthetic 3D spheres inserted in their stead at these locations. This manual intervention was tedious, time consuming and error prone. The scanner used has a laser beam at 660

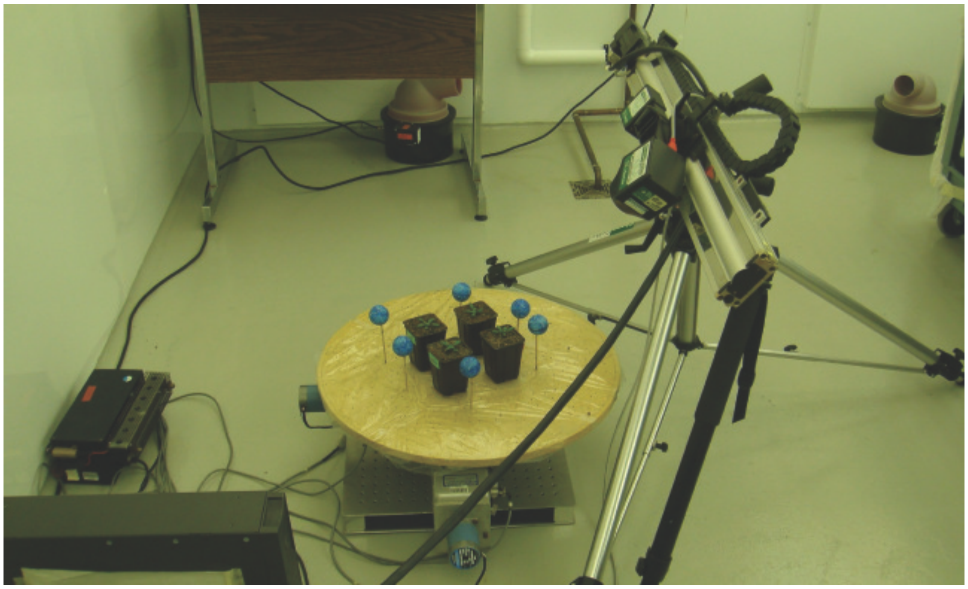


Figure 2.1: The experimental setup used for scanning the Arabidopsis plants.

nm in the visible spectrum and was intended to work under normal lighting conditions. One problem was that the scanner worked poorly in well lit environments and, as a result, most of the scanning was done in the night time. The effect of a 660 nm scam beam on the plant's growth is unknown (undocumented) but 660 nm is a wavelength known to be involved in plant growth. Using a 660 nm laser beam did make the measurement system a bit invasive. Zhao also performed manual segmentation of plant leaves and tracked then over time, plotting time versus leaf size as an indication of leaf growth. Occlusion problems were avoided by manually choosing leaves that were never occluded by other plant matter. The volumes of stems were also measured with some success. Generally, Arabidopsis stems were too thin to obtain highly accurate measurements. Lastly, plant canopy areas were measured with the Arabidopsis in its vegetative state. There was significant occlusion in the canopy leaves, which was ignored. Zhao show that range data could be used to recover quantitative plant growth data but no long growth sequences were produced (the longest was for 9 days).

Yang [156] focused on solutions for the problems encountered in Zhao's thesis [163]. For example, there were problems with manually editing the leaves in the polygonal mesh, especially when the leaves were near soil (remember the range data does not have any texture information). Yang drew a red line using Geomagic Studio to outline the contour of a leaf (and, hence its region) and then just eliminated everything outside the leaf's region. To handle the problem of the laser scanner being sensitive to highly lit scenes, Yang only measured growth from 8pm to 8am. Yang conducted her experiments in the Biotron [156]. The growth chamber used had uncontrollable wind (the only way to turn off the wind was to completely power-down the growth chamber). The plant still jittered a bit even though the plant and scanner setup was enclosed in plastic sheet structure. Yang acquired the Arabidopsis plant dataset used by Brophy *et al.* [18] to illustrate their registration algorithm. In that work, Brophy *et al.* showed the resilience of the registration algorithm to smaller jittering datasets. There were a lot of uncontrollable physical problems associated with using the Biotron, including loss of temperature control during 2 summer thunderstorms. The plans died in the resulting 50°C temperatures. The plant also often grew out of the field of view of the scanner, resulting in incomplete data. Sometimes, there were networking and computer problems as well. The full automation of plant growth in this thesis is meant to resolve all these and other problems.

Nevertheless, in spite of all her problems, Yang was able to acquire longer range datasets over the life cycle an Arabidopsis plant. She still had to contend with ping-pong ball placement and the laser beam wavelength being in the visible spectrum. The use of a near-infrared laser beam was proposed. At that time, such a scanner was not yet on the market.

Akter's thesis [4] is an important contribution on the rectification of the ping-pong ball placement experienced by the two previous theses [163, 156]. The main contribution of her thesis is to remove the problems caused by having to manually replace semi-sphere data with full sphere data. Her solution was to generate synthetic full sphere data and automatically detect semi-sphere data via a least squares fitting algorithm in each range image and replaces this data with the full sphere data. Thus, the balls only had to be manually placed once at the start of an experiment. This process was very useful in automating and speeding up the 3D measurement process.

It was eventually decided to give up on the Geomagic Studio software (the placement of ping-pong balls was problematic) and write the registration code from scratch in MatLab (with no requirement for ball placement at all). A near-infrared ShapeGrabber scanner also became available at that time and was used for this thesis' results. It allow scanning in the day and night.

2.3 Multi-View Alignment

Registration is a fundamental task in 3D reconstruction, where the aim is to find a coordinate transformation to align source and target views (referred to as *pairwise registration*). The basic Iterative Closest Point (ICP) [14] algorithm finds a rigid transformation that registers two views using local optimization. However, ICP is prone to getting trapped to local minima if the initial guess is not good. Several variants and applications of the classical ICP are reported in the literature ([142, 161, 94, 114, 116, 68, 17]).

Turk and Levoy [142] exploited ICP algorithm to align surface meshes from multiple scans. First they align the adjacent meshes by considering the topology of the object. Then all the surfaces are aligned together by computing local weighted average of the individual surfaces. Masuda and Yokoya [94] presented a registration algorithm by combining ICP and Least Median of Squares (LMS) estimator to determine the rigid motion parameters from two range images. The method is shown to be robust on noise and occlusion. Pulli [114] to some extent handled the problem of getting stuck to local minima of ICP algorithm. The method first aligns pairwise scans and then uses this as a constraint to globally register the whole data set. In other words, the multi-view registration algorithm "diffuses" the pairwise registration errors. Rusinkiewicz *et al.* [116] proposed an advancement of ICP algorithm by uniformly sampling the surface normals. However, the algorithm needs a good initial guess to find a reasonable solution. Huber and Hebert [68] presented a registration algorithm to align multiple rigid datasets without any need of manual intervention. The method is based on two phases: local and global. The local match performs pairwise matching and forms a graph by considering local consistencies of the views. Then the global phase filters out bad matches by enforcing global consistency criteria, and transforms the two views in a common coordinate system. Bouaziz *et al.* [17] improved the traditional ICP algorithm by considering large number of outliers. They introduced a sparsity regularization term in the ICP energy function to handle the outliers.

Fitzgibbon [52] modified ICP by deriving an error function between the model and target data which is minimized by the Levenberg-Marquardt algorithm (LMICP). The energy is formulated in terms of the L_2 distance of the closest point in the data to each point in the model, but he instead computed distances to each point in a discrete volume, thus allowing to compute the spatial derivatives needed for energy minimization. This approach makes the registration process more general than ICP. The method requires rigid data points and works well on the standard Stanford Bunny dataset (which satisfies this constraint).

Thin Plate Spline (TPS) [34] is one of the most successful and widely used algorithm for point set registration. This is is an advancement of the cubic B-spline with a regularization factor, controlled by regularization parameter λ. The physical analogy is like bending a thin sheet of metal to fit a surface consisting of the points. The regularization controls the amount of bending of the sheet. However, this involves a proper choice of λ. Ignoring the regularization factor ($\lambda = 0$), TPS results in exact interpolation.

Statistical approach to point set registration is shown to be successful to some extent. Tsin and Kanade [141] introduced a Kernel correlation (KC) method to measure the affinity of the point sets. The point set registration problem is defined as maximum kernel correlation configuration of the the two point sets. The kernel function is usually chosen as symmetric and non-negative. The technique is shown to be less sensitive to noise.

In recent years, The Point Cloud Library (PCL) [84] has introduced some libraries for point cloud alignment and is based on Rusu *et. al.*'s method [118, 117] for Fast Point Feature Histogram (FPFH). Each point in a dataset is assigned multiple informative labels as features and these are then used to establish correspondence, thus resulting in a good initial alignment for registration. The geometrical information (16 dimensional) for the neighbourhood of each point is extracted and stored in histograms. The method has also been shown to be robust to pose invariance. Unfortunately, this method still does not overcome the difficulties associated with iterative algorithms like ICP.

Gaussian Mixture Model (GMM) based registration has been extremely successful in wide class of problems and is a state-of-the-art technique for point set registration. Chui and Rangarajan [33] formulated the point matching problem as maximum likelihood estimation of mixture models. The Expectation Maximization (EM) algorithm is used to solve the Maximum a-priori Estimation (MAP) problem. Zhang *et al.* [160] introduced a variational method to register non-rigid image pairs. The dissimilarity measure of two images is modelled as Gaussian Mixture distributions. In a similar approach, Somayajula *et al.* [131] also exploited Gaussian Mixture Model based approach to register point sets.

Jian *et al.* [69] and Myronenko *et al.* (CPD) [101]'s GMM based registration algorithms have been the most successful in recent years. The basic idea of these algorithms is to represent the two point sets as Gaussian Mixtures, and minimize the distance between them. Usually TPS is used to model the transformation. CPD extends the idea by representing the two point sets as GMM centroids, and are forced to move "coherently" as a group. This helps to preserve the topological structure of the point sets. Recently Ma *et al.* [91] extended the GMM based approach by incorporating shape context [12] prior. Instead of assigning uniform membership probabilities to the GMM components, they used the prior probability to achieve better results.

2.4 Correspondence Estimation

Although *Pairwise Registration* and *Correspondence Estimation* basically refer to the same task of finding matching points in two point sets, correspondence estimation is often used as a part of the registration process. A typical work-flow of point set registration involves feature point detection, correspondence estimation, followed by outlier rejection from the estimated correspondences, and finally transformation estimation of the whole point set. Pairwise registration involves transformation of all the points in one set to the other (either locally, or globally, or both), and in many cases good initial guess is very important to get good results.

Registration algorithms can be broadly classified into two groups. The first type is a global registration method like CPD [101] or GMM [69]. The second type is based on the computation of local feature descriptors and matching ([90, 72, 96]). Global registration methods often fail because of local deformation between two views. On the other hand, the basic idea of feature point extraction is to find changes in local structure (intensity or local geometry). We consider only raw plant point cloud data, without any intensity or colour information. Applying a standard 3D key-point detector algorithm relies on local descriptors ([134, 96]) for feature extraction and matching is performed based on descriptor similarity. This idea works well for indoor/outdoor scene point cloud with color information and models with well defined surface and curvature. However, applying this idea to plant like structures, which contain many similar geometrical features, usually results in ambiguous matches in a correspondence calculation. Matching correspondences in forest scenes are studied in some recent articles ([64], [16]). However, the problem of finding efficient correspondences still remain unsolved.

Bucksch *et al.* [20] addressed this problem and proposed a localized registration algorithm. The method is highly dependent on the skeletonization, which might not work in the presence of noise and plant leaves. Also, their method is based on the assumption that an initial guess is available. They assume rough manual alignment of the junction feature points in two trees. For large rotation angle differences, the algorithm may fail to register two views correctly. Recently Zhou *et al.*[169] presented an approach extending [20]. However, their method is still dependent on skeletonization, which still involves manual intervention and doesn't handle occlusion and local deformation. Also, the idea of cylinder fitting does not work for thin stems.

2.5 Contour based Shape Matching

Shape matching is a long standing problem in Computer Vision and Pattern Recognition. Shapes can appear in different ways, such as discrete point clouds, triangular surfaces, silhouettes. Recognition of an object by its contour has been shown to be very feasible on large varieties of shapes [80]. Since the contour of an object is not affected by colour, illumination and other textural properties, contour based representation can be very effective. The recognition problem can be solved by either correspondence matching between two point sets, or as a part matching problem by "learning" object parts from database and then recognizing a object from the learned categories. In the former case, a typical approach is to find good feature points (or descriptors encoding the local and global geometrical structures of the shape) on the contour and then to match the feature points between two images to find the correspondence ([23], [149]). The quality of a match is typically determined by the "score" of the matching. Another common approach to find the point correspondence by formulating the point matching as an assignment problem and optimizing it. Shape Context (SC) [12] is a well known example of such a technique, which builds a log polar histogram for each contour point and finds the best match by formulating the matching as a bipartite graph matching problem and minimizing the cost function. The method is state of the art and works well on large varieties of data sets. Scott *et al.* [123] presented a contour matching technique that improved on the typical assignment methodology by considering the ordering of the discrete contour points. Ling *et al.* [85] extended the idea of shape context by replacing the Euclidean distance by a inner distance measure, called Inner Distance Shape Context (IDSC). They reported much higher recognition rates than SC by using a Dynamic Programming approach. Later the idea of shape context was used in various applications. Ma *et al.* [92] proposed a technique to match partially occluded object contours. They used a shape context based descriptor and formulated partial graph matching as a weighted graph matching problem. Also, there has been attempts to handle local deformation in the shape contour. Xu *et al.* [153] defined a deformable potential at each point on the contour, which can handle deformations. Kontschieder *et al.* [76] formulated a k-nearest neighbour graph as shape representation and performed an unsupervised clustering method to retrieve the closest match of a query shape. Hu *et al.* [66] proposed a shape descriptor based on angular pattern of contour points. The descriptor is invariant to scale and rotation due to the relative orientation of the contour points. Shen *et al.* [125] combined the contour and skeleton of the shape using the correspondences between them. A Bag of Features idea is exploited to form the feature vector, and shapes are classified by a linear Support Vector Machine (SVM) classifier.

Another approach for shape recognition involves part learning. The idea is to build a "dictionary" of object parts and then find the best match from the learned object parts. Opelt *et al.* [107] encoded contour fragments into a weak classifier and formulated a "boundary fragment model" to classify an object. They perform boosting, which learns from a number of weak classifiers and "boosts" these classifiers to form a strong classifier. One advantage of the algorithm is that, it needs a limited number of training samples. A similar approach was used by Shotton *et al.* [126]. They built a "codebook" of local fragments of contours and learned via a boosting algorithm. Ferrari *et al.* [49] demonstrated a rule based approach to recognize objects by their parts. Different parts of an object contour are connected to form a contour segment network and matching is performed by finding efficient paths through the network. Bai *et al.* [8] presented a descriptor learning approach by exploiting "bag-of-words" idea. A spatial pyramid based approach is adopted to learn local and global features at different levels. Recently, Bicego *et al.* [15] formulated the contour matching as biological sequence matching problem and demonstrated better results than the state-of-the-art on 5 shape databases.

Apart from matching contours, either via pairwise point matching or part learning, other approaches have been shown to be successful to some extent. In many cases, an object can be efficiently represented by its meaningful parts. Although its hard to define "meaningful" quantitatively, convexity and curvature information can be useful since these are strong cues for human vision. Discrete Contour Evolution (DCE) [79] is a popular technique to decompose an object into meaningful parts based on the convexity of the shape contour. Yang *et al.* [155] extended the idea of DCE and performed a dynamic programming based matching to recognize shapes, which is invariant to scale, rotation and deformation. Another way to recognize meaningful features is to find the points where the curvature undergoes change in sign. Mokhtarian *et al.*'s [98] Curvature Scale Space (CSS) feature matching technique has been quite successful for closed contour matching [1]. They detect feature points at different scales based on curvature sign change and match those features to find the shape similarity. Wang *et al.* [150] used this idea by representing the shape contour as B-spline curve. The idea of CSS was later extended for open curves [167]. Topological skeleton (medial axis) based approaches (*shock graph* [124]) are dependent on the skeleton of a curve being well represented by its skeleton. This works well for shapes with large variations. However, sometimes small local changes in the shape cause large change in the skeleton. For leaf shape recognition, with large local shape variations, this type of idea would not work.

Curve moments are shown to be effective for many cases of contour matching [164]. Representing the contour by a B-spline and matching the spline curves by curve moments is reported to work well for occlusion handling and missing data ([35], [67]). However, the method can't be applied to real time applications because the matching needs to be performed for all possible combinations of discrete curve sections in the database. Also, curve moments are very unstable, as small changes in shape can affect significant changes in the curve moment.

2.5.1 Contour Based Leaf Matching

Automatic identification of plant species is an active area of research these days. There are many ways to identify a plant, for example, by the flowers, fruit, leaves, or other organs of the plant. Although there has been some work on flower classification [105] and large scale plant identification from learned categories [46], classifying leaves in order to determine the species is the most common approach. There is a large body of literature ([97, 103, 65, 77, 24, 146, 27, 26, 100, 25, 87, 148, 162, 147, 71, 166]) over the last two decades on leaf recognition for plant species identification. Leaf classification was first reported by Mokhtarian *et al.* [97]. They studied recognition of leaf species having self intersections by Curvature Scale Space (CSS) matching. However, they focused on the particular case of self intersecting leaf images and did not report recognition for normal leaves. Nam *et al.* [103] performed a nearest neighbour based feature point matching technique to recognize leaves. Recently, Multiscale Distance Matrix (MDM) [65] has been a popular leaf recognition technique. The method is simple and fast. Basically, pairwise distances among all discrete contour points are represented in matrix form and the coarse to fine level details of the shape is encoded by a compact representation of the distance matrix. This matrix is used to perform recognition by reducing the dimensionality, similar to a principal component analysis. The method achieves good performance on two leaf databases. However, they did not study occlusion.

In recent years, one of the most successful and practical leaf recognition system is Leafsnap [77]. The mobile app system scans a leaf image (which needs to be on a uniform background), segments the image, extracts contour and matches with thousands of leaves in the database. The underlying method computes the Histogram of Curvature over Scale (HoCS) and matches via a nearest neighbour technique. They did not consider occlusion. Curvature Scale Space [24] and convexity based methods [146] are also known to have some success in leaf recognition. Cerutti *et al.*'s [27] method focuses on the segmentation of a leaf from its complex background and ultimately performs matching based on a CSS formulation. However, the method is dependent on sharp features (like leaf teeth and tip) on the leaf contour. Identification of compound leaves is more challenging than classifying individual normal leaves [26].

Some techniques exploit salient points on the leaf contour ([100], [25], [166], [71]), and matching is performed based on those features. Nevertheless, this idea fails when the leaf does not have enough distinguishable features. Hierarchical representation of the shape contour is an efficient way to capture the geometry at different scales, which can be exploited using a Shape Tree (ST) [48]. Another hierarchical approach for partitioning the contour into different lengths was presented by Wang *et al.* [148]. They captured the geometrical structure via the distribution of points around a straight line and then built a shape signature based on Fourier transform coefficients. The method achieves good performance on a public database of Swedish leaves including leaves that are collected from 100 tree species. Some recent work ([87], [147]) developed leaf recognition systems for mobile applications. Recently Zhao *et al.* [162] demonstrated a computationally fast technique for leaf recognition. However, it needs prior training for the classifier. Furthermore, occlusion is not handled by their method.

Chapter 3

Multi-View Alignment

This chapter presents a quantitative comparison of existing pairwise registration algorithms and extends the state-of-the-art algorithm to align multiple scans of plant data. The alignment algorithm is from PhD thesis of Mark Brophy [19]. Our contribution is mainly the quantitative comparison of the state-of-the-art algorithms on plant data. This work resulted in the following conference publication:

- M. Brophy, A. Chaudhury, S. Beauchemin, and J.L. Barron, *"A method for global non-rigid registration of multiple thin structures"*, In Proceedings of the 12th Conference on Computer and Robot Vision (CRV) 2015.

3.1 Introduction

Building a 3D model is extremely useful for many practical applications where inferring measurements about the shape or size of an object is a requirement. In biological science, building 3D models help in analysis of different properties of the object. For example, building a 3D model of a plant can be useful in in plant *phenotyping* ([110, 145, 119, 113, 120, 111]), or building accurate 3D model of molecular structures (e.g. protein, bacteria) can revolutionize the analysis in molecular biology [115].

With the rapid acceleration of hardware capability in terms of CPU power, storage and 3D scanners, we can now capture huge amounts of data and the fidelity of the resultant models is constrained by the quality of merging/reconstruction algorithms. Due to recent advancements of robotic technologies and low cost laser scanners, building real time automated systems is becoming possible for plant science and agricultural applications, where a major task is to build a 3D model of the plant to analyse different biological properties (like growth, etc.). However, the complex recursive structure of a plant makes the problem of aligning multiple views extremely hard, unlike building a 3D model of a rigid object. Also in medical robotics, automatic analysis of medical images is a crucial step. A common application is to build a 3D model of thin artery data from multiple tomographic or CT images. This involves pairwise registration and alignment of different scans. However, registration of vascular data is a challenging problem and despite several years of research ([129], [39]), it still remains an open problem in medical imaging community. We address the problem of aligning multiple views of thin recursive structures like plant or artery data.

3.2 Gaussian Mixture Model based registration

Registration of 3D point cloud data has been studied extensively in the literature. We refer to Chapter 2 for the literature background. Recently, Gaussian Mixture Models (GMM) have been very successful for the registration of non-rigid point sets.

Let us consider that the model point set is denoted by $\mathcal{M} = (x_1, x_2, ..., x_M)^T$, and the observed data points are represented by $\mathcal{S} = (y_1, y_2, ..., y_N)^T$. The model points undergo a non-rigid transformation $\mathcal{T}$, and our goal is to estimate $\mathcal{T}$ so that the two point sets become aligned. The GMM probability density function can be written as,

$$p(y_n) = \sum_{i=1}^{M+1} P(z_n = i) p(y_n | z_n = i), \tag{3.1}$$

where z_n are latent variables that assign an observed data point y_n to a GMM centroid. Usually all the GMM components are modelled as having equal covariances σ^2, and the outlier distribution is considered as uniform $1/a$. The unknown parameter $\omega \in [0, 1]$ is the percentage of the outliers. The membership probabilities π_{mn} are assumed to be equal for all GMM components. Denoting the set of unknown parameters $\boldsymbol{\theta} = \{\mathcal{T}, \sigma^2, \omega\}$, the mixture model can be written as,

$$p(y_n|\boldsymbol{\theta}) = \omega\frac{1}{a} + (1-\omega)\sum_{i=1}^{M} \frac{\pi_{mn}}{(2\pi\sigma^2)^{D/2}} exp[-\frac{||y_n - \mathcal{T}(x_m)||^2}{2\sigma^2}]. \tag{3.2}$$

The goal is to find the transformation $\mathcal{T}$. Sometimes a prior is used to estimate the transformation parameters. A common form of prior [158] is,

$$P(\mathcal{T}) \propto exp[-\frac{\lambda}{2}\phi(\mathcal{T})],$$

where $\phi(\mathcal{T})$ is the smoothness factor, and λ is a positive real number. The parameters $\boldsymbol{\theta}$ are estimated using the Bayes' rule. The optimal parameters can be obtained as,

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\mathcal{S}) = \arg\max_{\boldsymbol{\theta}} P(\mathcal{S}|\boldsymbol{\theta})P(\mathcal{T}),$$

which is equivalent to minimizing the negative log-likelihood:

$$\mathcal{L}(\boldsymbol{\theta}|\mathcal{S}) = -\sum_{n=1}^{N} lnP(y_n|\boldsymbol{\theta}) - lnP(\mathcal{T}). \tag{3.3}$$

Jian and Vemuri [69] followed this approach and represented the point set by Gaussian mixtures. They proposed an approach to minimize the discrepancy between two Gaussian mixtures by minimizing the L_2 distance between two mixtures. However, they can't handle large data sets. Their experimental results are for the downsampled Bunny dataset. The algorithm does not work for our plant dataset (over 2 million points). In fact, for complex plant structures, it is impossible to retain the geometry of the model by downsampling the over 2 million points (to about 5000 points, which the algorithm can process).[1] The plant under consideration has lot of branches and has a canopy of leaves at its base, which makes the geometry more complicated.

---

[1] Correspondence between the authors and ourselves revealed that this downsampling could not be avoided.

The Stanford Bunny can be approximated by a few hundred points, but for plant structures using such a small number of points is not possible. Hence, we could not apply this algorithm in our plant data.

The Coherent Point Drift (CPD) registration method was proposed by Myronenko et al. [102, 101]. Their method is based on GMM, where the centroids are moved together. Given two point clouds, $\mathcal{M} = (x_1, x_2, ..., x_M)^T$ and $\mathcal{S} = (y_1, y_2, ..., y_N)^T$, in general for a point x, the GMM probability density function will be $p(x) = \sum_{i=1}^{M+1} P(i)p(x|i)$, where:

$$p(x|i) = \frac{1}{(2\pi\sigma^2)^{D/2}} exp[-\frac{||x - y_i||^2}{2\sigma^2}]. \tag{3.4}$$

They minimize the following negative log-likelihood function to obtain the optimal alignment:

$$E(\boldsymbol{\theta}, \sigma^2) = -\sum_{j=1}^{N} log \sum_{i=1}^{M+1} P(i)p(x_j|i). \tag{3.5}$$

There are many ways to estimate the parameters, such as, gradient descent, Expectation Maximization (EM) algorithm, and variational inference. EM is a standard and widely used technique to optimize the cost function. Basically the E-step (or the *Expectation*) computes the posterior probability, and the M-step (or the *Maximization*) computes the new parameter values from the likelihood function. The aim is to find the parameters $\boldsymbol{\theta}$ and σ^2.

Let's denote the initial and updated probability distributions as P^{old} and P^{new} respectively. The E-step basically computes the "old" parameter values, and then computes the posterior probability distributions $P^{old}(i|x_j)$. In the M-step, the new parameter values are computed by minimizing the log-likelihood function:

$$\mathcal{E} = -\sum_{j=1}^{N} \sum_{i=1}^{M+1} P^{old}(i|x_j) log(P^{new}(i)p^{new}(x_j|i)), \tag{3.6}$$

which can be rewritten as,

$$\mathcal{E}(\boldsymbol{\theta}, \sigma^2) = \frac{1}{2\sigma^2} \sum_{j=1}^{N} \sum_{i=1}^{M+1} P^{old}(i|x_j)||x_j - \mathcal{T}(y_i, \boldsymbol{\theta})||^2 + \frac{N_p D}{2} log\sigma^2, \tag{3.7}$$

where

$$N_p = \sum_{j=0}^{N} \sum_{i=0}^{M} P^{old}(i|x_j) \leq N. \tag{3.8}$$

Now the current parameter values $\boldsymbol{\theta}^{old}$ is used to find the posterior probabilities:

$$P^{old}(i|x_j) = \frac{exp(-\frac{1}{2\sigma^{old^2}}||x_j - \mathcal{T}(y_i, \boldsymbol{\theta}^{old})||^2)}{\sum_{k=1}^{M} exp(-\frac{1}{2\sigma^{old^2}}||x_j - \mathcal{T}(y_k, \boldsymbol{\theta}^{old})||^2) + (2\pi\sigma^2)^{D/2} \frac{\omega}{1-\omega} \frac{M}{N}} \tag{3.9}$$

3.2.1 Quantitative Comparison of State-Of-The-Art

We present a comparative framework to perform a quantitative comparison in registering two point clouds. First, we show our experimental results on the Bunny data (downsampled to 376 points). Then we show our results on real Arabidopsis plant data. The results are shown in Figure 3.1 for the Stanford Bunny dataset and Figure 3.2 for the real Arabidopsis plant.

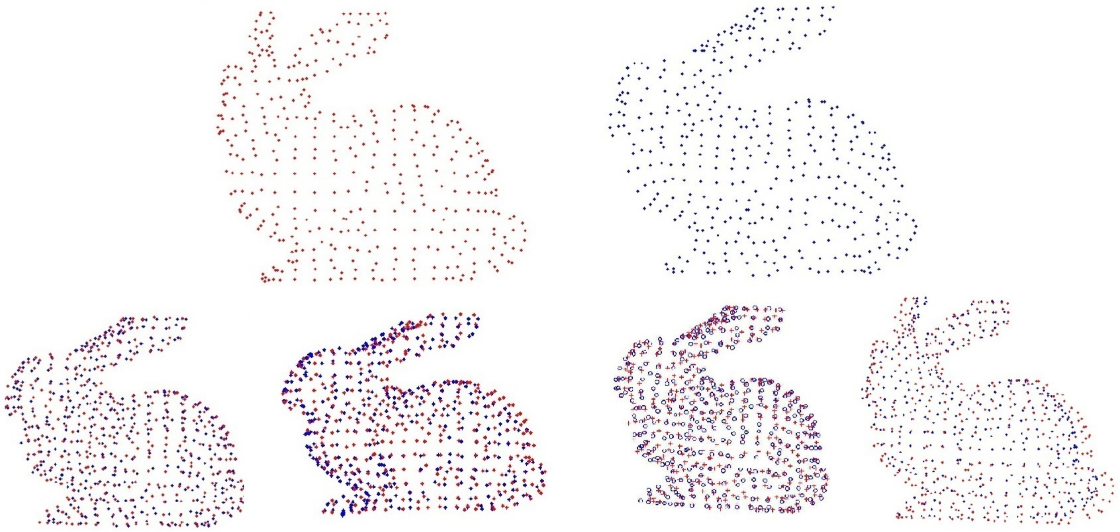


Figure 3.1: Pairwise registration results for two scans of the Stanford Bunny. Top row: original two views, Bottom row: results obtained using Rusu et al. (FPFH) [118], Fitzgibbon (LMICP) [52], Jian and Vemuri (GMMReg) [69] and Myronenko and Song (CPD) [101].

To perform quantitative analysis of results we manually chose a small number of 3D points in the 2 scans that we believe are correct correspondences. For the Bunny and Arabidopsis plant data, we acquired 18 and 44 correspondences respectively. Using that ground truth, we computed the average error rate as the L_2 distances (in mm) between source and target points, where the correspondence was manually measured for the different algorithms. These are listed in Table 3.1.

	Algorithms			
Datasets	FPFH	LMICP	GMMReg	CPD
Bunny	0.149	0.046	0.012	0.014
Arabidopsis Plant	9.661	4.114	N/A	2.312

Table 3.1: Quantitative results for different algorithms and datasets.

We can observe that for Bunny datasets, almost all the algorithms work well. However, for real Arabidopsis data, the error measures are considerably higher. We note that while collecting the ground truth correspondence points, there may have been some errors. Although

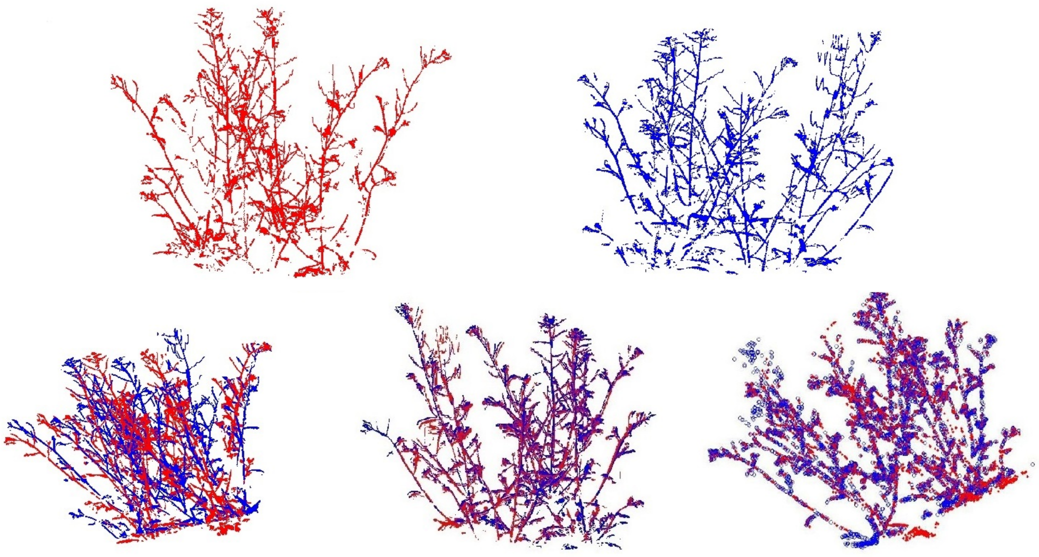


Figure 3.2: Pairwise registration results two scans of the real Arabidopsis plant data. Top row: the original two views. Bottom row: results obtained using Rusu et al. [118], Fitzgibbon [52] and Myronenko and Song [101].

it is difficult to capture the exact error, we expect the error to be in the range 2-5 mm for each point. As we can see, CPD works best among all other methods in processing plant data.

We have concluded from our experimental work, that Myronenko *et. al.*'s Coherent Point Drift (CPD) [102, 101] method works best for aligning two point clouds of the plant data. However, Myronenko *et. al.* didn't consider aligning more than two views.

Applying CPD for alignment of 12 views yielded poor results. One of the reasons was because of the inter-scan movements between different parts of the plant range images (the plant is not rigid). Also, our plant data (Arabidopsis plant) is extremely sparse in the flowering stage (with long stems, a few leaves and flowers), so it is necessary to use a large area of support to construct the GMMs to smooth the solution space, and as a result, the registrations are not sufficiently good to pass a visual inspection.

3.3 Proposed Method

This section comprises of a multiview alignment method from the PhD thesis of Mark Brophy [19]. We discuss the algorithm briefly.

The proposed algorithm is based on CPD which can align many views with minimal error. More specifically, it is a drift-free algorithm for merging non-rigid scans, where *drift* is the build-up of alignment error caused by sequential pairwise registration. Sequential pairwise registration entails the alignment of each scan to its neighbor, followed by the alignment of another neigboring scan to the resultant scan, and so on. The error between any two scans accumulates the error from the previously merged scans used in the current merging. Our data (of the Arabidopsis plant) consists of 12 views of the plant, uniformly sampled at 30° increments, thus allowing a complete 3D reconstruction of the plant. We roughly align adjacent point clouds from adjacent range scans by performing sequential pairwise registration and then use our global method to create the final point cloud.

Inspired by Toldo *et. al.*'s work [140] in rigid registration, we solve this problem by constructing an "average" scan to which we register. For a scan X, we find the set of points that are Mutual Nearest Neighbors (MNN) for each point in the scan from every other scan. That is, we compute the MNNs for each point in X for each scan individually, and we then combine them into a single scan that is composed of the calculated centroids from each point. We describe this in Section 3.3.

We first approximately align the scans sequentially, and then we use a global method to refine our result. The global method involves registering each scan X_i to an "average" shape, which we construct using the centroids of the *mutual nearest neighbors* (MNN) of each point. For X_i, we use scans X_j where $j \neq i$ to obtain the average shape Y_{cent} from the centroids, and X_i is then registered to this average shape. This is repeated for every scan until the result converges.

We modify Equation (3.4) to perform global registration

$$p(x|i) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp[-\frac{||x - \hat{y}_i||^2}{2\sigma^2}]. \tag{3.10}$$

where $\hat{y}_i \in Y_{cent}$ are the points in the target scan Y_{cent}, which is constructed from all scans other than itself.

For a pair of scans X and Y, we say that a point $x_i \in X$ and $y_j \in Y$ are MNN if $x_i = x_{i_n}$ and $y_{j_n} = y_j$, where

$$x_{i_n} = \min(|x_p - y_j|), \forall x_p \in X, \tag{3.11}$$

and

$$y_{j_n} = \min(|y_q - x_i|), \forall y_q \in Y. \tag{3.12}$$

For each point x_j in scan X_i, we find the set of points $\{x_k | x_k \in X_l \wedge MNN(x_k, x_j)\}$, where $l \neq i$, i.e. all scans other than X_i. For each of these sets of points x_k, we find the centroid

$$x_{cent} = \sum_{i}^{n} \frac{x_k}{n}. \tag{3.13}$$

We register X_{cent}, the set of centroids calculated for each x_j, to scan X_i.

Although CPD alone is effective in registering pairs with a fair amount of overlap, when registering multiple scans, especially scans that have not been pre-aligned, our method achieves a much better fit both visually and quantitatively than CPD by itself, utilizing sequential pairwise registration. Our method is a two step process, beginning with aligning the scans approximately. We then register a single scan to "average" shape, constructed from all other scans, and update the set to include the newly registered result and performing the same process with all other sets of scans. In this way, we avoid accumulation of merging error.

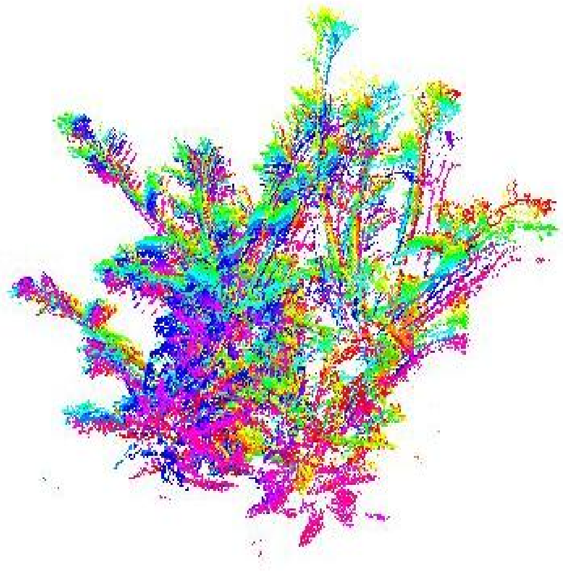


Figure 3.3: 12 scans of the Arabidopsis plant, prior to registration, but with rotation and and translation pre-applied. Different colours indicate different scans.

3.3.1 Approximate Alignment

We capture a set of scans around the plant at 30° increments. After acquiring them, we first solve for the rigid transformation $T_0 = (R_0, \vec{t_0})$ (where R is a rotation angle and $\vec{t}$ is a translation vector) between the the first scan (X_0) and the second scan (X_1) using the rigid version of CPD. After we solve for $\vec{t_0}$, for each scan X_i, we apply the transformation i times as follows:

$$\hat{X}_i = R_i X_i + \vec{t_i}, \tag{3.14}$$

where $R_i = \prod_{k=0}^{i} R_0$ and $\vec{t_i} = \sum_0^i \vec{t_0}$. Our new set of transformed scans $\hat{X}$ should now be roughly aligned. We use this method to obtain a rigid registration. The initial registration is important when the pair of scans to be registered has minimal overlap.

The approximately aligned scans can be seen in Figure 3.3.

3.3.2 Global Non-Rigid Registration via MNN

Once the initial registration is complete, we use CPD in conjunction with MNN to recover the non-rigid deformation field that the plant undergoes between the capture of each scan. At this point, the scans should be approximately aligned to one another. We now construct the centroid/average scan and then register to it.

Global Registration

We use Algorithm 1 to merge all scans, where MNN$(\cdot)$ computes the mutual nearest neighbor for each point in scans X_i and X_j and the centroids function likewise takes the centroids computed for each point in each scan and combines them into one average scan using Equation

(3.13). For each point in scan X_i, we find the single nearest neighbor from all other scans and use the set of distances to compute the L^2-norm.

Algorithm 1 MNN Registration

```
Require: $\mathbf{X} = [X_1, \cdots, X_n]$, where each $X_i$ is a range scan that has been approximately adjusted. A predefined tolerance $tol_{max}$.
  $tol = \sum_{i=1}^{N} \text{error\_}L^2(X_i)/N$
  while $tol < tol_{max}$ do
    for $i = 1$ to $N$ do
      for $j = 1$ to $N$ do
        if $j \neq i$ then
          $X_{i_{cent}} = \text{MNN}( X_j, X_i )$
        end if
      end for
      $X_{cent} = \text{centroids}( X_{1_{cent}}, \cdots, X_{N_{cent}} )$
      $X_i = \text{register\_cpd}( X_{cent}, X_i )$
    end for
  end while
```

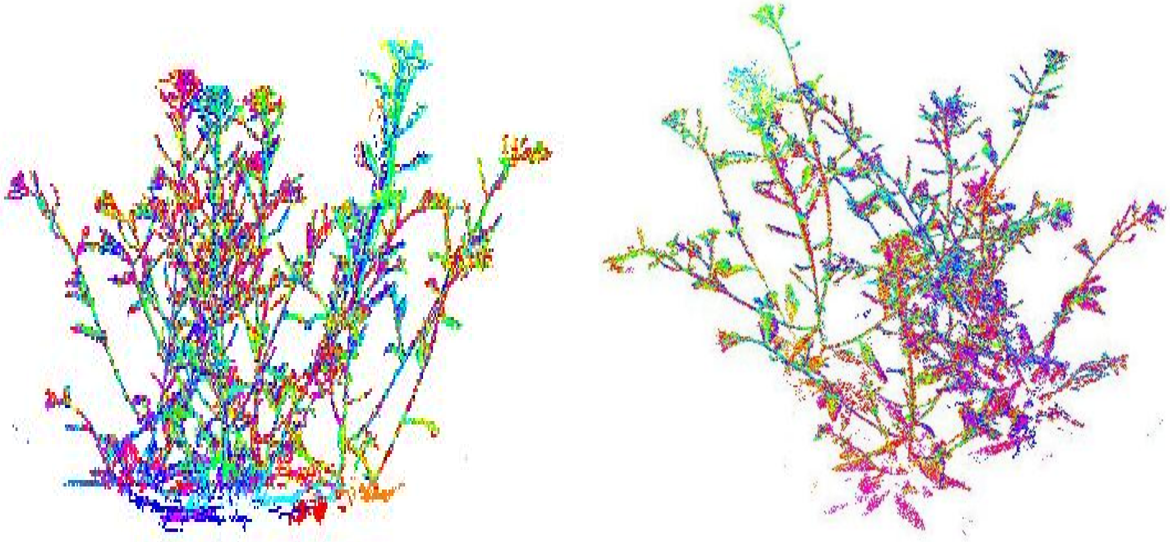


Figure 3.4: 12 scans captured in 30° increments about the plant and then merged into a single point cloud using MNN. Shown from two viewpoints, front facing on the left and from above on the right. Different colours indicate different scans.

3.4 Results

We show the performance of MNN versus sequential pairwise registration for plant and vascular data in the following 2 subsections.

3.4.1 Plant Data

Figure 3.4 shows all 12 scans, merged into a single point cloud after subsampling each scan. Each color in the point cloud represent a different scan. Despite the noisiness of the range scans from jitter, our method successfully performed the 12 view registration, resulting in a single point cloud that accurately captures the shape of the Arabidopsis plant. By ensuring that the scans are all approximately registered before proceeding (for both methods), we minimize the likelihood that erroneous parts of the point cloud datasets will bias the motion of a scan that is being registered. Figure 3.5 displays the resultant error between the first scan in the set and all subsequent scans. First, we see that the error is lower for scans that have more overlap (the first scan shares a fair deal of overlap with the last, for example) for both sequential pairwise and our proposed methods. We see that our method always outperforms its pairwise counterpart. Sequential registration still rendered a useful result, though as the number of scans grows, the drift would theoretically increase.

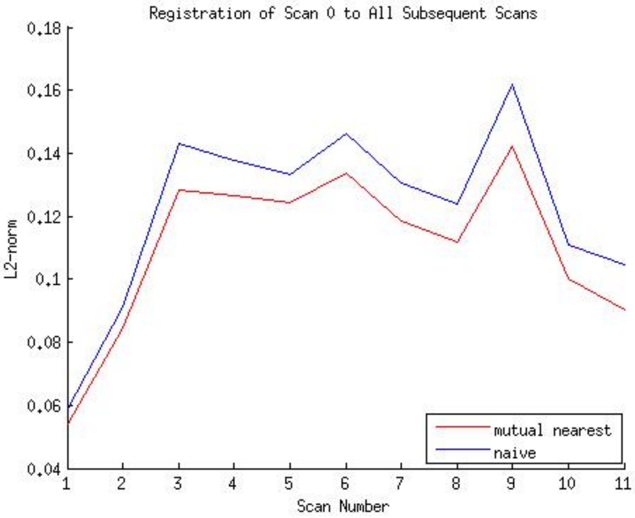


Figure 3.5: MNN versus sequential pairwise registration.

3.4.2 Synthetic Vascular Data

We further demonstrate the efficacy of our method on synthetic vascular data. We take a 3D point cloud of synthetic veins, as generated by VascuSynth [144] and apply a non-rigid deformation to the point cloud to create a total of 20 scans. This was performed using the deformation method provided with the CPD software, which essentially constructs a matrix from random data and uses that to construct a displacement field to apply. We use the parameters

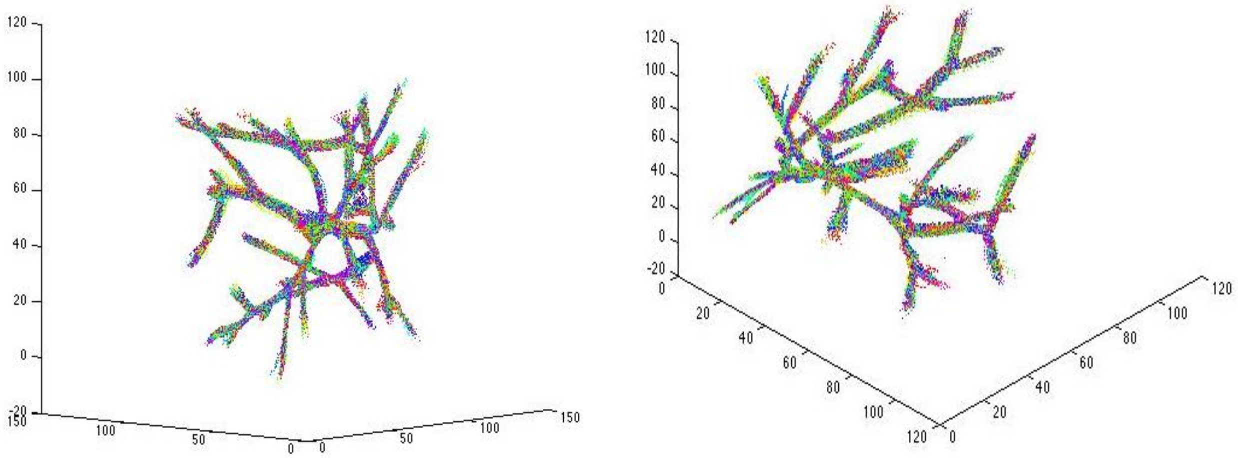


Figure 3.6: 20 synthetic scans of vascular data merged into a single point cloud using MNN, as seen from two viewpoints, front and back.

sampling = 0.1, **power** = 3 and λ = 4. Initially, σ = 5 and we increase its value by 0.5 for each successive deformation. This gives us a new set of point clouds, as seen in Figure 3.6. The magnitude of this transformation is substantially larger than those created by the wind in the Arabodopsis set. We utilized more scans than we did with the Arabidopsis in hopes of verifying our hypothesis that pairwise registration gets progressively worse as we add more scans, though the effect of drift is partially obscured by the fact that the deformation increases between the first and each successive scan.

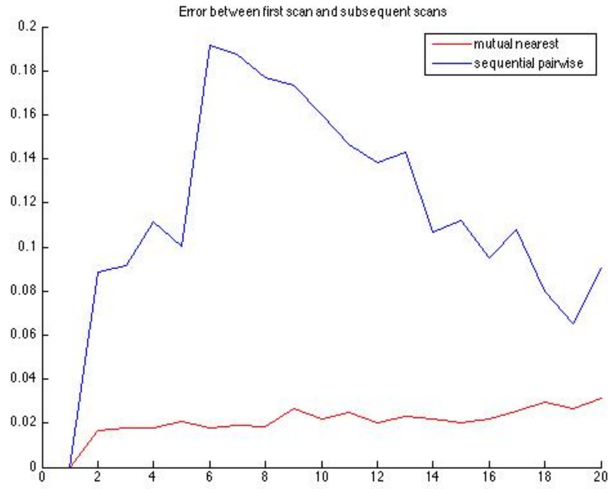


Figure 3.7: MNN versus sequential pairwise registration on vascular data, registering the first scan to each of the subsequent scans.

The L_2 error of the merged scans using sequential pairwise was higher than MNN and the resultant shape no longer looked like the initial one when we used the default CPD parameters

($\lambda = 1, \beta = 1$), where λ controls the "stiffness" and β controls the point "spread". In order to maintain the shape of the veins, we couldn't use a value of $\lambda < 70$. The quantitative results for the pairwise sequential method in Figure 3.7 were calculated using $\lambda = 90$, which still ends up rendering a badly warped result. In addition, as we increased the value of λ, the drift increased quickly.

By contrast, MNN performs very well on this data, as seen in Figure 3.6. Quantitatively, we have shown that it easily outperforms sequential pairwise registration, that our method limits drift.

Chapter 4

Machine Vision System for 3D Plant Growth Analysis

Machine vision for plant *phenotyping* is an emerging research area for producing high throughputs in agriculture and crop science applications. Since 2D based approaches have their inherent limitations, 3D plant analysis is becoming state of the art for current *phenotyping* technologies. We present an automated system for analyzing plant growth in indoor conditions. A gantry robot system is used to perform scanning tasks in an automated manner throughout the lifetime of the plant. A 3D laser scanner mounted as the robot's payload captures the surface point cloud data of the plant from multiple views. The plant is monitored from the vegetative to reproductive stages in light/dark cycles inside a controllable growth chamber. An efficient 3D reconstruction algorithm is used, by which multiple scans are aligned together to obtain a 3D mesh of the plant, followed by surface area and volume computations. The whole system, including the programmable growth chamber, robot, scanner, data transfer and analysis is fully automated in such a way that a naive user can start the system with a mouse click and get back the growth analysis results at the end of the lifetime of the plant. As evidence of its functionality, we show quantitative results of the rhythmic growth patterns of the monocot *Arabidopsis thaliana*(L.), and the dicot barley (*Hordeum vulgare* L.) plants under their diurnal light/dark cycles. This evidence supports our claim that we can obtain biologically relevant data from the proposed system. As far as we know, we are the first to capture the full 3D structure of a plant as a single closed 3D triangular mesh using a (near-infrared) laser scanner.

The work in this chapter was initially published in a conference and and subsequently expanded in a submitted journal version as follows:

- A. Chaudhury, C. Ward, A. Talasaz, A.G. Ivanov, N.P.A. Hüner, B. Grodzinski, R.V. Patel, and J.L. Barron, *"Computer Vision Based Autonomous Robotic System for 3D Plant Growth Measurement"*, In Proceedings of the 12th Conference on Computer and Robot Vision (CRV) 2015.

- A. Chaudhury, C. Ward, A. Talasaz, A.G. Ivanov, M. Brophy, B. Grodzinski, R.V. Patel, N.P.A. Hüner and J.L. Barron, *"Machine Vision System for 3D plant phenotyping"* (in review).

4.1 Introduction

With the advancement of robotic technologies, automation tasks have become easier. Apart from automating the phenotyping process, the data collection can also be automated efficiently in real time. However, there are a number of challenges involved in accomplishing this, such as communication among the hardware devices, reliable data transfer and analysis, fault tolerance, etc. We have developed a 3D plant phenotyping vision system which is capable of monitoring and analyzing a plant's growth over its entire life-cycle. Our system has several parts. First, a gantry robot system is used for the automation of data collection process. The robot is programmable and can be moved around the plant by specifying a particular trajectory. A 3D laser scanner is the robot arm's payload. The scanner can record 3D point cloud data from a number of viewpoints about the plant. The robot moves from one viewpoint to another, and communicates with the scanner to take a scan of the plant under observation. In the current setup, the plant is scanned six times a day from 12 viewpoints at 30° increments about 360° (we obtain six 3D triangular meshes a day). The 12 viewpoints result in overlapping range data between adjacent views, allowing the merging of all the views into a single 3D triangular mesh representing the whole plant. Note that our laser scanner uses a near infrared beam (about 825*nm*) and can scan the plant in the light and in the dark equally well.

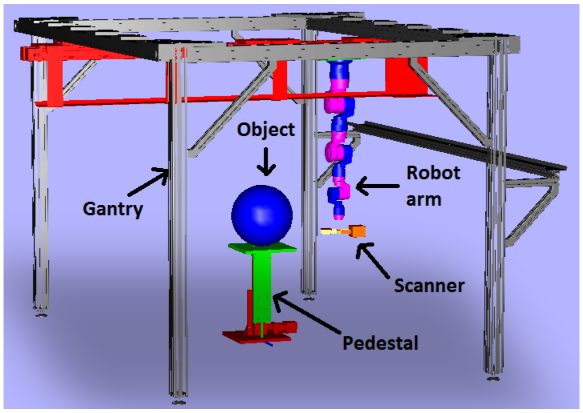


Figure 4.1: Schematic diagram of the gantry robot system

We report growth results and analysis for 3 weeks of the life-cycle of wild type *Arabidopsis thaliana*(L.) and barley plants. Without proper environmental conditions (e.g. light, temperature, wind, humidity, etc.), the plants would not have thrived in an indoor environment. The reason for using the Arabidopsis plant is that, this plant has used extensively as a model plant in biology and was the first plant whose genome was completely sequenced [138].

Reconstructing a plant's 3D model from multiple views is extremely challenging [19, 18, 20, 169]. Unlike using a rigid model like the Stanford bunny, reconstructing a highly non-rigid thin plant is difficult. We use a new multi-view alignment (registration) algorithm on individual 3D point clouds obtained by our scanning to obtain a 3D triangular mesh of the growing plant. Given this mesh, we can compute the 3D surface area and volume of the plant. We propose that area and volume make good plant growth metrics.

In the next section we explain the system components in detail, followed by their integration into a complete system and finally the operation of that system. Subsequently the multi-view reconstruction algorithm is discussed and we show experimental results and derive some conclusions based on those results.

There is a large body of work on imaging based plant *phenotyping*. We refer to Chapter 2 for the related work. Most of these methods have several limitations. None of these systems are designed to monitor a plant's growth for its whole lifetime in an automated manner using 3D imaging technique. To the best of our knowledge, we are the first to report a fully automated system which operates in near real time[1] over the lifetime of a plant using laser scanning technology. We present a novel approach to study plant growth in truly automated manner using 3D imaging technique. The system is described in next section.

4.2 System Description

The proposed system has several parts which are integrated to make a fully autonomous system. Each component is explained separately below.

4.2.1 Gantry Robot

A schematic diagram of the robotic system (manufactured by *Schunk Inc.*, Germany) is shown in Figure 4.1 comprising an adjustable pedestal and a 2-axis overhead gantry carrying a 7-DOF robotic arm. The plant is placed on the pedestal which can be moved up and down to accommodative different applications and plant sizes.

The 7-DOF robotic arm in Figure 4.2 provides a high level of flexibility for controlling the position and orientation of the 3D scan head, while the 2-axis gantry provides an extended workspace. The specification details of the robot is shown in Table 4.1.

Table 4.1: Robot Specifications

Degrees of freedom		7
Dimensions	x	$3m$
	y	$2m$
	z	$1.8m$
Accuracy		$0.1\mu m$
Repeatability		$0.1\mu m$
Speed		1m/s
Payload		10 kg

---

[1] Once the range scanning is complete, we start processing the data immediately. Due to the complexity of the reconstruction algorithm, it takes about $2 - 3$ hours to align each set of 12 views of a plant (more fully grown plants take $7 - 8$ hours). Using a cluster of computers provided by SharcNet, at the end of the lifetime of the plant, the user has all the processed results within at most four hours. SharcNet is a supercomputing facility available to researchers at the University of Western Ontario having many clusters where jobs can be run in parallel, see https://www.sharcnet.ca/

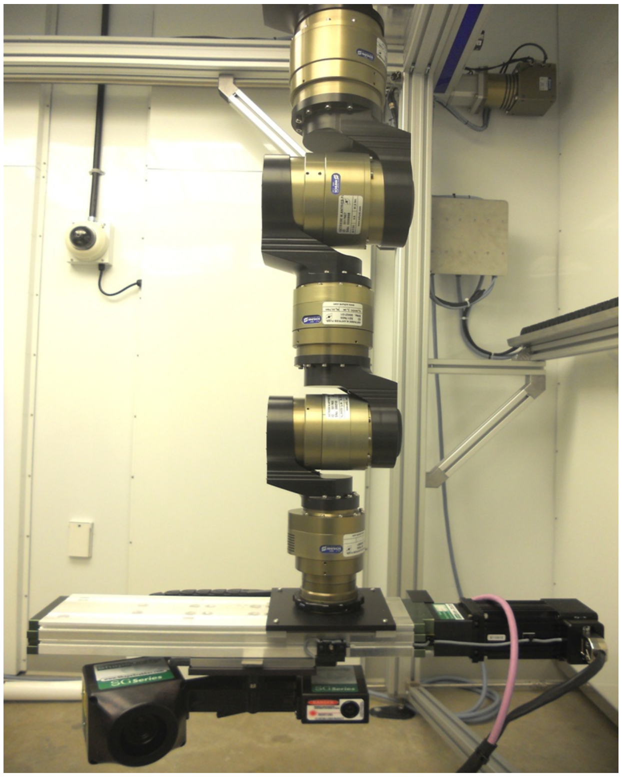


Figure 4.2: Robot arm

We have programmed the robot to move in a circular trajectory around the plant to take scans. Initially the robot stays in its home position with the arm resting vertically downwards (Figure 4.3). After the initiation of the commands, it moves from home position to the desired location by alternating macro and micro joint movements.

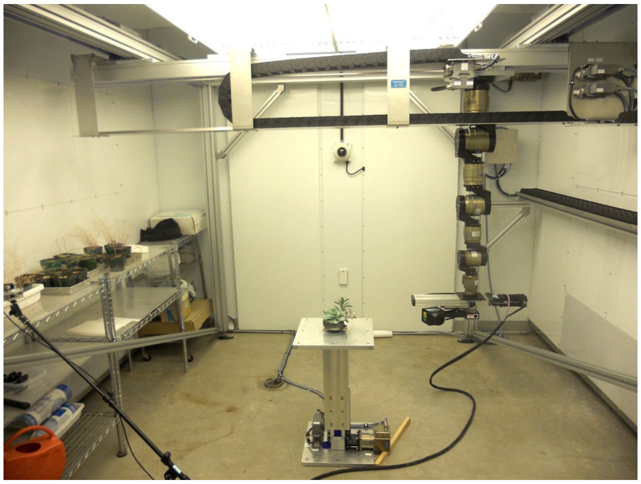
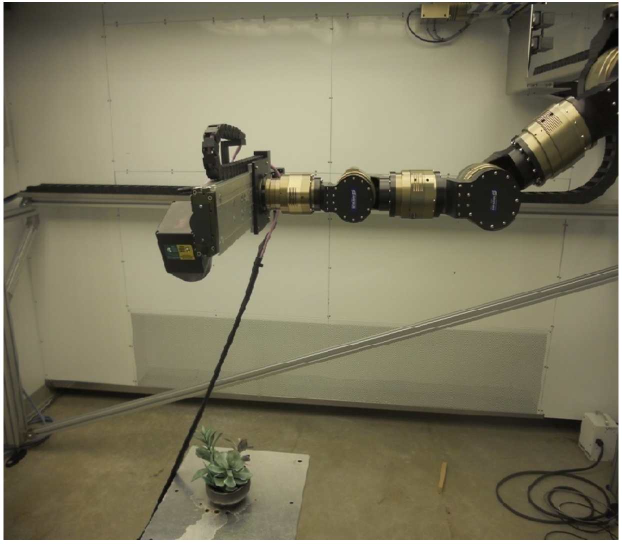


Figure 4.3: Robot room where the experiment was performed

Top Scanning



Side Scanning

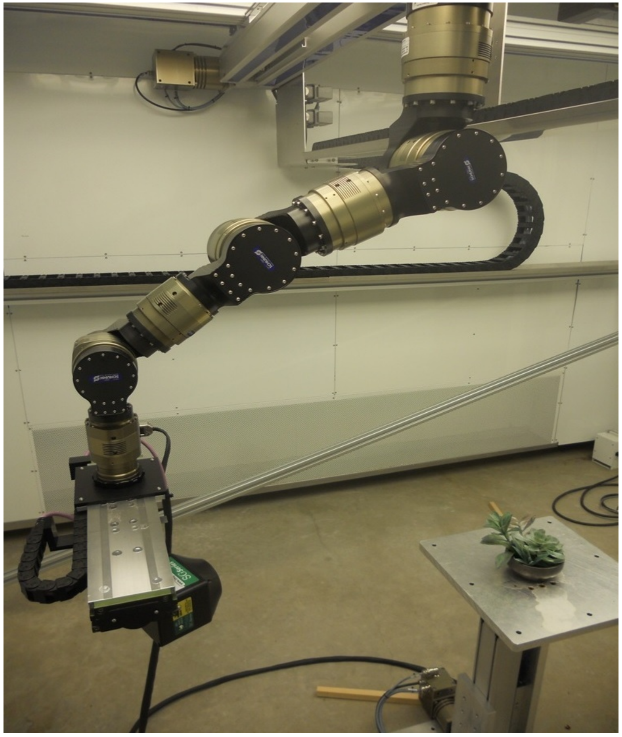


Figure 4.4: Different scanning positions of the robot arm around the plant

Two different positions of the robot arm are shown in Figure 4.4. The upper picture shows the position of the arm for scanning the plant from the top when the plant is in the vegetative stage. The lower picture shows the position of the arm used for side scanning. We maintain a horizontal distance of $0.56m$ from the center of the pedestal to the vertical axis of the robot arm, and $0.26m$ from the pedestal plane to the scanner (Figure 4.5). These distances were set empirically to obtain good scan data.

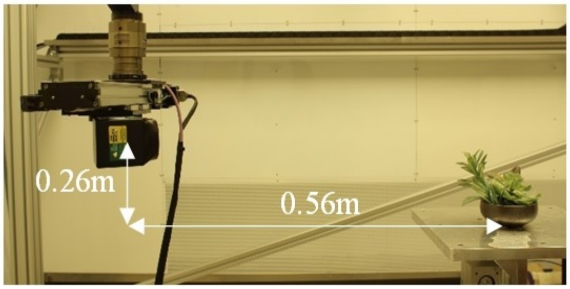


Figure 4.5: scan head distance from the plant

4.2.2 Scanner

We use a *SG1002 ShapeGrabber* range scanner, which is the payload of the robot arm. This scanner can measure dense depth maps of the visible surface of an object in point cloud format. The scanner uses near-infrared light at 825*nm* (versus visible red light at 660*nm* for most scanners) which has minimal effects on plant growth. Different parameters of the scanner (e.g. Field of View, laser power, etc.) are set empirically. We have performed the whole experiment with laser power 1.0*mW* (means the laser has a beam radius of 1.0*mm*). The scanner software Communicates over a UDP (User Datagram Protocol) link with the robot control software. Each time, after the robot stops at a scanning position, it communicates with the scanner to take a scan and then moves to the next position.

4.2.3 Growth Chamber

We have designed the whole robotic set-up inside a growth chamber (manufactured by *BioChambers Inc.*, Canada). The chamber is fully programmable allowing control of the temperature, the humidity, the fan speed and light intensity. Also, it can be monitored remotely using an IP camera. The chamber is 5.2m^2 and equipped with a combination of 1220*mm* T5HO fluorescent lamps and halogen lamps The whole chamber is a dedicated embedded system, and can be controlled from the robot control software,

4.3 Alignment of multi-view scans

Multi-view alignment is a major task in building a 3D model of an object. Pairwise registration is a crucial part in performing multi-data alignment and this has been studied extensively in the computer vision literature [33, 141, 160, 131]. However, registration of thin non-rigid plant structure is very challenging and little studied. Although Iterative Closest Point (ICP) [14] and its variants [142, 68, 17] have been successful in some cases, registering highly non-rigid thin plant structures is still problematic.

We use the alignment algorithm discussed in Chapter 3. However, the basic assumption of the alignment algorithm is that, in order to obtain reasonable result, the scans should be approximately aligned to each other. In other words, the initial guess of the transformation parameters should be good. Although the multiple view registration algorithm works well in aligning different scan data, registering two views with huge rotation angle difference can pose difficulties. Unfortunately after decades of research on point cloud registration, state of the art algorithms fail when the views are not roughly aligned. The problem is more challenging for the cases of complex plant structures due to occlusion and local deformation between two views [20, 169, 29]. One approach to estimate the rough alignment of two views is to find corresponding feature points. However, because of the complex structure of the plants, finding repeatable features is extremely hard [38] and typical feature point matching algorithms fail.

Junctions are strong features for plant-like structures. We adopt the idea of using junction point of branches as feature points and then match these features [29] (we refer to Chapter 5 for the details). However, the idea in [29] for detecting junction points does not consider occlusion in leaves. This results in false feature point detection, which may not be repeatable in two views. We adopt the algorithm using a simple but effective density clustering technique as discussed below.

4.3.1 Feature Clustering

The basic idea of junction detection algorithm [29] is to first extract local neighbourhood around every point using *kd*-tree and perform a statistical dip test (refer to Chapter 5 for details) to determine the non-linearity in the data. Then the branches are approximated by fitting straight lines to the point cloud. Finally the straight line equations are solved to find out the junction point. This approach results in detection of multiple feature points around the junction, because all the points around a small neighbourhood at the junction are potential candidates of true junction points, from which the best candidate is picked up by non-maximal suppression of the dip value. But this idea results in detecting false feature points as junctions in occluded areas, especially in leaves. We handle the problem by applying density based clustering to extract the true junctions and filter out rest of the feature points.

The idea of density based clustering is to find out the group of points which are denser than the remaining points. As true junction features tend to appear in higher density than false junctions, we intend to cluster the detected feature points to find out the cluster of points that are formed at true junction points. We use a density based clustering algorithm proposed by Ester *et al.* [45]. The algorithm does not need the number of clusters in advance (unlike *k*-means) and can perform clustering in presence of large number of outliers. Finally, we compute the centroid of each cluster to find the true junctions and match these features using the subgraph matching technique as in [29]. Figure 4.6 illustrates this idea. It shows a single view of a plant having occlusions. Red dots represents feature points detected by junction detection algorithm [29]. The clustering algorithm detects clusters around true junction points (denoted by blue circles) and discards rest of the points as outliers.

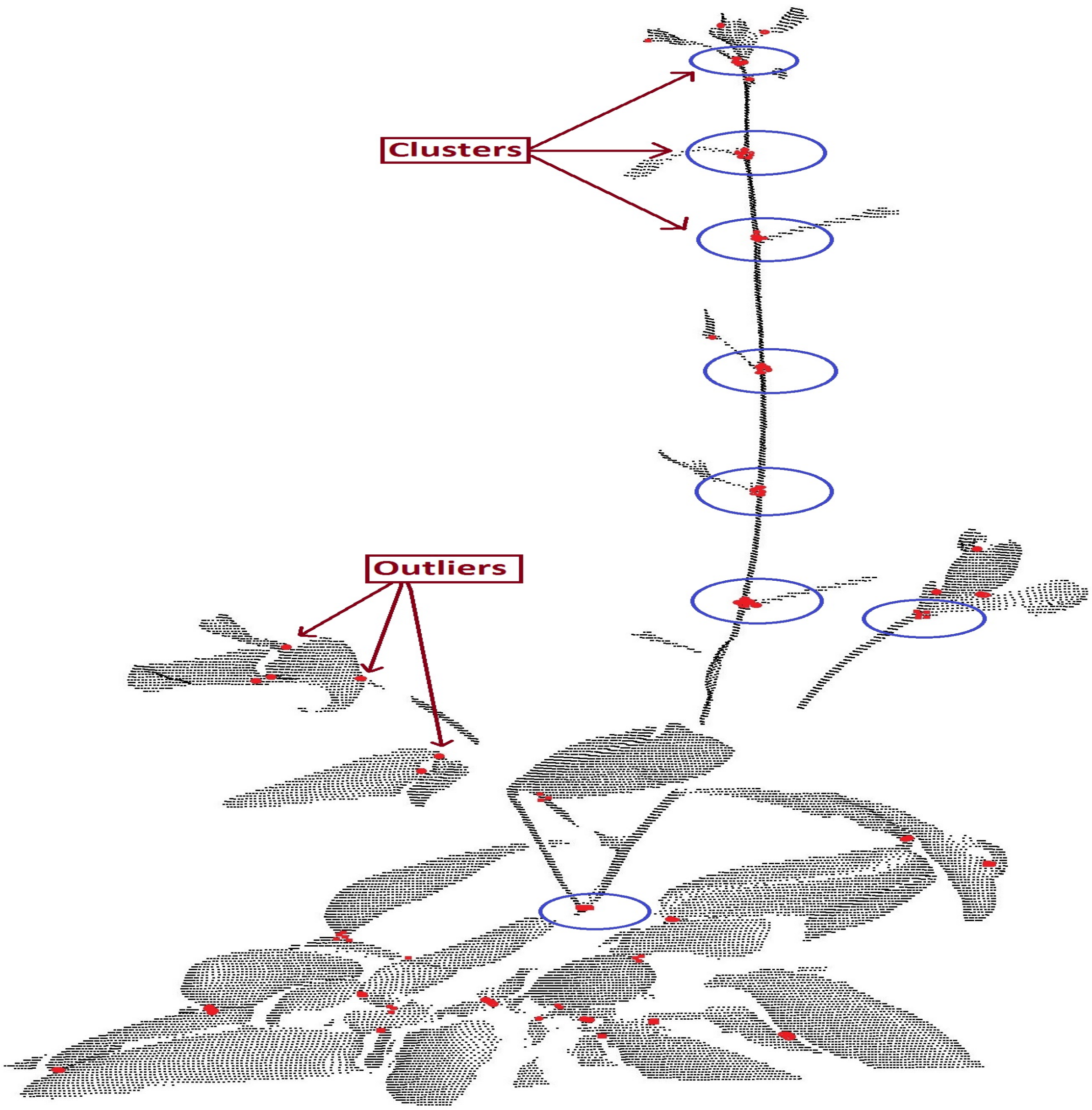


Figure 4.6: Demonstration of feature clustering. Red dots represent junction feature points [29]. Density clustering algorithm [45] detects clusters at true junctions (denoted by blue circle) and treats false feature points as outliers.

4.4 System Integration

We integrate the chamber, robot and scanner. The system components and their connections are shown in Figure 4.7. The robot and scanner are operated from different computers which communicate over a dedicated UDP link. The chamber is accessed remotely over the internet.

Communication between the chamber and robot operation needs to be done frequently. While scanning a plant, we need to shut down or significantly decrease the fan speed inside the chamber (at present we just shut it down completely), otherwise the scan data will be erroneous as the plant will be jittering (this makes the multiple view reconstruction problem extremely

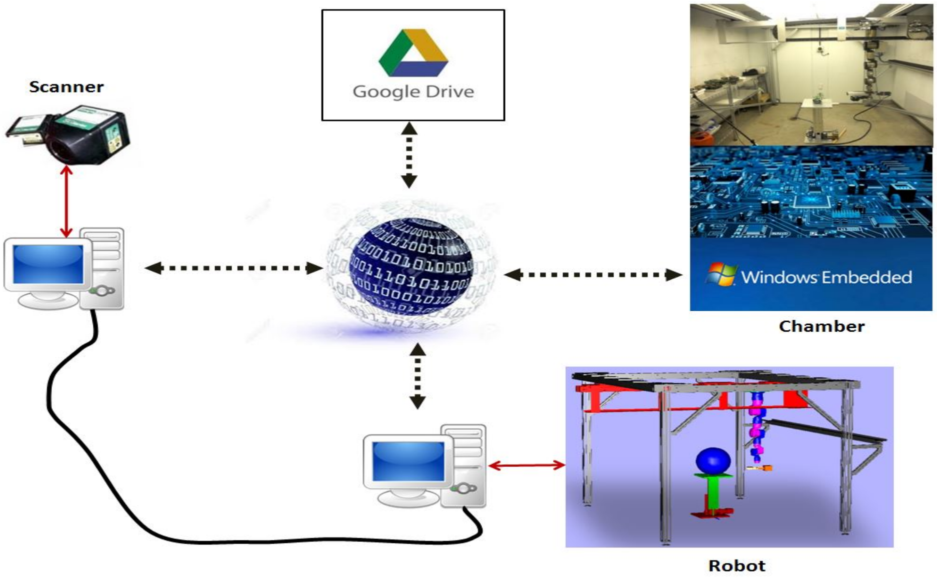


Figure 4.7: High level view of the system

dificult). Also, experimenting in different lighting conditions (short day versus long day) needs communication between the chamber and the scanning schedule of the robot. Before starting the experiment, chamber parameters are set according to the need of the application. During the experiment before a set of scans, the robot communicates with the chamber, turns off the fan (and light if needed), and restores the default chamber settings when the scan is completed. This process is repeated at each scan.

As the size or dimension of the plant is not known in advance, determining the scanning boundaries to enclose the whole plant needs to be done dynamically during each scan. Moreover, as the plant grows, it might lean towards a particular direction, which needs the scanner position to be adjusted accordingly. We perform a simple bounding box calculation before performing each scan (Figure 4.8). Before doing the actual scanning, a pre-scan procedure is performed from 2 directions (front and side as shown in the figure). From these scans, the centre of the bounding box of the plant is approximated and used to update the centre of rotation for the circular scanning trajectory. Once the plant centre has been determined, the system causes the gantry and robotic arm to translate and rotate the scan head to specified discrete positions around the plant.

At each scanning position, the system waits 10 seconds to allow the plant and the scanner to settle before initiating a scan. Once a scan has finished, the resulting scan data are analyzed to ensure that the plant has been fully captured (i.e. there is no clipping). Sometimes the scanner FOV is not wide enough to capture the full width of the plant due to a limited travel of the scanner's linear stage ($0.2m$). In that case, we perform more than one scan of the view by sliding the scanner in a sideways direction. From empirical observations, for plants like Arabidopsis, no more than 3 partial scans are needed to enclose a single view as illustrated in Figure 4.9. Usually, if the plant is not too wide, a single view from P is sufficient. Otherwise, we perform side scans at positions P_1, P_2 and P_3.

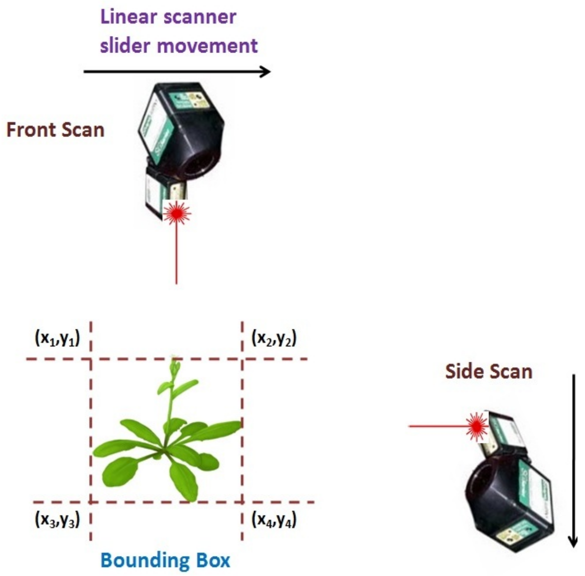


Figure 4.8: Computing the bounding box to determine the center of the plant

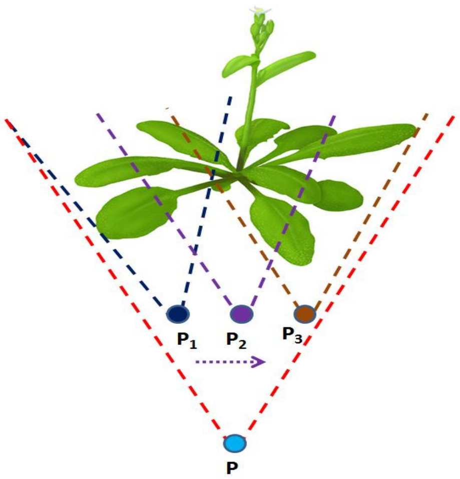
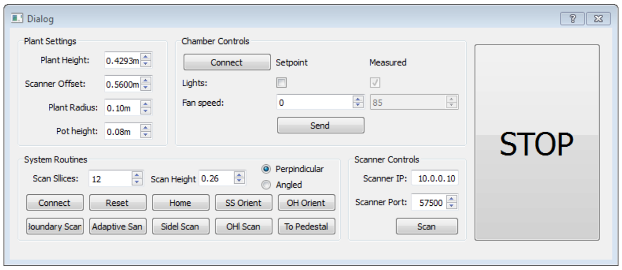


Figure 4.9: Scanner Field of View (FOV) to enclose the whole plant. When the plant is big, one scan can't capture the whole plant and multiple scans are required

We operate the whole system from a single script. The user controls everything from the GUI (Figure 4.10). Once the system is started, it continues working until the experiment is terminated.

4.5 Experimental Results

First we have performed a 21 day experiment with a wild type Arabidopsis plant (after which the plant was almost dead and was falling down) with a 12/12 hours light/dark cycle at the



Plant Height: 0.4293m
Scanner Offset: 0.5600m
Plant Radius: 0.10m
Pot height: 0.08m

Setpoint / Measured
Lights: ☐ ☑
Fan speed: 0 / 85

Scan Slices: 12
Scan Height 0.26
● Perpindicular
○ Angled

Scanner IP: 10.0.0.10
Scanner Port: 57500

Figure 4.10: System GUI or user interface

temperature of 25°C and a light intensity of 250μmol photons $m^{-2}s^{-1}$. We also performed another 15 day experiment with barley under the same conditions. Although the algorithms used in the different stages are not new, the main challenge was to make the whole system work continuously for the life-cycle of the plant. We encountered several problems while integrating the system parts (robot, chamber, scanner, etc). As the robot and the chamber needed to communicate frequently (every 4 hours as we are performing 6 scans per day, throughout the life of the plant), there were issues of lost communication and hardware faults (e.g. problems in electronic chips at robot joints, etc).

4.5.1 Merging of Multi-View Plant Point Cloud

Using the reconstruction algorithm discussed above, we performed alignment of the multi-view point cloud data of the growing plant over time. We used Sharcnet[2] computing machines for simultaneous processing of large amount of data while the scan data was collected throughout the life of the plant. The merged point cloud data for fully grown Arabidopsis plant (day 20th) and the barley plant (day 15th) are shown in Figures 4.11 and 4.12 respectively. Each colour represents different scans.

4.5.2 Polygonal Mesh Formation

Once we have the aligned point cloud from the multi-view data, it needs to be triangulated in order to compute the mesh surface area and volume. Accurate triangulation of point cloud data is a challenging problem. An efficient triangulation should represent all the details of the shape of the object. Triangulation of plant structures is more challenging due to the thin branches. Although Delaunay triangulation is typically used for modeling a surface, the algorithm does not produce good result for plant structures. We used the α-shape algorithm [44] for triangulation. The algorithm works well when its parameters are properly tuned.

α-Shape Triangulation

Let $P = \{p_1, \cdots, p_n\} \subset \mathbb{R}^d$ be a set of points, which are called *sites*. A *Voronoi diagram* is a decomposition of $\mathbb{R}^d$ into convex polyhedra. Each region or *Voronoi cell* $\mathcal{V}(p_i)$ for p_i is defined to be the set of points x that are closer to p_i than to any other site. Mathematically,

---

[2] https://www.sharcnet.ca/

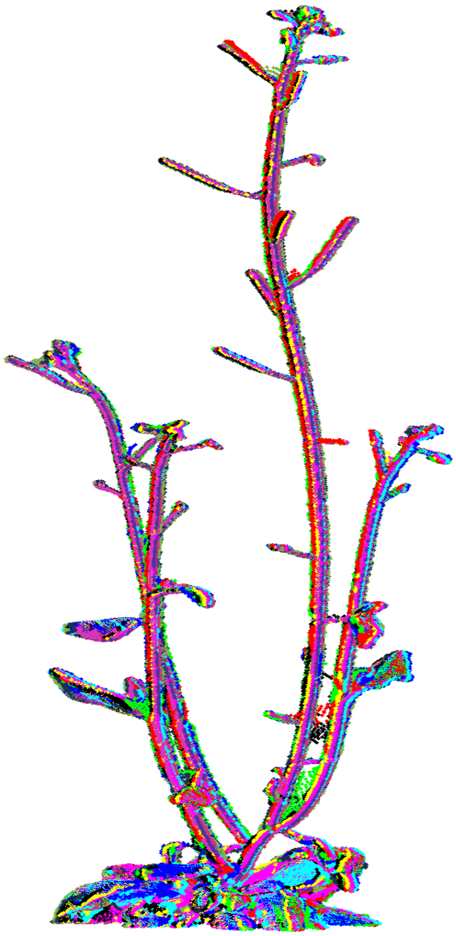


Figure 4.11: Reconstructed Arabidopsis plant point cloud (different colors indicate different scans). Note that there was no plant jittering in our setup as the wind could be fully controlled, unlike for the setup used in Brophy's work (described in Chapter 3).

$$\mathcal{V}(p_i) = \{x \in \mathbb{R}^d \mid \|p_i - x\| \leq \|p_j - x\| \; \forall j \neq i\},$$

where ||.|| denotes the Euclidean distance. The *Delaunay triangulation* of P is defined as the dual of the Voronoi diagram.

The *α complex* of P is defined as the Delaunay triangulation of P having an empty circumscribing sphere with squared radius equal to or smaller than α. The α shape is the domain covered by alpha complex. If $\alpha = 0$, the α-shape is the point set P, and for $0 \leq \alpha \leq \infty$, the boundary $\partial\mathcal{P}_\alpha$ of the α-shape is a subset of the Delaunay triangulation of P. The main idea of the algorithm is that the space generated by any point pairs can be touched by an empty disc of radius α. The value of α controls the level of detail in triangulation. The algorithm is simple and effective. We have empirically chosen $\alpha = 0.6$ for the plants. Also note that we have already performed "smoothing" of the point cloud while applying Gaussians, so we do not need further surface smoothing (such as Poisson smoothing). Example results are shown

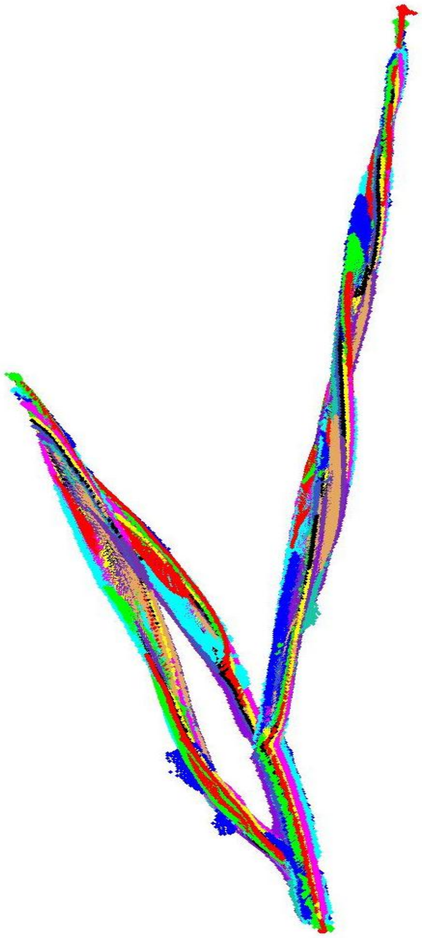


Figure 4.12: Reconstructed Barley plant point cloud (different colors indicate different scans)

in Figures 4.13 and 4.14. The rectangular cutouts show some smaller portions of the plants at higher resolution. The surface area is simply the summation of area of each triangle in the mesh and volume can be computed using the technique described by Zhang *et al.* [159]. The "loop" structure is due to the angle by which the barley plant is viewed, some of parts of the plant occludes other parts.

4.5.3 Biological Relevance

The fact that plants grow mostly at night is well known [106]. It is observed that the changes in stem diameter depends on the lighting conditions [55, 128, 56]. While the diurnal nature can involve changes in stem length, width, diameter, leaf surface area, we have observed the diurnal pattern in both volume and surface area of the plant. The mesh surface area and volume are plotted against time in the same graph for Arabidopsis plant in Figure 4.15. A similar plot for barley plant is shown in Figure 4.16. In the graphs, red dots represent night time scans and blue dots represent day time scans. As we have 6 scans per day, there are 3 blue dots followed by 3 red dots in the graph. For the Arabidopsis experiment we had 4 scans missing due to

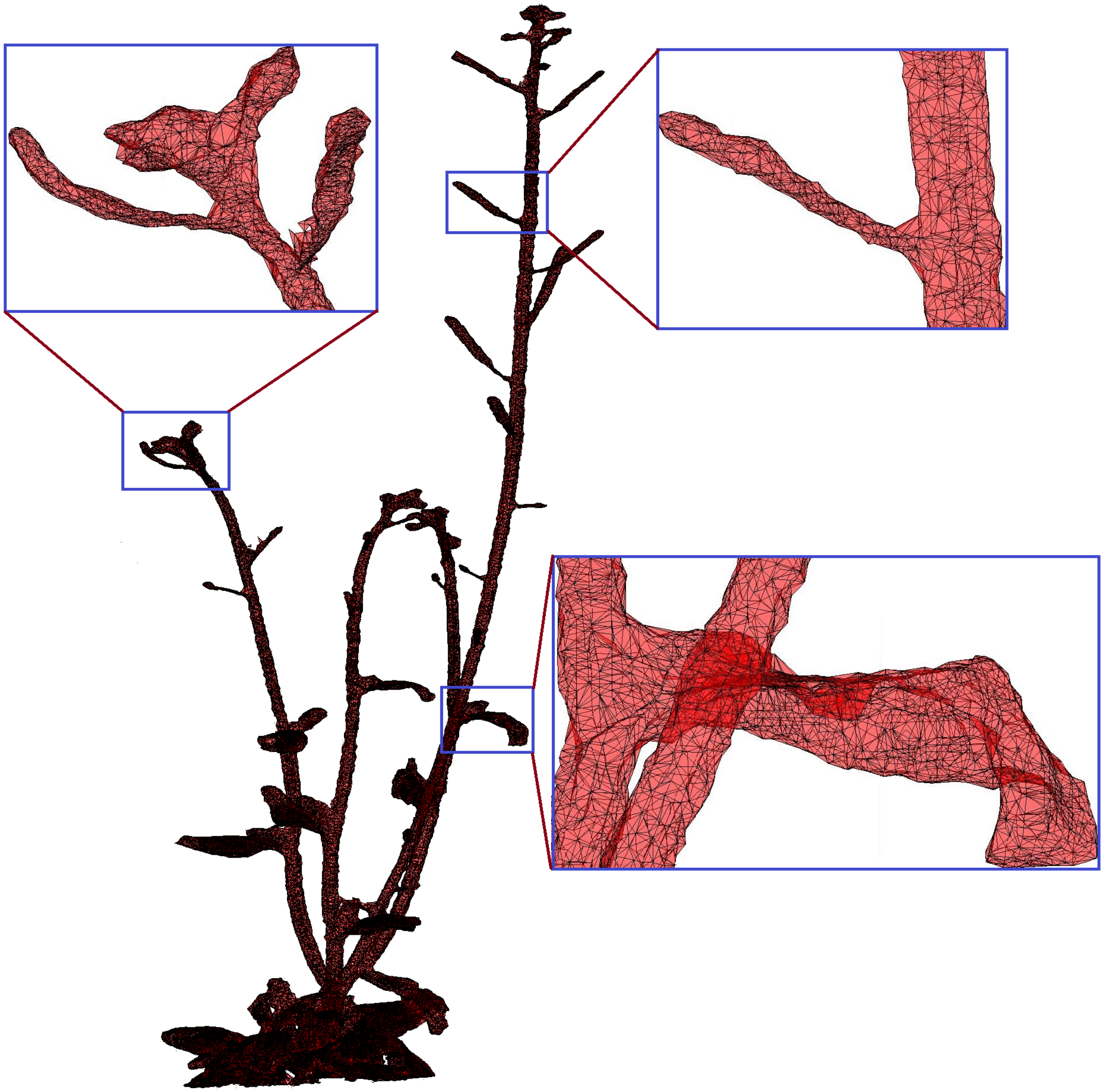


Figure 4.13: Triangulated Arabidopsis plant data

networking problems. These missing data are generated by taking the average of previous and next scan data. These are shown as green dots in Figures 4.15 and 4.16.

It can be noticed from the growth curves that the plants exhibited more growth in the night time than in the day time, which supports the biological relevance of diurnal growth pattern of plants. Finally, note that the changes of volume are greater than the changes of surface area in the later period of the growth cycle (this is logical as volume grows faster that area).

The initial short stage of plant growth looks linear, the long intermediate stage of plant growth looks exponential while the short end stage of plant growth looks stationary (or constant). Often, biologists compute the **growth rate** as the logarithm of mesh surface area values and then fit a straight line to this data, yielding the maximum exponential growth rate. Figures 4.17a and 4.17b show the surface area and volume growth rates for the Arabidopsis plant while

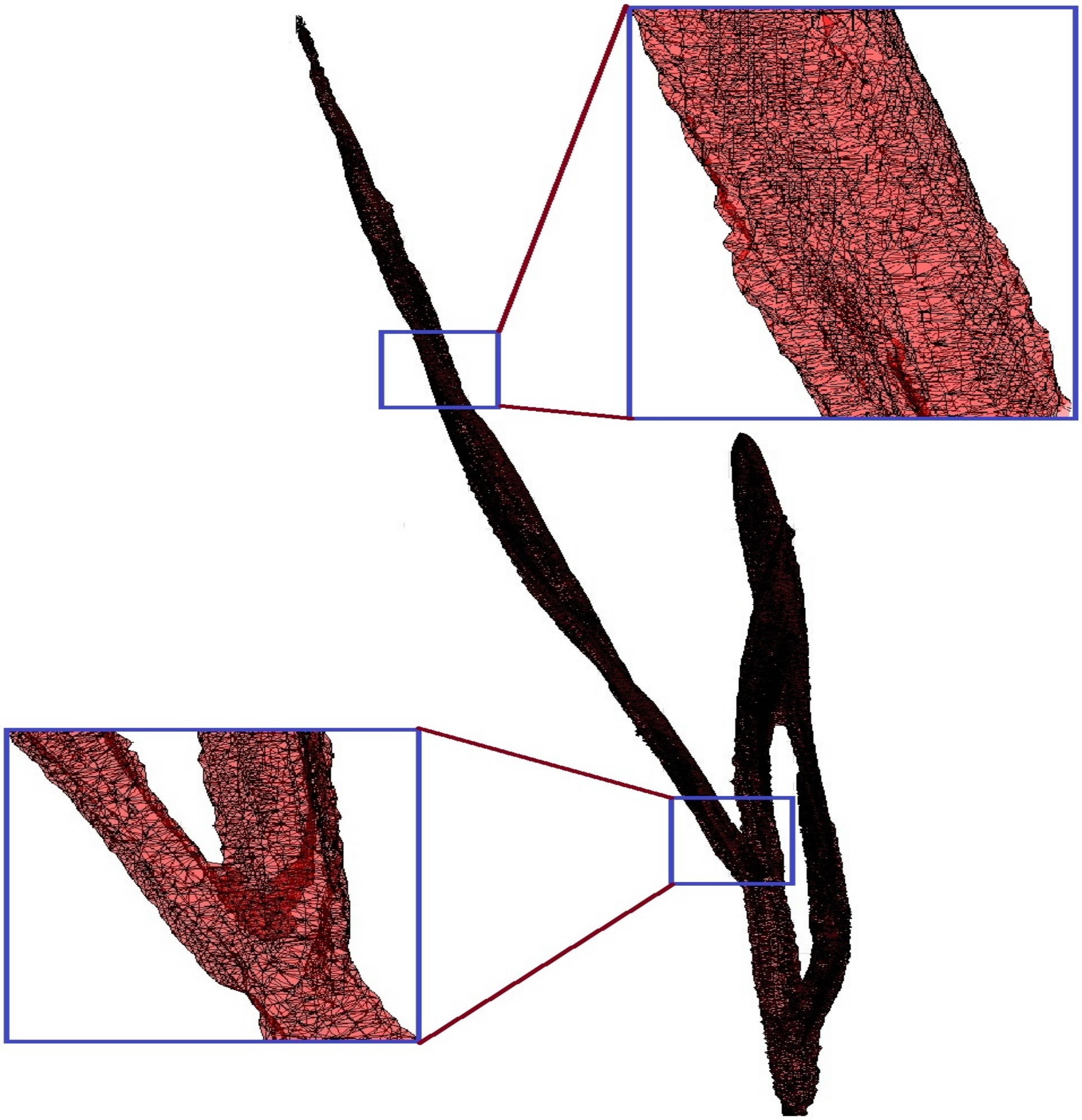


Figure 4.14: Triangulated Barley plant data

Figures 4.17c and 4.17d show the surface area and volume growth rates for the barley plant. The growth rates (slopes of the growth rate lines) are printed as text items in the upper left corner of each graph and show that the surface area and volume growth rates for the two plants are roughly the same. However, as the growth rate of Barley plant exhibits a highly non-linear pattern, fitting a straight line to compute the actual growth rate may not be appropriate. We believe instead that a polynomial curve fitting scheme might be a better, with local growth rates being the slopes of the tangents on this curve.

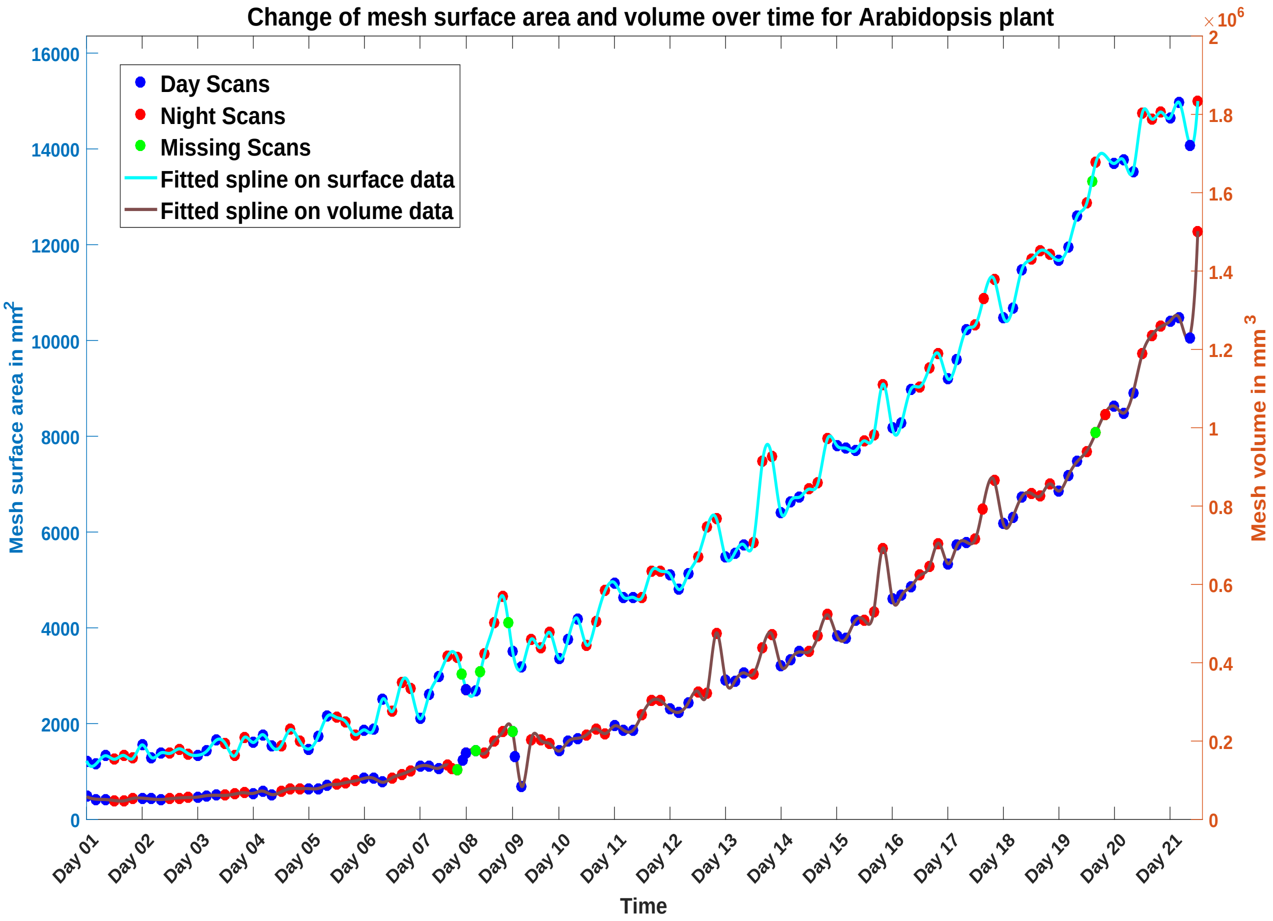


Figure 4.15: Diurnal growth pattern of mesh surface area and volume for the Arabidopsis plant. The red dots represent night time scans, the blue dots represent day time scans and the four green dots represent missing scan data. A spline is fitted to both surface and volume scan data (shown in different colours). The *y*-axis in the left and right hand side represents the range of surface area and volume data respectively.

4.6 Limitations

In this section we present some experimental results to demonstrate the limitations of the proposed system and the room for future research directions. We have presented the results of growth analysis of a monocot Arabidopsis and a dicot Barley plant. The experimental results are shown to be accurate enough to capture the well known diurnal growth pattern of the plants. However, the proposed system and the algorithms have certain limitations in analyzing more complex plants. We have experimented with a conifer and a bean plant. Due to the fine needle structures, an accurate reconstruction of conifer plant is very challenging. And the bean plant has many leaves that occlude each other, which make the pairwise registration extremely difficult. We show the results in next subsections and discuss about possible ways to solve the problems.

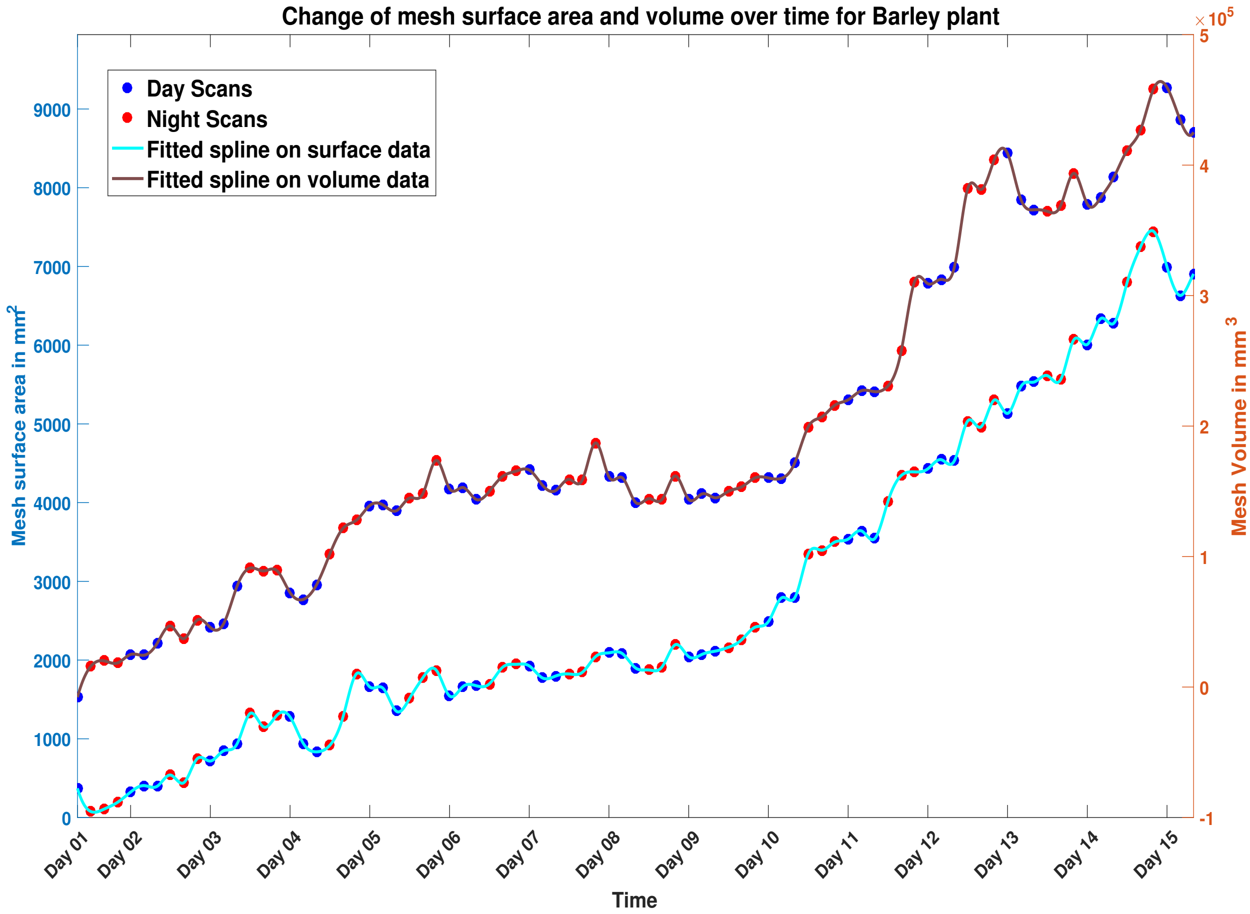
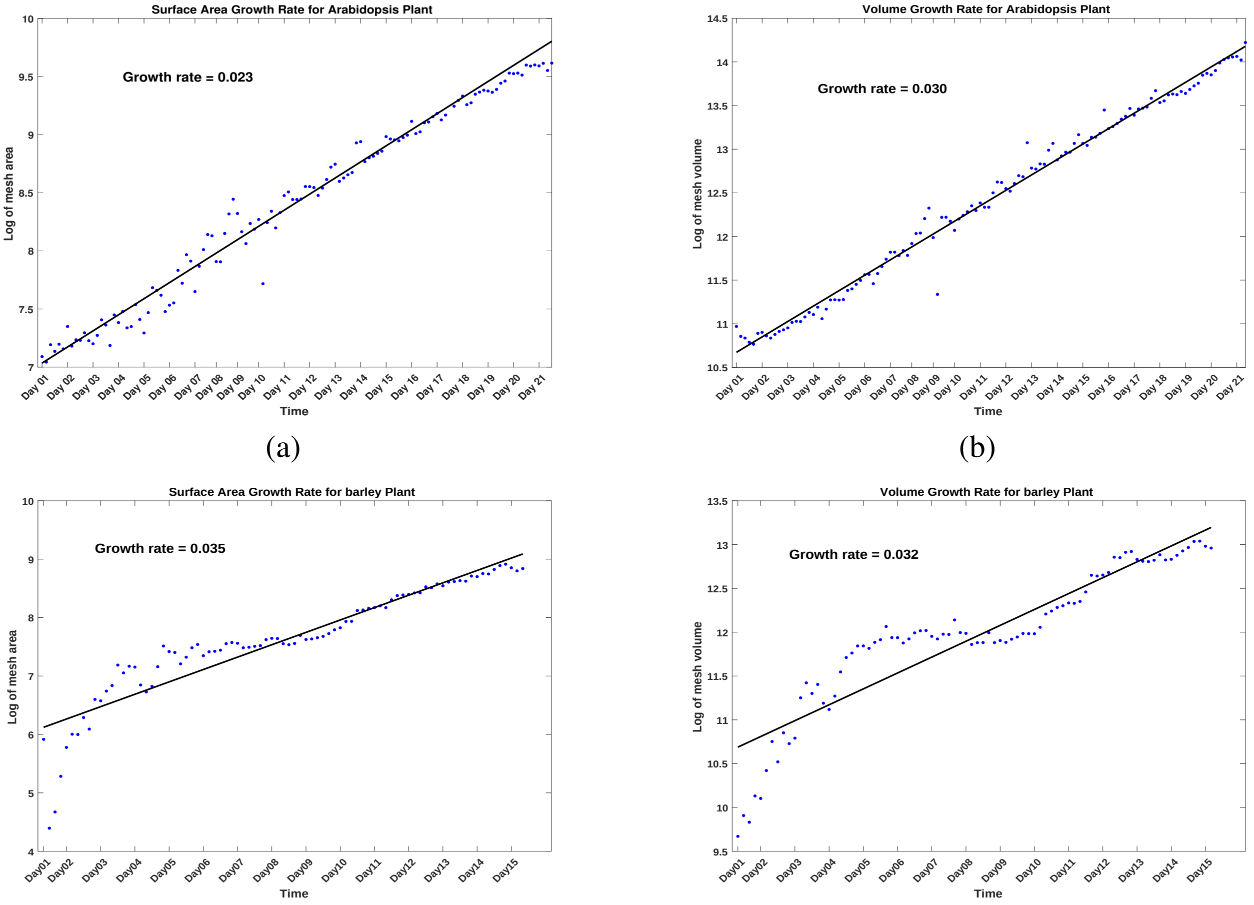


Figure 4.16: Diurnal growth pattern of mesh surface area and volume for the barley plant. The red dots represent night time scans while the blue dots represent day time scan data. A spline is fitted to both surface and volume scan data (shown in different colours). The y-axis in the left and right hand side represents the range of surface area and volume data respectively.

4.6.1 Conifer Experiment

Although growth analysis of plants, like the *Arabidopsis* plant (which is a model plant in biology), has extensively been studied, the growth of conifer plants (*Pinophyta*) have not been explored much. Usually the conifer plants have lifetime over 50 years, and it is practically impossible to experiment for the lifetime of the plant. We have performed a one month experiment with a conifer plant for a short-term study of its growth pattern.

Accurate reconstruction of the conifer is challenging due to the fine structures of its needles. For example, Figure 4.18 shows the reconstruction of 12 view of a conifer tree with a single cut-out shows one part of its surface at higher resolution. We can see that the range data does not capture the needle structure adequately. Ideally, each needle of the conifer should be clearly and completely visible in the reconstructed point cloud but this is not the case. The initial growth pattern of conifer plant is shown in Figure 4.19. The growth rate calculated



(a) (b)

(c) (d)

Figure 4.17: (a) and (c): the surface growth rate lines and (c) and (d): the volume growth rate lines for the Arabidopsis and barley plants.

from this pattern is a flat horizontal line (Figure 4.20) with a slope of 0.0 (effectively, plant growth cannot be captured at the sampling rate of twice a day we are currently using). Lower sampling rates, for example, once per week might capture a growth pattern but certainly not any nightly diurnal growth patterns. One of the reasons to measure this plant's growth was to see if we could observe a diurnal growth pattern (it is currently unknown if one exists). Obviously, this is not possible with our current set-up. It is unknown if the needles shrink and expand from night to day (and if they do, can we capture this information?). Perhaps, some simple opening/closing morphological operations to close gaps in the range data for the needles would be helpful here.

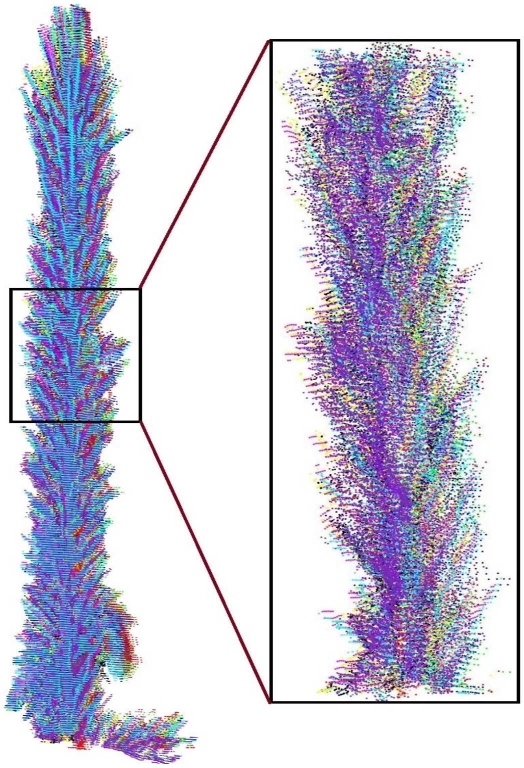


Figure 4.18: Reconstruction of a conifer plant (different colors indicate different scans).

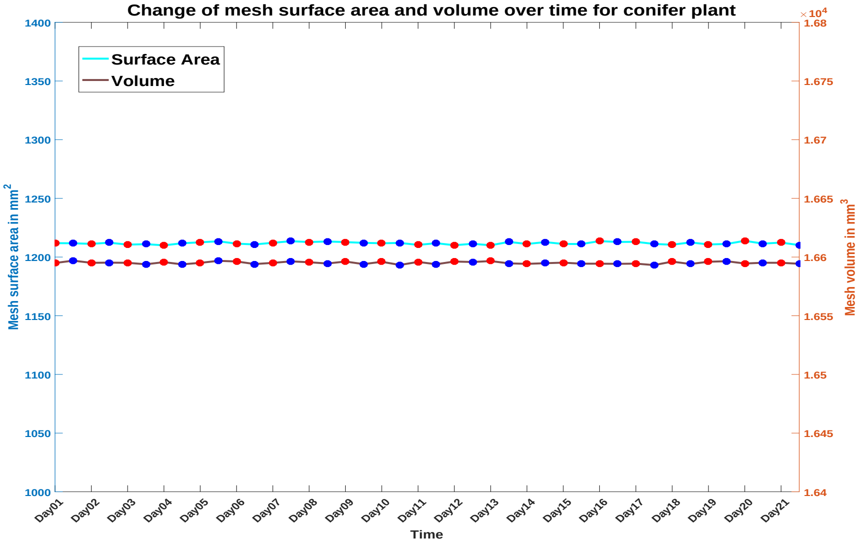


Figure 4.19: Growth data for the conifer plant. The conifer as scanned at 14:00 in the afternoon (light) and at 2:00 in the night (dark).

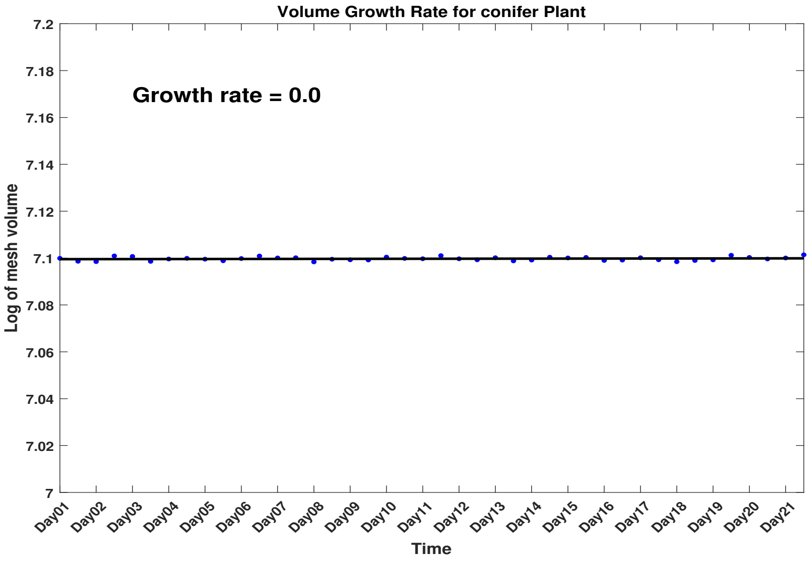


Figure 4.20: Volume growth rate for the conifer plant

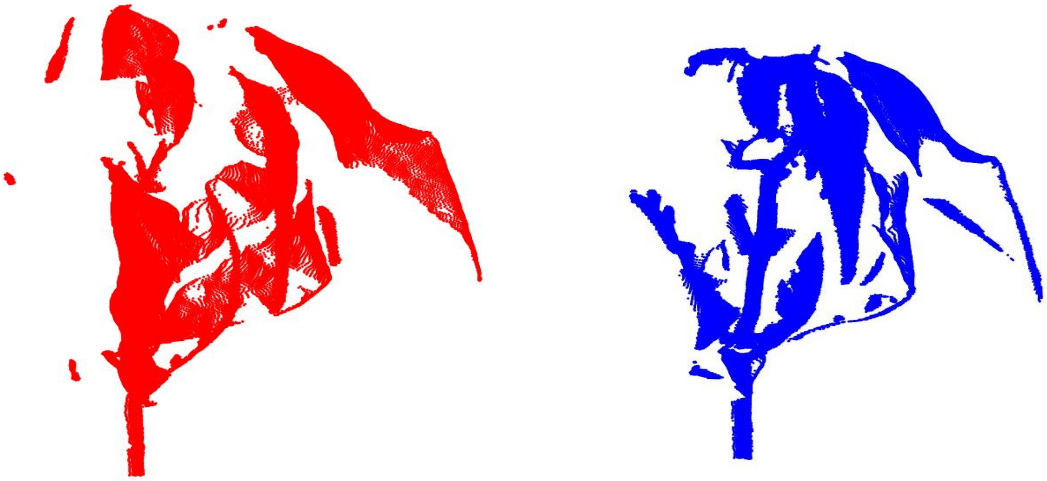


Figure 4.21: Two adjacent views of a bean plant

4.6.2 Bean plant experiment

Next, we have performed an experiment with a bean plant (*Phaseolus vulgaris*). The plant has many leaves which occlude each other. We have taken 12 scans around the plant in 30° intervals. Two adjacent views are shown in Figure 4.21. The reconstruction result is shown in

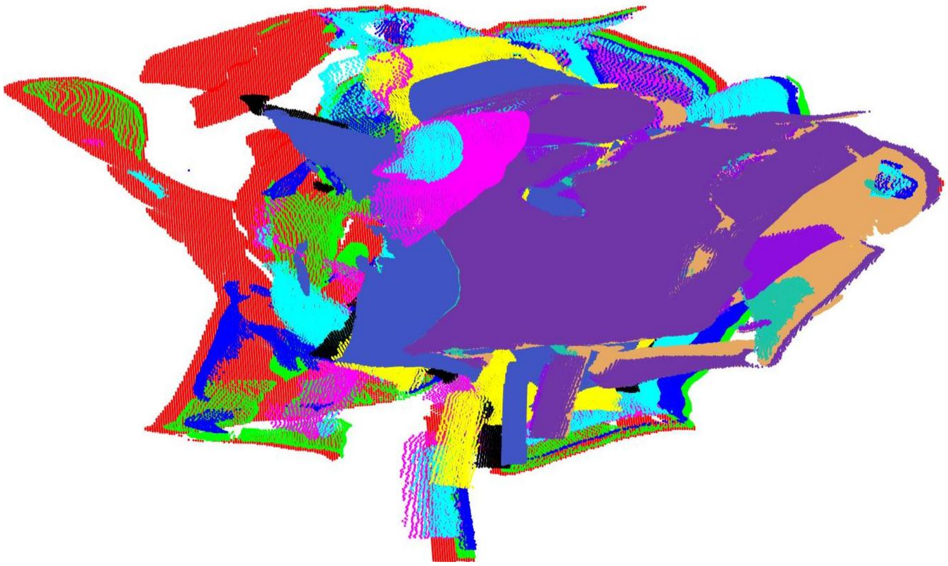


Figure 4.22: Reconstruction result of bean plant (different colours indicate different scans)

Figure 4.22.

Clearly, the reconstruction result is extremely poor. We believe that too much occlusion is the reason of poor performance. As can be seen from Figure 4.21, there is very little little overlap of common structure between two views. This makes the pairwise registration to be extremely difficult. If the pairwise registration fails to perform well, the alignment result will also be poor as the alignment exploits the pairwise registration results. This is an extremely challenging problem where the state-of-the-art algorithms fail. An idea to solve the problem might be to incorporate prior information of the structures in the pairwise registration. We have not investigated these issues, and we leave as future research. However, taking more scans (instead of 12) might also be a good idea to obtain good registration results.

Chapter 5

Junction Based Correspondence

Correspondence Estimation refers to the matching of interest points in stereo pairs. Although Gaussian Mixture Model (GMM) based point cloud based registration algorithms work well on wide class of problems, the rough initial alignment is extremely crucial to get reasonably good results. We show that if the scans are not approximately aligned, beyond a certain threshold the registration results drastically deteriorate. The motivation of this chapter is to address the problem of automatic correspondence estimation of the plant point cloud data, that can be used a preprocessor of GMM based registration without any knowledge of rough initial alignment. We show how to replace the initial alignment assumption of the transformation parameters in a fully automatic way (refer to section 4.3).

This work in this chapter resulted in a journal paper as follows:

- A. Chaudhury, M. Brophy, and J.L. Barron, *"Junction Based Correspondence Estimation of Plant Point Cloud Data using Subgraph Matching"*, IEEE Geoscience and Remote Sensing Letters, Vol. 13, No. 8, pp. 1119-1123, August 2016.

5.1 Introduction

Processing botanic tree point cloud data is of interest in many application areas. Apart from being an integral part of studying plant growth in biological applications, vegetation analysis is an active research area in in remote sensing and terrestrial applications [78]. With the advancements of remote sensing technologies and near-infrared laser scanners, 3D automatic non-invasive analysis of growing plants is becoming possible. Recently, a new body of literature has appeared on point cloud imaging based plant analysis for terrestrial applications ([20], [21], [169]).

Many of these applications require the construction of a 3D model from multiple scans and much work has already been done on this area. However, modelling the 3D structure of rigid objects is less demanding than complex plant structures. Problems like non-rigidity, complex morphology and thin stem structures, etc. make the problem much more complicated. Point to point matching (*Pairwise registration*) is a basic task in aligning two datasets and building the 3D model. Very often in the computer vision literature, *feature points* are used to match two images. These points are usually points where there is a sharp discontinuity in some local feature. However, in the case of a 3D plant point cloud, this kind of generalized feature can result in ambiguities because of the complex morphology of the plant. One motivation of the proposed work is to address this problem. We argue that *junction* points can be more effective than traditional features for registering plant like structures (thin self repetitive structures having branches). We show the efficacy of our method by matching corresponding junction points by a sub-graph matching optimization technique.

This work is an improvement of the work proposed by Bucksch *et al.* [20]. We present three major contributions:

- **First**, we consider the case of pairwise registration when the initial alignment is not known in advance between two views of the plant.

- **Second**, we propose an efficient junction detection algorithm for matching of feature points.

- **Third**, we formulate the correspondence estimation as a sub-graph matching optimization technique and demonstrate better results than the state of the art.

Temporarily, we transform the 3D problem into 2D by performing the appropriate 3D coordinate transformations to the neighbourhood of each 3D point. Our proposed method is two step. First, a statistical dip test of multi-modality is performed to detect non-linearity of the local structure. Then each branch is approximated by sequential RANSAC line fitting and an Euclidean clustering technique. The straight line parameters of each branch are extracted using Total Least Squares (TLS) estimation. Finally, the straight line equations are solved to determine if they intersect in the local neighbourhood. Such junction points are good candidates for subsequent correspondence algorithms. Using these detected junction points, we formulate a correspondence algorithm as a sub-graph matching problem and show that without using traditional descriptor similarity based matching, good correspondences can be obtained by simply considering geodesic distances among graph nodes. Experiments on synthetic and real (*Arabidopsis* plant) data show that the proposed method outperforms state of the art.

In the next section we describe the rotations and translations needed to convert the 3D problem into an equivalent 2D problem. Then we discuss the dip test for multi-modality. After demonstrating RANSAC fitting and TLS approximations, we demonstrate sub-graph matching for correspondence estimation. Finally experimental results and analyses for synthetic and real datasets are shown.

5.2 Coordinate Transformation

We use a *kd*-tree algorithm to compute the nearest neighbour points within a certain radius (set at 2.0*mm*). Given such points in a local neighbourhood about some 3D point, we transform the data so that the surface normal of the plane fitting the data is a line-of-sight vector (0,0,1).

More specifically, we compute the center of mass (x_{cm}, y_{cm}, z_{cm}) of the neighbourhood 3D points. To reformulate as a 2D problem, we perform the following steps: translate the origin to the center of mass by $-(x_{cm}, y_{cm}, z_{cm})$, rotate about the x-axis onto the $x-z$ plane by some Euler angle α, rotate about the y-axis onto the longitudinal axis (0,0,1) by some Euler angle β

and finally transform the origin back to the previous location by (x_{cm}, y_{cm}, z_{cm}). The detailed calculations are shown below.

5.2.1 Coordinate Transformation: 3D to 2D

We show the detailed calculations of converting a 3D problem into 2D. We use the *kd*-tree algorithm to compute nearest neighbour points within a certain radius. First, a plane of the form $ax + by + cz + d = 0$ is fitted to the neighbourhood data and the parameters are obtained. Consider 3 points on a planar surface: $P_1(x_1, y_1, z_1)$, $P_2(x_2, y_2, z_2)$, $P_3(x_3, y_3, z_3)$. Compute the vectors $\vec{V_1}$ and $\vec{V_2}$ (see Figure 5.1) as:

$$\vec{V_1} = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix}, \vec{V_2} = \begin{bmatrix} x_3 - x_1 \\ y_3 - y_1 \\ z_3 - z_1 \end{bmatrix}$$

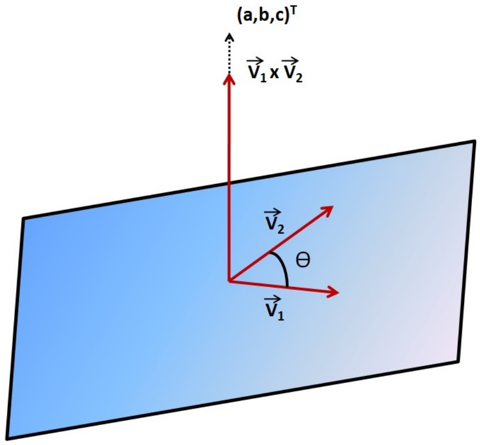


Figure 5.1: Planar vector orientations

Then $\vec{V_1} \times \vec{V_2}$ is the normal to the surface $ax + by + cz + 1 = 0$. That is, $\vec{V_1} \times \vec{V_2}$ and $(a, b, c)^T$ are in the same direction.

We aim to nullify the effect of z-coordinates, which require following steps. First we translate the origin to the center of mass (CM) $(-x_m, -y_m, -z_m)$ so that the origin coincides with the CM. We use homogeneous coordinate (4D) to have all transformations specified matrix multiplications. In 3D heterogeneous coordinates, translation is specified as vector addition but in 4D homogeneous coordinates it is specified by matrix multiplication, as are all other operations, allowing matrix concatenation of all matrices into one matrix. The 4D homogeneous transformation matrix has the following form:

$$T(T_x, T_y, T_z) = \begin{pmatrix} 1 & 0 & 0 & T_x \\ 1 & 0 & 0 & T_y \\ 1 & 0 & 0 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.1}$$

Thus $T(-x_m, -y_m, -z_m)$ does the translation to the center of mass (the new origin). Next we project the rotation axis onto the z-axis. This requires two steps: rotate by some unknown α angle about x-axis so that the vector $\hat{u}$ is in the xz-plane, and then rotate by some unknown β angle about the y-axis to bring vector $\hat{u}$ onto the z-axis. We show how to calculate α and β in next 2 subsections. Finally we re-translate back the origin to the previous location by the inverse translation $T(x_m, y_m, z_m)$.

Let's consider rotation about the z-axis. In that case, $\vec{V}$ is the rotation axis with endpoints (x_1, y_1, z_1) and (x_2, y_2, z_2). We rotate about $\vec{V}$ (see Figure 5.2), given by:

$$\vec{V} = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

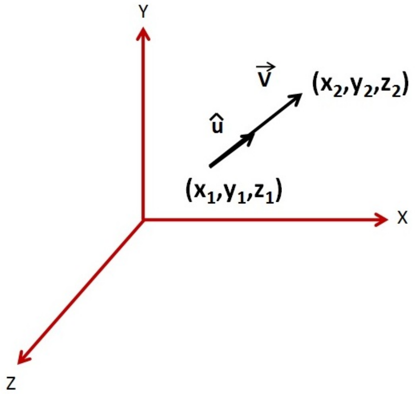


Figure 5.2: Rotate about V

In this case, $\hat{u} = \frac{\vec{V}}{||\vec{V}||_2} = (a, b, c)$ is the unit vector in $\vec{V}$'s direction. The direction cosines of $\vec{V}$ are given by:

$$a = \frac{x}{||\vec{V}||_2}, \quad b = \frac{y}{||\vec{V}||_2} \text{ and } c = \frac{z}{||\vec{V}||_2}.$$

We use the following convention: $\hat{u}$ is the normal vector, and $\vec{u}$ is the unnormalized vector (of the projection of $\hat{u}$ onto $y - z$ plane.

Rotate $\hat{u}$ into the xz-plane

Let α is the rotation angle between the projection of $\vec{u}$ in the yz-plane and the positive z-axis and $\vec{u}'$ is the projection of $\hat{u}$ in the yz-plane (Figure 5.3). That is, $\hat{u} = (a, b, c)^T \implies \vec{u}' = (0, b, c)^T$. Then the angle α can be obtained simply from the equation,

$$\vec{u}' \cdot \hat{k} = ||\vec{u}'||_2 ||\hat{k}||_2 \cos\alpha.$$

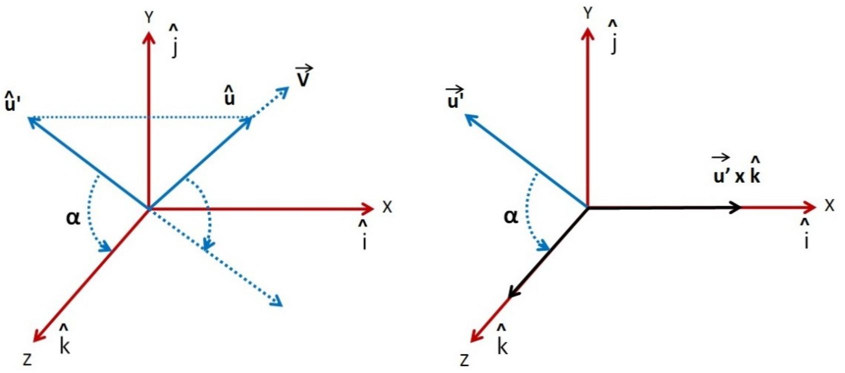


Figure 5.3: Rotate $\hat{u}$ about xz-plane. Left: Project $\hat{u}$ onto the $y - z$ plane as $\hat{u}'$. Right: $\hat{u}'$ is rotated by α about the x axis onto the $\hat{k}$ axis.

Let $\hat{k} = (0, 0, 1)$ is the unit vector in the z-direction, i.e. $||\hat{k}||_2 = 1$. Then

$$\begin{aligned} ||\vec{u}'||_2 = \sqrt{\vec{u}' \cdot \vec{u}'} &= \sqrt{(0, b, c) \cdot (0, b, c)} = \sqrt{b^2 + c^2} \\ &\implies \vec{u}' \cdot \hat{k} = (0, b, c) \cdot (0, 0, 1) = c. \end{aligned}$$

Thus,

$$c = \sqrt{b^2 + c^2} \cdot 1 \cdot \cos\alpha \implies \cos\alpha = \frac{c}{\sqrt{b^2 + c^2}}.$$

The vector product can also be used to compute $\sin\alpha$. Note that $\vec{u}' \times \hat{k}$ is a vector in x's direction, i.e., $\hat{\imath}$. Then

$$\vec{u}' \times \hat{k} = \hat{\imath}||\vec{u}'||_2 \, ||\hat{k}||_2 \sin\alpha = \hat{\imath}\sqrt{b^2 + c^2} \sin\alpha.$$

and

$$\vec{u}' \times \hat{k} = \begin{vmatrix} \hat{\imath} & \hat{\jmath} & \hat{k} \\ 0 & b & c \\ 0 & 1 & 1 \end{vmatrix} = b\hat{\imath}.$$

Then $b\hat{\imath} = b\hat{\imath}\sqrt{b^2 + c^2} \sin\alpha$, or $\sin\alpha = \frac{b}{\sqrt{b^2+c^2}}$.

Given $\sin\alpha$ and $\cos\alpha$, we can specify the 4D homogeneous rotation matrix for rotation about the x-axis as:

$$R_X(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{c}{\sqrt{b^2+c^2}} & \frac{-b}{\sqrt{b^2+c^2}} & 0 \\ 0 & \frac{b}{\sqrt{b^2+c^2}} & \frac{c}{\sqrt{b^2+c^2}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.2}$$

This matrix rotates $\hat{u}$ onto the xz-plane.

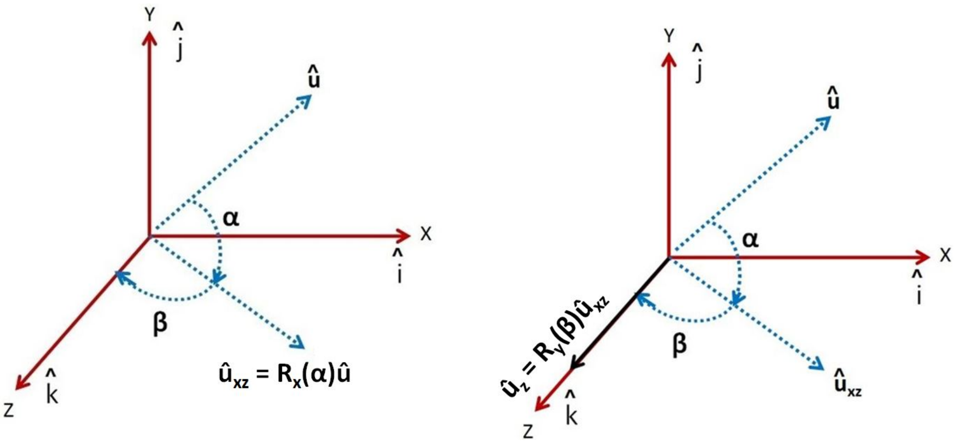


Figure 5.4: Aligning $\hat{u}$ along z-axis. Left: $\hat{u}$ is rotated by α about the x axis onto the $x-z$ plane as $\hat{u}_{xz}$. Right: $\hat{u}_{xz}$ is rotated by β about the y-axis onto the $\hat{k}$ axis as $\hat{u}_z$.

Align $\hat{u}_{xz}$ along z-axis

As shown in Figure 5.4, we need to compute $\sin\beta$ and $\cos\beta$ in this case.
Using the dot product we can write:

$$\begin{aligned}\hat{u}_{xz}\cdot\hat{k} &= \overbrace{||\hat{u}_{xz}||_2}^{=1}\ \overbrace{||\hat{k}||_2}^{=1}\cos\beta\\ = (a,0,\ \sqrt{b^2+c^2})\cdot(0,0,1)^T &= \sqrt{b^2+c^2}\\ \implies \cos\beta &= \sqrt{b^2+c^2}.\end{aligned}$$

Also, using the vector product, $\hat{k}\times\hat{u}_{xz}$ is a vector in the direction of the y-axis, thus resulting:

$$\hat{u}_{xz}\times\hat{k} = \hat{j}\overbrace{||\hat{u}_{xz}||_2}^{=1}\ \overbrace{||\hat{k}||_2}^{=1}\sin\beta$$

and

$$\begin{vmatrix}\hat{i} & \hat{j} & \hat{k}\\ a & 0 & \sqrt{b^2+c^2}\\ 0 & 0 & 1\end{vmatrix} = -a\hat{j}.$$

Thus, $-a\hat{j} = \hat{j}\sin\beta$, or $\sin\beta = -a$. Then the 4D homogeneous rotation matrix about the y-axis can be specified as:

$$R_Y(\beta) = \begin{pmatrix}\sqrt{b^2+c^2} & 0 & -a & 0\\ 0 & 1 & 0 & 0\\ a & 0 & \sqrt{b^2+c^2} & 0\\ 0 & 0 & 0 & 1\end{pmatrix} \tag{5.3}$$

which aligns $\hat{u}_{xz}$ with z-axis. Thus we apply the transformations:

$$R_y(\beta)R_x(\alpha)T(-x_m,-y_m,-z_m) \tag{5.4}$$

to all 3D points. If we wish to undo this transformation we could use:

$$T(x_m, y_m, z_m) R_x^T(\alpha) R_y^T(\beta), \tag{5.5}$$

where $R_x^T(\alpha) \equiv R_y^{-1}(\alpha)$ and $R_y^T(\beta) \equiv R_y^{-1}(\beta)$ because rotation matrices are unitary and orthogonal.

Next, a plane of the form $ax + by + cz + d = 0$ is fit to the neighbourhood data using Crammer's rule. The parameters $\vec{n} = (a, b, c)$ are the plane's surface normal. We checked the residual of this fit for many randomly chosen 3D neighbourhoods and the residual is always very small, conforming our local planarity assumption.

Since these transformations results in vertical surface normals we need only be concerned with the structure in the $x - y$ plane, i.e. the problem is now 2D.

5.3 DIP Test for Multi-Modality

Detection of multi-modality in numeric data is a well known problem in statistics. A probability density function having more than one mode is denoted as a multi-modal distribution. Hartigan *et al.* [62] proposed a dip test for unimodality by maximizing the difference between the empirical distribution function and the unimodal distribution function. In the case of a unimodal distribution, the value for the dip should asymptotically approach 0, while for the multi-modal case it should yield a positive floating point number. Inspired by the work of Zhao *et al.* [165], we use the dip test for detecting non-linearity in the data. Points having non-linear local neighbourhood are potential candidates for a junction point. The idea is to perform the dip test for a local neighbourhood of a point. If its a stem or a leaf, the data should be uniform and the distribution should only be unimodal. For a junction point likely due to a bifurcation, it should exhibit multi-modality. We use the dip value as a measure of multi-modality.

We perform this dip test along the x and y directions (note that as we have reduced the dimensionality from 3D to 2D the z-coordinates can be ignored) and obtain the maximum dip value. The neighbourhood is determined to be multi-modal if the dip value is over some threshold. However, the threshold value of dip is highly dependent on the data and should be tuned carefully. In our case, we empirically set it to 0.04 to obtain useful results.

We use the dip measurement for initial filtering of non-junction neighbourhood data. Note that nonlinearity and high dip values in local neighbourhood do not guarantee that those points are junction points. For some leaf and stem data, we notice that sometimes the data shows high dip values. Instead of relying blindly on dip test results, we do further processing, described in the next section.

5.4 RANdom SAmple Consensus (RANSAC) fitting and Total Least Squares (TLS) approximation

We consider at most three branches at an intersection point. Three branches may intersect at a single point (the red dot in Figure 5.6c) or at two different points (the red dots in Figure 5.6d). We also assume that the main stem will be thicker than the branches. We extract this thick stem

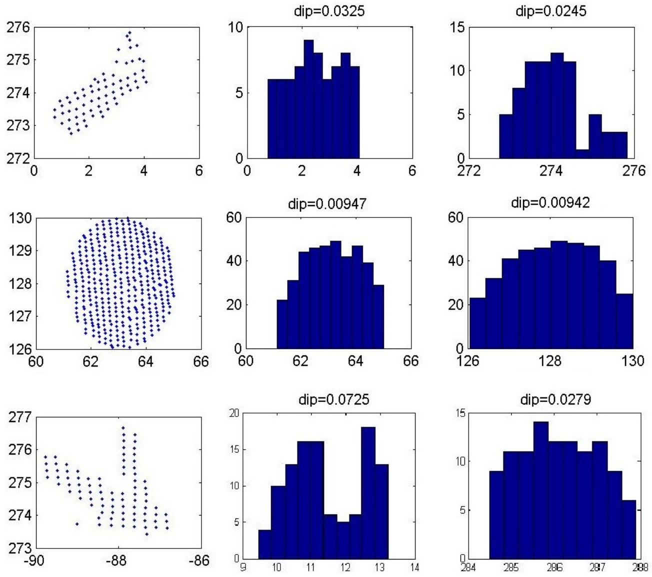


Figure 5.5: Distribution of data: column 1 are the neighbourhood point clouds under consideration, column 2 are the histograms of the x coordinates distribution and column 3 are the histograms of the y coordinate distribution for a single stem (first row), a leaf (second row) and a stem with 2 branches (last row). The later is potentially a junction point.

using RANSAC straight line fitting using a high distance threshold for inliers (empirically chosen as $0.8mm$ for our case). Sequential RANSAC is used to fit a straight line for every branch (set to $0.5mm$ for non-stem branches). However, there may be other points due to additional branches, a leaf or a noise event (or some combination of the three). After removing the RANSAC fitted main stem, we perform Euclidean clustering on the rest of the data and choose the biggest connected component(s) to extract the sub-branches. Two set of points, $\mathcal{X}_i = \{p_i \in \mathcal{P}\}$ and $\mathcal{X}_j = \{p_j \in \mathcal{P}\}$ form two different clusters, if the following condition holds: $\mathbf{min}\|p_i - p_j\|_2 \geq \tau$, where τ is the distance threshold (set to 1.0). The branches may be straight or curved, but by using RANSAC we can estimate the principal direction of the branch[143]. A criterion is imposed to estimate a broken branch shape (due to occlusion): we merge two branches if they are spatially close to each other and have the roughly same direction.

After estimating the points for each branch, we use TLS to approximate the straight line represented by a set of points in a branch and extract the parameters by minimizing $E = \sum_{i=1}^{n}(ax_i + by_i + c)^2$ and check if the intersection point is contained in the local neighbourhood or not. Note that the obtained intersection point is 2D, so we can apply the inverse coordinate transformation determined earlier to compute the actual 3D point. Finally we perform non-maximal suppression based on highest dip value to reduce the number of points. The distance threshold for non-maximal suppression is chosen as $4mm$ (the same threshold is used also in [96]). The procedure is described in Algorithm 2.

Algorithm 2 Junction Point Detection

```
Require: Neighbourhood point set $\mathcal{P} = \{x_1, x_2, \cdots, x_n\}$
Translate $\mathcal{P}$ to the center of mass
Rotate about $x$-axis
Rotate about $y$-axis
Re-translate $\mathcal{P}$ back to the previous origin
Compute $dip$ along $x$ and $y$ axes: $dip_x$, $dip_y$
if max($dip_x$, $dip_y$)$> \tau$ then
    Fit RANSAC to $\mathcal{P}$ & store the point set into $\mathcal{M}_1$. Remove the points from $\mathcal{P}$, resulting
    in $\mathcal{P}_1 = \mathcal{P} - \mathcal{M}_1$
    if $|\mathcal{P}_1| > n_t$ {sufficient number of points} then
        Extract the two largest Euclidean clusters $\mathcal{O}_1$, $\mathcal{O}_2$ of $\mathcal{P}_1$ {Locate sub-branch(s)}
        if $|\mathcal{O}_1|, |\mathcal{O}_2| > n_k$ {sufficient number of points} then
            Fit RANSAC to $\mathcal{O}_1$ (and $\mathcal{O}_2$) & store the point set in $\mathcal{M}_2$ (and $\mathcal{M}_3$)
            Apply TLS to $\mathcal{M}_1$, $\mathcal{M}_2$ (and $\mathcal{M}_3$) and obtain the straight line parameters
            Solve the straight line equations using the computed parameters for $\mathcal{M}_1$, $\mathcal{M}_2$ (and
            $\mathcal{M}_1$, $\mathcal{M}_3$). Store the computed point(s) in $J_1$ (and $J_2$)
            Check if the corresponding 3D coordinates of $J_1$ (and $J_2$) are contained in $\mathcal{P}$. If $J_1$
            and $J_2$ are close enough, merge them into $J = (J_1 + J_2)/2$
            return $J$ (or $J_1$, $J_2$) {One or two junction points}
        end if
    end if
else
    return NULL {No junction point}
end if
```

5.5 Correspondence Matching

The detected junction points from the last phase are potential candidates for correspondences and can be used as features points for matching. For raw 3D point cloud data, local surface normals, neighbourhood information, etc. are typically used for encoding the local structure and points are matched based on the descriptor similarities. We observed empirically that this idea fails for plant data because the thin structures do not allow for good local surface normal calculations and because of deformations, the local structure can change abruptly in adjacent images. This problem was also addressed by Duchenne *et al.* [43] where triplets of feature points were used to compute mutual angles. However, they assume that the coordinate axes are known in advance. Also, smart selection of triangles is an important factor in computing good correspondences. There are many other properties that can be considered for mutual matching, but we show that simple geodesic distance can be used efficiently in finding correct correspondence, emphasizing the simplicity and effectiveness of the proposed algorithm.

5.5.1 Graph Formation

First, we triangulate the data using Delaunay triangulation in 3D (note that we converted the problem temporarily to 2D just for detection of junction points). Using the vertex information from triangulation, we construct a graph connecting all the points. To handle the cases of missing or occluded data, we connect the points to the nearest triangle vertex so that all the points are included in a single graph. Then, for each junction point, we apply Dijkstra's shortest path algorithm to compute the geodesic distance to all other junction points. The same procedure is followed for the second point cloud as well. Then we exploit pairwise distances to be the criteria for matching. This kind of local descriptor-less approach was also proposed by Leordeanu *et al.* [81]. However, they assumed some prior knowledge of the object categories. In our case we do not have any prior knowledge or make any assumptions about the data.

5.5.2 Sub-graph Matching Formulation

Given all the pairwise distances of all junction points, we formulate our correspondence matching problem as a sub-graph matching problem. Consider two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Each junction point is considered to be a node of the graph. Each node stores the geodesic distances to all other nodes. In the end this yields a set of edges. Compatibility of two nodes in G_1 and G_2 are defined as a closest distance match. For example, let's suppose two graphs G_1 and G_2 have n_1 and n_2 nodes. Each node V_{1_i} in G_1 stores all distances to all other nodes. We denote this is as the set of attributes of node V_{1_i}: $\mathcal{D}_{v_{1_i}} = \{d_{v_{1_i}v_{1_j}}\}, \forall j \in n_1$. Similarly in G_2, the set of attributes of node V_{2_i} is defined as $\mathcal{D}_{v_{2_i}} = \{d_{v_{2_i}v_{2_k}}\}, \forall k \in n_2$. The compatibility of two nodes, V_{1_i} and V_{2_i}, are formulated as sum of the squares of the difference of nearest distances, multiplied by the number of matches. Suppose G_1 and G_2 contain 5 and 7 nodes respectively. Let the attributes of a node V_{1_i} contains the following distances: $\{d_1, d_2, d_3, d_4\}$ (ignoring self distance). Similarly, V_{2_i} contains the distances $\{d'_1, d'_2, d'_3, d'_4, d'_5, d'_6\}$. We use a threshold ϵ (= 0.2) for the match of two distances. Suppose there are 3 distance matches given by: $d_1 \sim d'_4$, $d_3 \sim d'_2$, $d_4 \sim d'_1$. Then the affinity of the two vertices is computed as:

$$\mathcal{A}_{v_{1_i} v_{2_i}} = 3 * [(d_1 - d'_4)^2 + (d_3 - d'_2)^2 + (d_4 - d'_1)^2] \tag{5.6}$$

The logic for using this kind of distance matching is that any outlier is likely to be eliminated by a lower number of matches. On the other hand, compatible points will only have the maximum number of distance matches.

5.5.3 Matching of Nodes

Using the compatibility of two vertices, we obtain the initial node correspondence by using the Hungarian algorithm. The outliers are likely to get rejected by unmatched distance attributes. However, there still may be non-optimal matches of the vertices. We follow the approach of attribute graph matching proposed by Cour *et al.* [36]. Given two graphs, $G_1 = (V_1, E_1, A_1)$ and $G_2 = (V_2, E_2, A_2)$, where each edge $e = V_i V_j \in E$ has an attribute A_{ij}. The objective is to find $\mathcal{N}$ pairs of correspondences (V_i, V_j) where $V_i \in V_1$ and $V_j \in V_2$. The affinity A_{ij} (Equation (5.6)) defines the quality of the match between nodes V_i and V'_i. Denoting the similarity function of pairwise affinity as $f(\cdot, \cdot)$, the matching score can be computed as:

$$\lambda(\mathcal{N}) = \sum_{ii', jj' \in \mathcal{N}} f(\mathcal{A}_{ij}, \mathcal{A}'_{i'j'}) \tag{5.7}$$

Representing $\mathcal{N}$ as a binary vector x so that $x(ii') = 1$ if $ii' \in \mathcal{N}$, the above equation can be written as:

$$max\, \lambda(x) = x^T W x, \tag{5.8}$$

where $W_{ii', jj'} = f(\mathcal{A}_{ij}, \mathcal{A}'_{i'j'})$. The optimal solution of the above equation is given by:

$$x^* = argmax(x^T W x). \tag{5.9}$$

The permutation matrix provides the correspondence among the vertices (or the junction points in our case). Finally the outliers (or wrong matches) are pruned out using RANSAC.

5.6 Experimental Results

First, we tested our method on synthetic dataset. We have used *Vascusynth* [144] to generate artificial artery data, used their ground truth for junction points and have obtained 100% correct matches.

Second, we show the result of junction point detection (red dots) on a variety of Arabidopsis datasets in Figure 5.6. The dataset is extremely challenging due to occlusion, missing data and deformations. Results show the robustness of the algorithm in detecting junction points correctly. We have validated our junction detection algorithm by showing efficient correspondence matching based on these feature points. Figure 5.7 shows two examples from the dataset. The left image shows matching on dataset having local deformations and missing data, whereas the right image shows the matching when the dataset is rotated 180° relative to each other. The few bad matches (red lines) are eliminated by RANSAC in both cases.

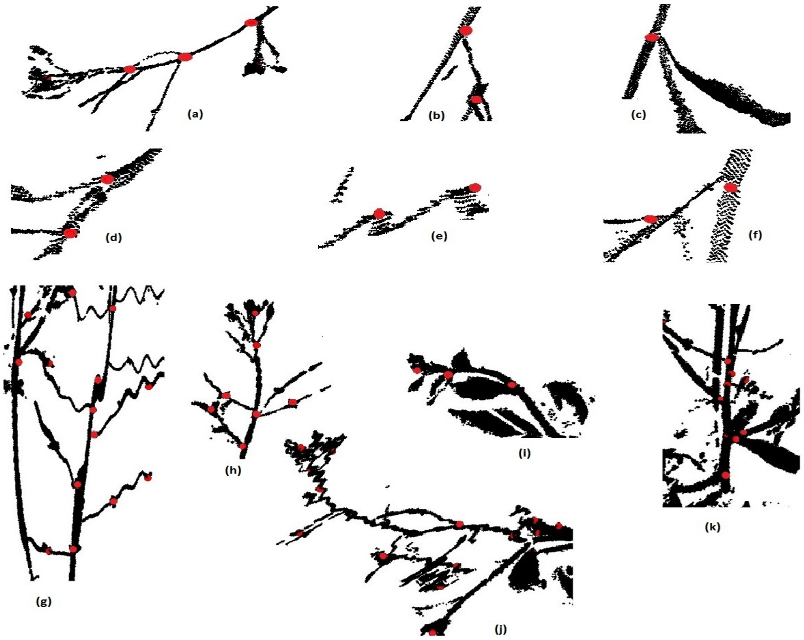


Figure 5.6: Examples of detected junction points (red dots) on the real Arabidopsis plant data

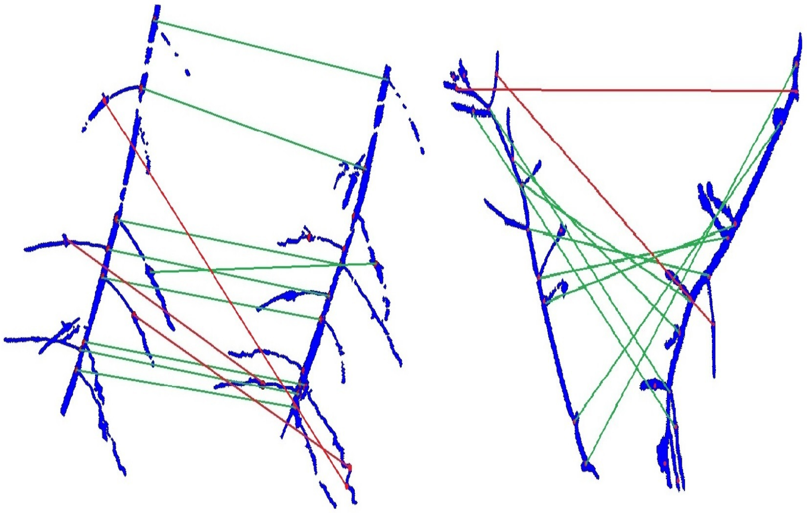


Figure 5.7: Two examples of correspondence matching (green lines are good, red lines are bad matches which are eliminated by RANSAC)

We compare our results with 3 state of the art algorithms: CPD [101], GMM [69] and SISI [37]. CPD and GMM are well established methods for global registration, and SISI is a state of the art algorithm for feature based correspondence matching, which has been shown to perform better than SIFT [90] and the spin image descriptor [72]. We observe that both CPD and GMM perform poorly in the cases of occlusion, deformation and missing data. We infer that since these algorithms are based on global optimization, they often can not handle ambiguities in local structure, whereas our algorithm performs well for these cases (the left image in Figure 5.7 is an example of such a case). It is also observed that for large rotation angles between the two images, the performance of CPD and GMM fails drastically. Figure 5.8 shows a few point to point correspondences obtained from the registration results for CPD and GMM for the rotated data in the right image in Figure 5.7. For GMM based algorithms we have experimented with different parameter settings and reported results for the best case. More specifically, we have used the default values of [69] in their publicly available code. For CPD [101], we have set the regularization weight $\lambda = 3$ and the width of Gaussian kernel (smoothness) $\beta = 2$. We have tried other different values of λ but but only recovered poorly warped registrations.

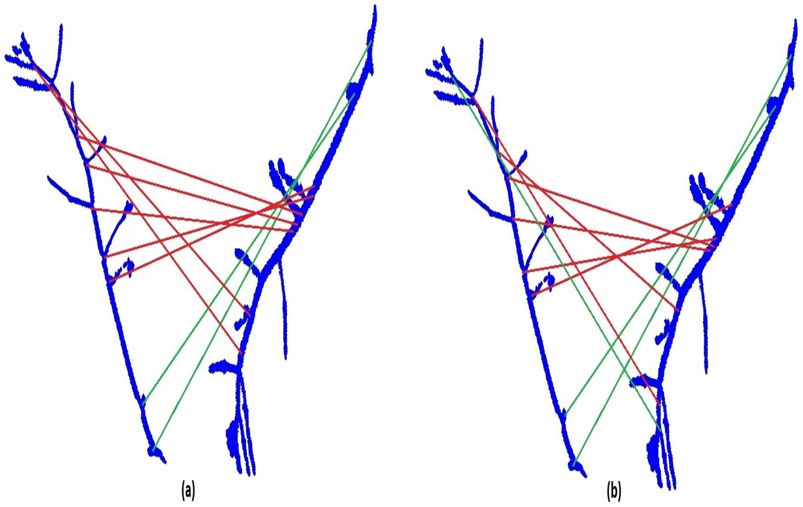


Figure 5.8: The point to point correspondence results obtained from (a) CPD and (b) GMM for large rotation angle. Good matches are shown in green lines and bad matches are in red lines.

CPD and GMM fail to perform well because the algorithms get stuck in local minima for large rotation angles. However, using our graph based matching method, the rotation angle does not affect performance. We present a quantitative comparison of CPD, GMM and our method for varying rotation angle in Figure 5.9. We perform similar analysis on increasing rotation angle as in [168].

We have also performed experiments using the SISI descriptor matching [37]. The algorithm failed to find meaningful correspondences between two views. We believe that the reason for this is the algorithm's dependence on local descriptors based on surface normals tend to be unreliable for our plant data. The precision-recall curves in Figure 5.10 show the superior performance of our method over CPD, GMM and SISI.

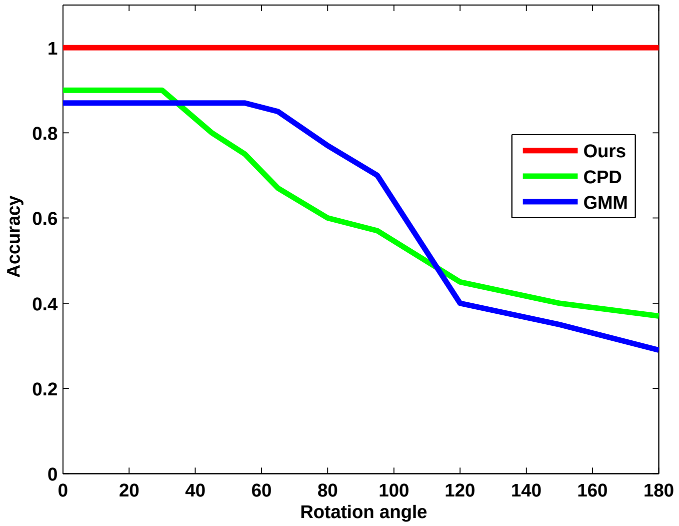


Figure 5.9: Effect of rotation angle: The rotation angle (in degrees) is plotted against (normalized) accuracy.

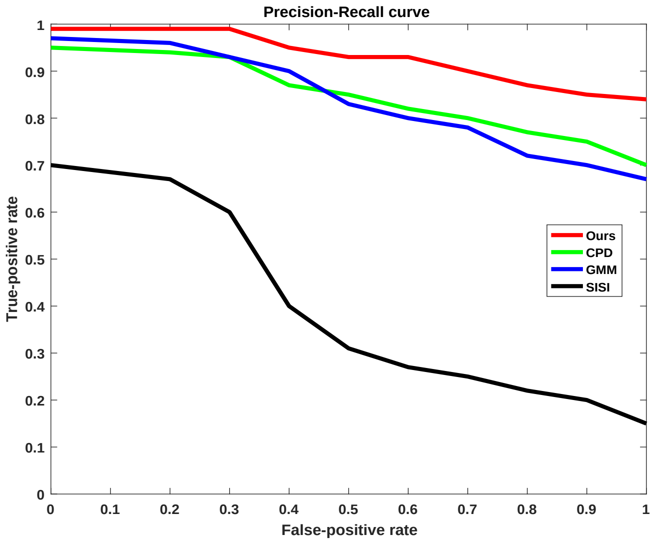


Figure 5.10: Precision-Recall curve comparing CPD, GMM, SISI and our method.

Chapter 6

Partially Occluded Leaf Matching

Recognition and analysis of leaves is an important parameter for growth analysis of plants. Leaves can appear in different shapes, sizes and varieties. When a plant is growing, leaves often occlude each other. This can make the growth analysis to be extremely complicated. While performing the range sensing a plant over time to measure its growth (as discussed in Chapter 4), we can store the depth and grayvalue images of specified plant leaves, and exploiting the depth and intensity discontinuities, individual leaves can be segmented from the matter around them. Now if we intend to track individual leaves over time, as done in the MSc theses of Zhao [163] and Yang [156], as the leaves grow they can potentially become occluded by other plant matter. The work described in this chapter will be very useful for this task. Another motivation is to address the problem of classifying the species of a plant from an occluded leaf. We have tested the algorithm on 3 public leaf databases, and achieved better results than the state-of-the-art.

This work in this chapter is under review of a journal as follows:

- A. Chaudhury and J.L. Barron, *“Partially Occluded Leaf Recognition via Subgraph Matching and Energy Optimization”* (in review)

6.1 Introduction

In last decade, there has been a body of work on plant species identification from leaf images (we refer to Chapter 2 for related work). This chapter presents an approach to classify partially occluded plant leaves from databases of full plant leaves. Although contour based 2D shape matching has been studied extensively in the last couple of decades, matching occluded leaves with full leaf databases is an open and little worked on problem. Classifying occluded plant leaves is even more challenging than full leaf matching because of large variations and complexity of leaf structures. In general, matching an occluded contour with all the full contours in a database is an NP-hard problem [135], so our algorithm is necessarily suboptimal.

The amount of occlusion is defined as the percentage of the contour that is missing (say, occluded by other leaves or objects). We assume that the leaves are segmented and the contour of the occluded leaves are known.[1] Currently, only one occlusion event per occluded leaf can be handled. The task is to find partial contours in full leaf contours that match the occluded contour. Although existing methods work well on standard shape databases, they have limited success in handling occlusions. When the occlusion level is high, most of the methods fail to produce good recognition rates. Other papers sometimes say they are robust to occlusion but offer no evidence ([41, 47, 137, 32]) or offer evidence for small occlusions only (about 10%) ([40, 88]). These later papers do not explicitly model occlusion but treat occlusion as measurement error. To the best of our knowledge, we are the first to present a method to recognize plant leaves when the tested occlusion level is as high as 50%. Note that it is not the percentage of the occluded leaf that is available for matching that matters but the amount of "structure" (local variations) the occluded leaf has for matching purposes. Therefore various leaves could support more or less occlusion matching than we achieved for our leaves.

6.2 Algorithm Overview

First, we represent the 2D contour points as a β-Spline curve. We extract interest points on the curves via the Discrete Contour Evolution (DCE) algorithm. To find the best match of an occluded curve with segments of the full leaf curves in the database, we formulate our solution as a subgraph matching algorithm, using the feature points as graph nodes. This algorithm produces one or more open curves for each closed leaf contour considered. These open curves are matchable, to some degree, with the occluded curve. We then compute the affine parameters for each open curve and the occluded curve. After performing the inverse affine transform on the occluded curve, which allows the occluded curve and any subgraph curve to be "overlaid", we then compare the shapes using the Fréchet distance metric. We keep the best η matched curves. Since the Fréchet distance metric is cheap to compute but not perfectly correlated with the "goodness" of the match, we formulate an energy functional that is well correlated with the "goodness" of the match, but is considerably more expensive to compute. The functional uses local and global curvature, angular information and local geometric features. We minimize this energy functional using the well known convex-concave relaxation technique. The curve of the best η curves retained having the minimum energy is considered to be the best overall match with the occluded leaf. Experiments on three publicly available leaf image database shows that our method is both effective and efficient, outperforming other current state-of-the-art methods. Our algorithm can match leaves that are 50% occluded on their contour and still can identify best full leaf match from the databases.

6.3 Our Approach

Splines are powerful tools for representing a curve mathematically. Among the different types of spline curves, B-spline based contour representations has been successfully used for 2D shape matching ([50], [150], [152]). Due to its smoothness and continuity properties, B-splines are extremely useful in approximating the boundary of an object. B-spline curves are piecewise polynomial functions where local curve approximation is performed using control points.

---

[1] Full leaf contours are closed boundaries while occluded leavers are open boundaries.

Local control is extremely useful for modelling a contour to a desired level of detail. Another interesting property of B-spline is that it is affine invariant. This property is very useful when matching two curves which are related to each other by an affine transformation.

In general, a $(p+1)$-th order (i.e. p-th degree) B-spline is C^p continuous. A B-spline curve $P(u)$ defined by $(n+1)$ control points $P_0, P_1, \cdots, P_n$ is defined as:

$$P(u) = \sum_{i=\lfloor nu-2 \rfloor}^{\lceil nu+2 \rceil} B(nu-i)P_i, \tag{6.1}$$

where $B(nu-i)$ is the blending function of the spline (a bell shaped curve is non-zero when the inequality $-2 < nu - i < 2$ holds) and $u \in [0,1]$. In case of a cubic B-spline, at most four nearest control points are used to compute the blending function for a spline point (if $u = i/n$ then only 3 control points are needed).

For any set of 2D control points $P_i = (x_i, y_i)$, we can always obtain the spline point $(x(u), y(u))$ for any u. Thus the number of spline points can be greater than, equal to or less than the number of control points. Any curve thus can be represented as the polyline joining continuous set of spline points.

6.3.1 β-Spline Based Representation

Due to computational efficiency, cubic B-splines are usually used to model a contour [22]. Cubic B splines are good for approximating the curve whose shape is controlled by the control points. However, to interpret the control points one need to solve linear systems of points, which can be computationally expensive and complex. β-splines ([60, 61]) provide an intermediate representation between approximation and interpolation by providing a tension parameter, τ. $\tau = 0$ yields a B spline while $\tau > 0$ (say $\tau \approx 10$) provides a spline that almost perfectly interpolates the control points. [A second skew parameter, s, causes discontinuities in the spline curve and is usually ignored, i.e. we keep $s = 1$.] As for B splines, β-splines are 1^{st} and 2^{nd} order continuous and require at most 4 control points to produce a spline point value $P(u)$.

Other approaches to produce smooth contour curves include Thin Plate Splines (TPS) [34] or Relevance Vector Machine (RVM) regression techniques [57]. These techniques probably could be used in place of β splines with good results being obtained. We chose β-splines for simplicity and efficacy reasons.

Having $(n+1)$ control points $P_0, P_1, \cdots, P_n$ where the original contour points are used as the control points, a β-spline can be computed as:

$$P(u) = \sum_{i=\lfloor nu-2 \rfloor}^{\lceil nu+2 \rceil} \beta(nu-i)P_i, \tag{6.2}$$

where $u \in [0,1]$. Note that the minimum and maximum values of i ensure that the inequality $-2 < nu - i < 2$ is satisfied and $\beta(nu-i)$ values are (mostly) non-zero. [All other i values yield $nu - i$ values outside this range and so β is always 0. Note that a few β values could be zero because of the floor and ceiling functions used in the calculation of i but no non-zero β values are ever missed.] The blending function $\beta(\cdot)$ is given by ([11, 22]),

$$\beta(\tau) = \begin{cases} \frac{2}{\delta}(2+\tau)^3, & -2 \le \tau \le -1 \\ \frac{1}{\delta}((t+4s+4s^2) - 6(1-s^2)\tau - 3\tau^2(2+t+2s) \\ -2\tau^3(1+t+s+s^2)), & -1 \le \tau \le 0 \\ \frac{1}{\delta}((t+4s+4s^2) - 6\tau(s-s^3) - 3\tau^2(t+2s^2+ \\ 2s^3) + 2\tau^3(t+s+s^2+s^3)), & 0 \le \tau \le 1 \\ \frac{2}{\delta}s^3(2-\tau)^3, & 1 \le \tau \le 2, \end{cases} \tag{6.3}$$

where $\tau = nu - i$, t is the tension parameter and s is the skew parameter. δ is given by:

$$\delta = t + 2s^3 + 4s^2 + 4s + 2.$$

For our purpose, skew is not a useful parameter and we use $s = 1$. Like cubic B-spline, the blending function is still symmetrical with respect to τ, which implies $\beta(-\tau) = \beta(\tau)$. Also, the blending function has the following property:

$$\beta(\tau-2) + \beta(\tau-1) + \beta(\tau) + \beta(\tau+1) = 1. \tag{6.4}$$

A β-spline is second order continuous everywhere. For tension $t = 0$ and skew $s = 1$, the β-spline reduces to cubic B-spline. Positive values of tension ($t > 0$) increase the amplitude of the two middle segments of the third-order curve with respect to the first and last segments, whereas increasing the skew ($s > 1$) increases the amplitude of the two segments on the right with respect to the two segments to the left side of the curve and so may produce discontinuities and other unwanted artifacts.

6.4 Approximate Curve Section From Full Leaf

Once we have the β-spline representation of the leaf contour, we perform curve matching as discussed below. For the occluded test leaf, we need to determine the closest match of the curve section to the full leaves in the database. Matching two curves is a well known problem and has been well studied in Computer Vision for shape matching, handwriting recognition, etc. However, the solution is highly dependent on the application. For matching of curves representing leaf contours, the problem is even harder due to several factors. There are too many intra-class variations for many leaf species. Moreover, the boundary of the leaf contour with the background or other leaves might not be smooth due to segmentation errors. In our case, the curve matching problem is complicated because we have to determine which part of the full curve matches the occluded curve. One idea could be to use a brute force approach to consider every possible combinations of discrete points in the full curve and find the match with the occluded curve. This is a NP-hard problem [135] that can't be solved realistically for any reasonable number of contour points. We handle the problem in the following way.

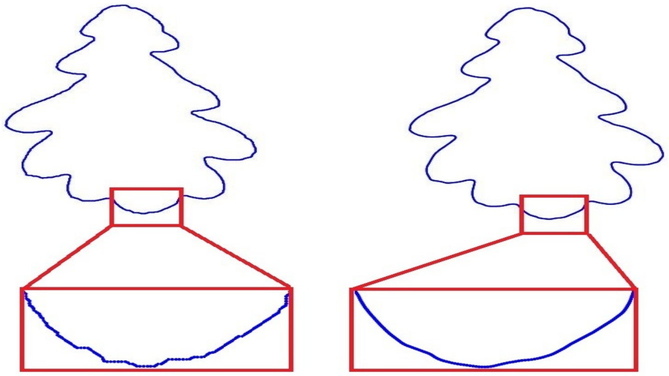


Figure 6.1: (Left) original unsmoothed leaf contour and (right) smoothed leaf contour.

First, we smooth the contour using the Savitzky-Golay filter. The idea of this filter is to perform convolution of the data points with a low degree polynomial by least squares. More specifically, we used the *smooth()* function in the Matlab curve fitting toolbox. The purpose of smoothing is to get rid of local noise and segmentation error at the boundary of the leaf. Figure 6.1 shows an example of original leaf contour (left image) and smoothed leaf contour (right image). A part of leaf contour is zoomed in to show the effect of this smoothing. However, a small neighbourhood should be chosen to perform the smoothing operation, otherwise the smoothing may suppress useful local geometrical structure of the leaf. With the smoothed curve sections, we next perform feature detection and then match the features of the two curves.

Although global shape descriptors have been shown to have some success in shape matching, the idea mostly works for shapes with large variations. Leaves not having unique distinguishing properties cannot be matched with traditional shape descriptors. Convexity is an important cue in human vision, and shapes can be decomposed into meaningful parts using convexity information. We use the Discrete Contour Evolution (DCE) [79] to find interest points in the 2D contour of the leaf. The idea of DCE is to simplify the contour by hierarchically decomposing the boundary by representing the shape with a polygon. The vertices of the polygon represent convex parts of the shape. Figure 6.2 shows examples of DCE features. The first leaf contour consists of lot of variations, whereas the second leaf contour lacks any meaningful features in the contour. One advantage of using DCE feature points is that, the number of features is invariant with respect to scale. Thus, two similar leaves of different sizes should have similar feature point pattern in their contours.

With the DCE feature points in the occluded and full contour, we next need to determine what is the best match between two set of feature points. We formulate the problem as a subgraph matching optimization problem.

6.4.1 Subgraph Matching

Finding the correspondence between two set of points is a fundamental problem in Computer Vision. Treating the feature points as graph nodes, the problem can be formulated as a subgraph isomorphism problem. Two graphs $G_1 = (E_1, V_1)$ and $G_2 = (E_2, V_2)$ are isomorphic, denoted

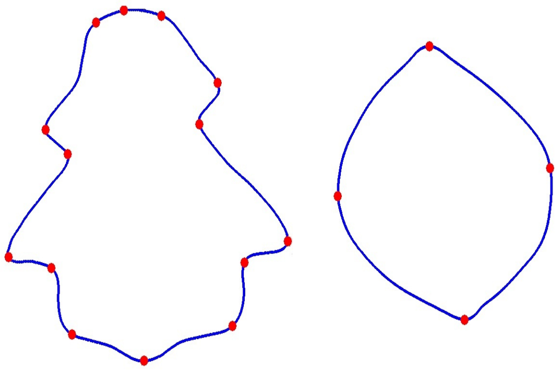


Figure 6.2: Feature points obtained from Discrete Contour Evolution (DCE) method [79]. The contour is decomposed into meaningful parts based on the convexity and the feature points represent the joint of these parts (or the vertices of the polygon represent the shape).

by $G_1 \cong G_2$, if there is a bijection $\varphi : V_1 \rightarrow V_2$ such that for every pair of vertices $v_i, v_j \in V_1$, edge $(v_i, v_j) \in E_1$ if and only if $(\varphi(v_i), \varphi(v_j)) \in E_2$. Although there are many exact isometric matching algorithms, our case is more complicated for several reasons. First, we need to deal with missing nodes. Graph nodes can be missing due to the inconsistency of the detected feature points for two similar leaves, which are slightly different in local geometry. Second, we have to consider the cases where where $G_1 \cong G_2$ or $G_2 \cong G_1$ because the number of feature points in the occluded curve can be less than the number of feature points of the full curve, and vice versa.

Figure 6.3 shows a few examples for matching an occluded curve section with some full curves in the database. In each of the four examples, the left curves are occluded curves and the right curves are the full curves. First, we extract the feature points on the contour via DCE as discussed previously. Then we build a connected graph with all the feature points as nodes and the geodesic distance as edges. In Figures 6.3a, 6.3b and 6.3c, the full curves are from same species, whereas in Figure 6.3d the full curve is from a different species. In each case, we want to find a match with an open curve section (or an occluded leaf). In the first three cases we have shown how the ambiguities in graph match can occur within a single species. Figure 6.3a shows a good match. Although it is not "exact", the overall topologies are similar in the occluded curve and the section of the full curve under consideration. Matches are shown in green lines. Now consider Figures 6.3b and 6.3c. Although these curves are also from the same species as in Figure 6.3a, there are matching ambiguities in the graph nodes. Matches with no ambiguities are shown as solid green lines while ambiguous matches are shown as dotted green lines. This is because one node in the occluded curve can match multiple nodes in the full curve and vice versa. Finally in Figure 6.3d, the occluded curve section is matched with a full curve from a different species. The vertices in the occluded curve still find matches with the vertices in the full curve, and there are some ambiguities in the matches. From all the examples in Figure 6.3, we infer that there is a need to quantify the quality of match by a "score". Also, we need to handle the cases of ambiguities of unmatched vertices.

We adopt the idea of graph matching from Mcauley *et al.* [95]. For two graphs G_1 and G_2,

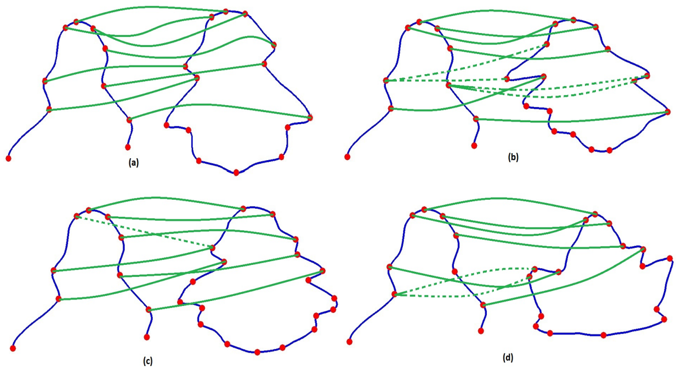


Figure 6.3: Four examples of graph matching. An occluded curve section is matched with 4 full curves in the database (the left curve is the occluded curve and the right curve is the full curve). In examples (a), (b) and (c), an occluded curve section is matched with full curves from the same species. In example (d), the same occluded curve is matched with a different species. Good matches are shown as solid green lines while ambiguous matches are shown as dotted green lines.

we assume that outliers can be present in either of the graphs. We wish to find a function $\hat{f}$: $G_1 \rightarrow G_2$ such that the distances between the points in G_1 and G_2 are minimized. Along with the penalty term for unmapped points between the two graphs due to occlusion and missing data, the graph matching problem can be formulated as the following objective function:

$$\begin{aligned}\hat{f} = \arg \min_{f:G_1 \rightarrow G_2} \sum_{(i,j) \in G_1} & |d(g_i, g_j) - d(f(g_i), f(g_j))| + \\ & \lambda \underbrace{(|G_1| - |f(G_1)|)}_{\text{number of unmapped points}} ,\end{aligned} \tag{6.5}$$

where d is the geodesic distance between the nodes and λ is the maximum number of outliers that are likely to be present. The above equation returns the arguments that minimize the expression: optimal cost and the matching vertices (see algorithm 3). We use the graph topology as proposed by Mcauley *et al.* [95] and use two nodes g_1 and g_2 as reference nodes. We order the vertices in counter clockwise direction and choose the first and last nodes as reference nodes. First we find the optimal mapping of two point sets using Algorithm 3. Then we repeat the process by selecting every pair of nodes as root node and find the optimal cost, which gives the optimal correspondence. After finding the best matched feature points between the two graphs, we retrieve the corresponding curve section from the full curve.

So far, we have extracted possible curve sections from all the full curves in the database. Because we perform the matching for all curves independently, it is unlikely to miss a potential match in the process. However, we still need to find the best match within hundreds or thousands of curves (depending on the size of the database). The idea is to use a two-stage technique to reduce the search space. Fist, we want to filter out the curves which are too different from each other in terms of global structure. This filtering is performed by first recovering

Algorithm 3 Subgraph matching algorithm

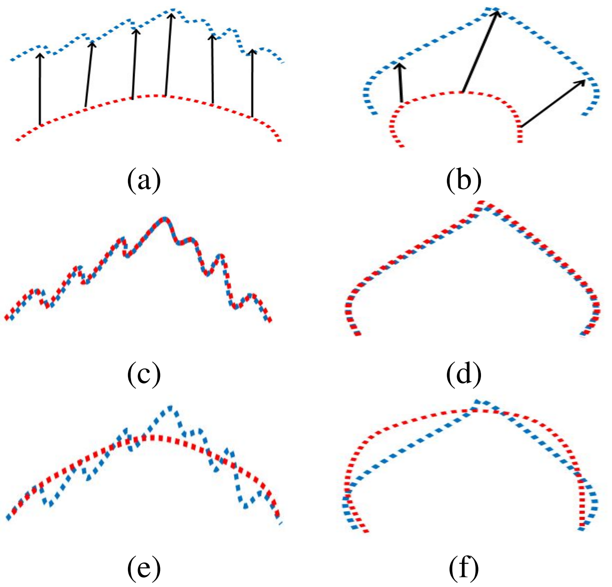
```
$best \leftarrow \phi$ , $bestcost \leftarrow \infty$
for each possible mapping $f(g_1), f(g_2) \in G_2 \times G_2$ do
    for $r \in \{1, -1\}$ do
        $cost \leftarrow \|g_1 - g_2| - |f(g_1) - f(g_2)\|$
        for for each node $g_i \in G_1 \backslash \{g_1, g_2\}$ do
            Find expected position $p$ of $f(g_i)$
            Find $p$'s nearest neighbour $n_p$
            $cost = cost + \|g_1 - g_i| - |f(g_1) - n_p\| + \|g_2 - g_i| - |f(g_2) - n_p\|$
            if $cost > bestcost$ then
                break
            end if
        end for
        if $cost < bestcost$ then
            $best \leftarrow f$
            $bestcost \leftarrow cost$
        end if
    end for
end for
return $best, bestcost$
```

the global affine transform parameters between any two curves. Then the Fréchet distance metric is used to measure the closeness between any two curves and only the best η matches are retained. In a last step, we perform energy optimization to find the overall best answer from these closely related curve matches (because, infrequently, the Fréchet error can sometimes incorrectly indicate a good match when the global structures are similar but the local structures are different). We discuss these steps in detail in next subsections.

6.4.2 Inverse Affine Transform

For the same leaf type, the occluded curve and the extracted curve sections from the full curves are assumed to be related by some global affine transformation. This allows us to recover the global translation and scale parameters (the rotation is already given by the graph matching calculation and we haven't encountered shear in any experimentation on our datasets). Usually the registration papers refer to the point set registration as warping, where the deformation parameters needed to warp one point set into another, are computed. However, we do not do this but rather use the affine parameters to "overlay" one curve on another and then perform a shape similarity calculation. Figure 6.4 illustrates this point.

Figure 6.4 show two examples curve matches to emphasize that we do not to do pairwise matching. The first row of Figure 6.4 shows two point sets from two different curves. The model set is represented in blue and the data set is represented in red. Some corresponding points are shown via black arrows in Figures 6.4a and 6.4b. Note that, since we have modelled the contour with a β-spline, we can re-generate exactly same number of equally spaced points for both spline curves.


(a) (b)

(c) (d)

(e) (f)

Figure 6.4: (a) and (b) the correspondence of some points on 2 curves, (c) and (d) the registration (warping) of the 2 curves and (e) and (f) the 2 curves overlaid via the computed affine parameters (rotation, translation and scale).

If we perform pairwise registration (warping), the algorithm will find corresponding points between two curves, and then the warped data set will "approach" the model set shape. This transformation is specified using local and global parameters. An efficient pairwise registration algorithm will consider local deformations and will transform each point in the data set to the nearest point in the model set. This results in something like that shown in Figures 6.4c and 6.4d. The local structures are treated as deformations and finally the registration aligns or warps the two point sets to coincide.

We do not model these deformations in our solution because the goal is to find the difference between two curves, and not to find the correspondences between two point sets. Undoing the affine transform between the two curves does not affect the local structure, it brings the two curves into an "overlaid" state. Note that, as we have matched feature points via graph matching and then extracted the curve section from that calculation, this is automatically rotation invariant. Undoing the affine transform is a simple but effective way to handle translation and scaling as well in the global sense. We compute the affine transform for two curves using the MatLab function **fitgeotrans**. We show examples of undoing an affine transformation in the third row of the figure, i.e. in Figures 6.4e and 6.4f. In these figures, the two curves still have very different structures.

For points $\mathbf{p}_i = (p_{i1}, \ldots, p_{in})^T$ and their corresponding affine transformed points $\mathbf{q}_i = (q_{i1}, \ldots, q_{in})^T$ $(i = 1, ..., m)$, the goal is to find the matrix $\mathbf{A}$ and the translation vector $\mathbf{t}$ such that

$$\mathbf{p}_i \approx \mathbf{A}\mathbf{q}_i + \mathbf{t}. \tag{6.6}$$

$\mathbf{A}$ incorporates rotation and scaling. To find $\mathbf{A}$ and $\mathbf{t}$, the following objective function must be minimized:

$$S(\mathbf{A}, \mathbf{t}) = \sum_{i=1}^{m} ||\mathbf{p}_i - \mathbf{A}\mathbf{q}_i - \mathbf{t}||_2^2 \,. \tag{6.7}$$

6.4.3 Curve matching by *Fréchet Distance*

After undoing the affine transform, we measure the similarity between the two curves. A naive way is simply to sum up the squared difference of all the points from start to the end of the curves. However, that does not give a good result. Typically, the *Hausdorff distance* is used to find similarity between two point sets. By definition, two sets are close in the *Hausdorff* sense if every point of either set is close to some point of the other set. However, this idea is not effective for finding the similarity between two curves because it considers only the set of points, not their order on the curve. We use the *Fréchet distance* [6] for measuring the similarity between two curves. Informally, it can be defined as the following example[2]: *Let us suppose that a man is walking a dog. Assume that the man is walking along one curve and the dog is along the other curve. Both of them are allowed to control their speed, but can't go backwards. The Fréchet distance between the two curves is the minimum length of the necessary leash to connect the man and the dog from the beginning till the end of the two curves.*

Mathematically, the continuous *Fréchet distance* between two curves, P and Q, is defined as:

$$F(P, Q) = \inf_{\alpha,\beta} \max_{t\in[0,1]} ||P(\alpha(t)) - Q(\beta(t))||,$$

where $P, Q : [0, 1] \to \mathbb{R}^2$ are parameterizations of the two curves and $\alpha, \beta : [0, 1] \to [0, 1]$. In the discrete case, the algorithm can be implemented using dynamic programming [99]. If the curves, P and Q, have sizes m and n respectively, then the discrete version of $F(P, Q)$ can be computed by following dynamic programming recurrences:

$$d_{0,0} := ||p_0 - q_0||_2$$

$$d_{0,j} := \max\{d_{0,j-1}, ||p_0 - q_j||\}, \text{ for j=1:n}$$

$$d_{i,0} := \max\{d_{i-1,0}, ||p_i - q_0||\}, \text{ for i=1:m}$$

$$d_{i,j} := \max\min\{d_{i,j-1}, d_{i-1,j}, d_{i-1,j-1}\}, \text{ for i=1:m, j=1:n}$$

Using the *Fréchet distance* metric, we see that similar curves tend to have smaller distance values. This approach helps to drastically reduce the search space. However, the approach focuses mostly on the global shape of the leaf and can't handle the cases where the leaves have similar global shape, but have different local structures. We retain the top η matches from the curve matching algorithm as discussed above and process these further as discussed in the next section.

---

[2] https://en.wikipedia.org/wiki/Fréchet_distance

6.4.4 Objective Function

Now the recognition problem of searching the full database of leaves reduces to finding the best match of an occluded curve with η curves (which are parts of full curves) in the database. However, these η curves are very close to the test curve, especially in terms of global structure. To find the best match among these curves, we need to incorporate both local and global information to uniquely identify a curve. The tunable parameter η has value 5 in our work. This is based on empirical observation and is used to reduce the search space because energy optimization is computationally expensive. Because the problem of partial contour matching is NP-hard, performing energy optimization on the best η Fréchet matches is a reasonable suboptimal solution. Otherwise, a trivial solution of the partial shape matching problem would be to consider all possible combinations of contour points and find the best match, i.e. a brute force approach, which has exponential complexity. However, one can keep more than $\eta = 5$ matches in the energy optimization phase. This was not needed for our experimentation with the 3 leaf datasets. For other applications, η might indeed need to be changed.

Local curvature can be useful to encode the local geometric structure of a leaf. If the leaf does not have much variation (can be thought of as a "smooth boundary"), local curvature will have almost the same value at all points on the contour, except the tip. Leaves having variations at the boundary will have different curvature values at different points. To encode the global structure of the leaf, global curvature can be an useful characteristic. However, even if two leaves have similar local and global structures, one idea is to investigate how they differ in terms of how the boundary points are distributed relatively to each other. Another idea for encoding local and global geometrical structure of the contour, is to consider the orientation of other contour points with respect to a particular point. Relative angles of other points with a particular point can be used as this metric.

We formulate an objective function that involves several geometric features, such as curvature, local and global geometrical structures, etc. and minimize it to find the best match between two curves. We formulate the energy functional as:

$$\begin{aligned} E_{total} = \lambda_1 E_{localCurvature} + \lambda_2 E_{globalCurvature} + \\ \lambda_3 E_{angularDistribution} + \lambda_4 E_{stringCut}, \end{aligned} \tag{6.8}$$

where λ_1, λ_2, λ_3 and λ_4 are weight factors for each term, currently set to 0.25. Basically, we compute four adjacency matrices for the four terms in the function. For every point on the contour, we compute these factors for every other point and encode these in an adjacency matrix [157]. We explain each term below:

1. **Local Curvature:** Curvature is an important property for matching shapes. Geometrically, if φ is the angle between the tangent line and the x-axis, then the curvature of a curve $y = f(x)$ is defined as:

$$E_{localCurvature} = \left|\frac{d\varphi}{ds}\right| = \frac{y''}{[1 + (y')^2]^{3/2}}. \tag{6.9}$$

   We choose a small number of points (20 points) to define the local neighbourhood around a point of interest and perform the same for all points on the contour to get local curvature at those points. This information helps to capture the local geometric structure at the leaf contour.

2. **Global Curvature:** This is similar to local curvature, but a bigger neighbourhood is used to capture the global shape. Also, a bigger neighbourhood is used to smooth the curve while computing the global curvature, which helps to capture the global geometrical structure of the curve. We use $1/3$ of the contour length to compute global curvature. This captures the overall leaf shape in a global sense. Note that, we are not dealing with a closed curve and we use reflector at the end points.

3. **Angular Features:** As discrete points are uniformly distributed along the contour (because spline points are uniformly distributed in the interval $u \in [0, 1]$), their relative orientations differ when the shape changes. We considered incorporating relative angular orientation of the points in our objective function. However, instead of simply computing the angles, we use the Shape Context descriptors [12]. The idea is to consider a set of vectors originating from a point to all other points on the boundary and then compute the distribution of all the points over relative positions. For every point p_i, a histogram h_i is created in log-polar space, which uses the relative coordinates of all the points with respect to that point. Mathematically it is defined as [12]:

   $$h_i(k) = \#\{q \neq p_i : (q - p_i) \in bin(k)\},$$

   where k is the index of the angle-distance bins. The advantage of using log-polar space is that, the descriptor is more sensitive to the nearby points than to points which are further away. This helps to exploit the local geometric structure.

   If p_i and q_j are two points on two curves that are to be matched, then the cost of matching of the curves, computed with the χ^2 statistic, is given as:

   $$E_{angularDistribution} = \frac{1}{2}\sum_{k=1}^{K}\frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}.$$

4. **StringCut Features:** In addition to computing the angular distribution of points in the contour, another way to compute local geometric structure is to find the distribution of points in a small neighbourhood around a straight line. Inspired by the work of Wang *et al.* [148], we introduce “StringCut” features which contributes to the fourth term of our objective function. Figure 6.5 shows a few samples of the local neighbourhood around a point. By drawing a straight line (or “string”) through the end points (which “cuts” the set of points), there can be three possible set of points: the points on two sides of the straight line (can be at left and right side, or top and bottom side), and the points lying on the line. This idea allows to extract the local geometry of the neighbourhood.

   Let a neighbourhood be defined by the points $\{p_i, p_{i+1}, \cdots, p_j\}$ which starts from p_i and ends at p_j and the straight line which passes through the points is denoted by ξ_{ij}. The point sets above, below and on the line are denoted as S_a, S_b and S_o. For example, in the

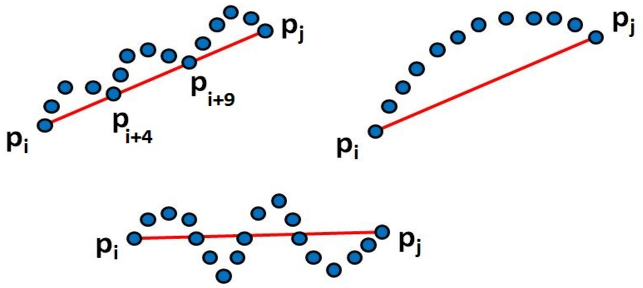


Figure 6.5: The idea of "StringCut". Some samples of neighbourhood around a point is shown. A straight line is drawn between the first and the last point of the neighbourhood and the distribution of the points around the straight line are used as StringCut features.

first example of Figure 6.5:

$$S_a = \{p_{i+1}, p_{i+2}, p_{i+3}, p_{i+5},$$
$$p_{i+6}, p_{i+7}, p_{i+8}, p_{i+10}, p_{i+11}, p_{i+12}, p_{i+13}\},$$

$$S_b = \{\phi\}$$

and

$$S_o = \{p_i, p_{i+4}, p_{i+9}, p_j\}.$$

Let $h(p_k, \xi_{ij})$ be the perpendicular distance of the point p_k to the straight line ξ_{ij}. The distance l from a point (x_0, y_0) to a straight line $ax + by + c = 0$ is simply given by,

$$l = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}.$$

Let $d(p_i, p_j)$ be the Euclidean distance between points p_i and p_j. Let L_{ij} be the geodesic length of the curve segment. Then we define f_{below}, f_{upper}, $f_{onTheLine}$ and $f_{bending}$ as the features for the points which are above the straight line, below the straight line, on the straight line, and the bending of the line respectively. Mathematically:

$$f_{below} = max\left(\frac{1}{N_a}\sum_{p_k \in S_a} h(p_k, \xi_{ij}), \frac{1}{N_b}\sum_{p_k \in S_b} h(p_k, \xi_{ij})\right), \tag{6.10}$$

$$f_{upper} = min\left(\frac{1}{N_a}\sum_{p_k \in S_a} h(p_k, \xi_{ij}), \frac{1}{N_b}\sum_{p_k \in S_b} h(p_k, \xi_{ij})\right), \tag{6.11}$$

$$f_{bending} = \frac{L_{ij}}{d(p_i, p_j)} \tag{6.12}$$

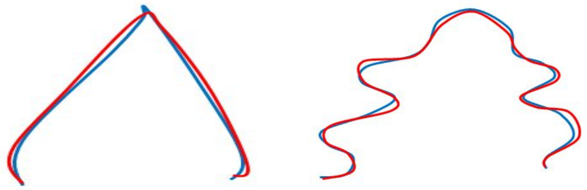
and

$$f_{onTheLine} = \{S_o\}, \tag{6.13}$$

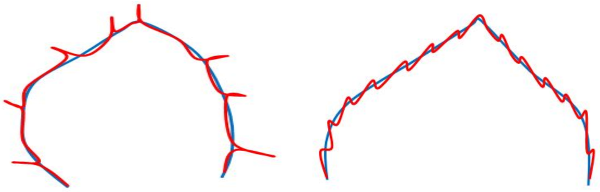
where N_a and N_b are the number of points in S_a and S_b respectively. Note that, the the points can also lie to the left and right side of the line, depending on the orientation. We combine all the StringCut features as:

$$E_{StringCut} = f_{below} + f_{upper} + f_{onTheLine} + f_{bending}. \tag{6.14}$$

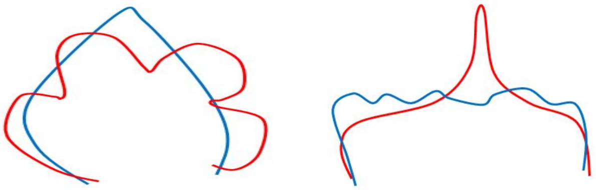
After computing all the terms in the objective function in Equation (6.8), we have to optimize it, which is discussed in next section.



(a) F=80,E=912 (b) F=90,E=963



(c) F=100,E=6322 (d) F=91,E=7255



(e) F=453,E=14432 (f) F=523,E=16227

Figure 6.6: Fréchet errors, F, and energy optimization values, E, for a number of curve sets. Curves (a) and (b) have both low F and E values, curves (c) and (d) have low F and high E values (showing the two are not always i well correlated) and curves (e) and (f) have both high F and E values.

It is instructive to consider the relationship between Fréchet matching error and the energy optimization values. Again, this demonstrates the need to use both. We use the Fréchet distance metric to cheaply compute the similarity of the two curves. The larger the curve differences, generally the greater the Fréchet errors. On the other hand, similar curves tend to have lower Fréchet errors. Consider Figure 6.6. Different curves are shown in the blue and red colours, and are overlaid (as determined from the inverse affine transform being applied to the results of the subgraph matching). The top row shows examples of curves which are similar in terms of global structure and local structures. In this case, low Fréchet errors, F, are well correlated with low optimization energy, E. The bottom row shows examples where the curves are very different. Now the F and E values are again well correlated but now are much larger, indicating poor matches. The middle row shows some (rare) cases, where the Fréchet errors are low but the energy values are high. Fréchet thresholding in this case would fail to remove these bad matches that the expensive energy optimization calculation would. The Fréchet error, by itself, would correctly accept/reject the matches in the top and bottom rows. Thus, the Fréchet errors and energy optimization play a complementary role of reduced computational costs with reasonable accuracy versus high accuracy at considerably increased computational costs.

6.5 GNCCP optimization

The energy functional defined in Equation (6.8) can be optimized in different ways. Because of the need for computational efficiency, we adopted a recently proposed efficient Graduated Non-Convex and Concavity Procedure (GNCCP) [89] approach. GNCCP basically solves the combinatorial optimization problem using permutation matrices. In our case, we have formulated the energy functional as a weighted combination of adjacency matrices. In another words, the problem is formulated as an assignment problem, where we explore all possible combinations of discrete points on the curve to find the best match (note that we are dealing with open curves at this stage). The adjacency matrix automatically handles the ordering of points (we chose the counter-clockwise direction). Also, all the curve sections have the same number of points, which is obtained by representing the original points by a β-spline and re-sampling the spline by controlling the increment parameter Δu, to obtain any specified number of equally spaced spline points.

The energy functional in our case is a combination of adjacency matrices, whose optimization gives the one-to-one correspondence of the points. Given two graphs, $G_1 = (E_1, V_1)$ and $G_2 = (E_2, V_2)$, where E and V are the set of edges and vertices respectively, a one-to-one mapping function $f : V_1 \rightarrow V_2$ specifies the correspondence between the two graphs. A typical approach to finding this solution is to formulate an energy functional, f, and minimize it. In general, this is an NP-hard combinatorial optimization problem [89].

Let the two graphs have M and N vertices respectively. We want to find the optimal matching of vertices between the two graphs. Let c_{ij} denotes the cost of matching i-th vertex of G_1 with j-th vertex of G_2, and $x_{ij} \in \{0, 1\}$ denotes the assignment. Then, the optimization problem can be written as:

$$\begin{aligned}\min_{\mathbf{X}\in\mathcal{P}} F(\mathbf{X}) &= \arg\min_{x} \sum_{i=1}^{M} \sum_{j=1}^{N} c_{ij} x_{ij} \\ &= \arg\min_{\mathbf{X}} \mathbf{X}^T \mathbf{C},\end{aligned} \tag{6.15}$$

such that the following condition is satisfied:

$$\begin{gathered}\mathbf{X} \in \mathcal{P}, \mathcal{P} := \Big\{\mathbf{X} | x_{ij} = \{0, 1\}, \\ \sum_{j=1}^{N} x_{ij} = 1, \sum_{i=1}^{M} x_{ij} \leq 1, \forall i, j\Big\}, M \leq N\end{gathered} \tag{6.16}$$

where $\mathcal{P}$ is the set of $(M \times N)$ permutation matrices.

Following the approach by Maciel *et al.* [93], the domain of the problem is relaxed from $\mathcal{P}$ to its convex hull, which is the set of doubly stochastic matrices Ω as follows:

$$\Omega := \Big\{ \mathbf{X} | x_{ij} \geq 0, \sum_{j=1}^{N} x_{ij} = 1, \sum_{i=1}^{M} x_{ij} \leq 1, \forall i, j \Big\}. \tag{6.17}$$

Then the GNCCP algorithm approximately solves the above problem as,

$$F_\zeta(\mathbf{X}) = \begin{cases} (1-\zeta)F(\mathbf{X}) + \zeta tr(\mathbf{X}^T\mathbf{X}) & if \;\; 1 \geq \zeta \geq 0 \\ \\ (1+\zeta)F(\mathbf{X}) + \zeta tr(\mathbf{X}^T\mathbf{X}) & if \;\; 0 \geq \zeta \geq -1, \end{cases} \tag{6.18}$$

where $\mathbf{X} \in \Omega$ and and $tr(\cdot)$ denotes the trace of a matrix. As the algorithm converges, the variable ζ decreases from 1 to -1. The steps are shown in Algorithm 4.

Algorithm 4 GNCCP algorithm

```
$\zeta \leftarrow 1, \mathbf{X} \leftarrow \mathbf{X}^0$
while $\zeta > -1 \wedge \mathbf{X} \notin \mathcal{P}$ do
    while $\mathbf{X}$ has not converged do
        $\mathbf{Y} = \arg\min_Y tr(\nabla F_\zeta(\mathbf{X})^T\mathbf{Y}), \mathbf{Y} \in \Omega$
        $\alpha = \arg\min_\alpha F_\zeta(\mathbf{X} + \alpha(\mathbf{Y} - \mathbf{X})), 0 \leq \alpha \leq 1$
        $\mathbf{X} \leftarrow \mathbf{X} + \alpha(\mathbf{Y} - \mathbf{X})$
    end while
    $\zeta \leftarrow \zeta - d\zeta$
end while
return $\mathbf{X}$
```

From the energy optimization performed as discussed above, we rank the η best Fréchet curves according to their energy values and choose as the best match the curve having the minimum energy.

6.6 Experimental Results

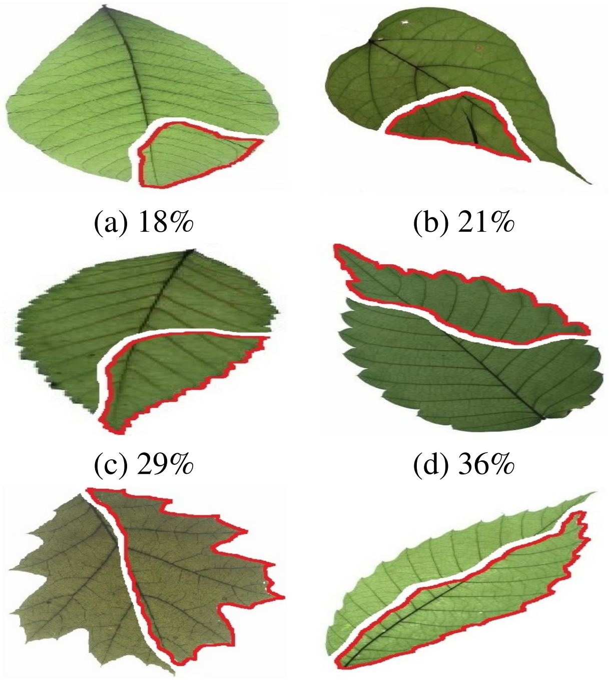
We validate the proposed method by applying it to 3 publicly available leaf datasets: the Swedish dataset [130], the Flavia dataset [151] and the Leafsnap dataset [77]. As pointed out by Hu *et al.* [65], the Smithsonian leaf dataset [85] contains too few samples per species, which makes it unsuitable for extensive experimentation. To make fair comparisons with the other algorithms, we compare our method with state of the art algorithms for which code is publicly available, or whose implementation is straightforward (the paper contains enough details to implement the idea). We compare with our algorithm with the Shape Context (SC) [3] [12] and the Inner Distance (IDSC) [4] [85] methods. Implementations of both algorithms are

---

[3] `https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/shape/sc_digits.html`

[4] `http://www.dabi.temple.edu/~hbling/code_data.htm`

publicly available. IDSC actually uses SC but improves it by using Dynamic Programming (DP). Also we have implemented Multiscale Distance Matrix (MDM) [65].


(a) 18% (b) 21%

(c) 29% (d) 36%

(e) 50% (f) 49%

Figure 6.7: Some typical occluded leaves with varying contour occlusion levels (specified below each image as a percentage). The part of each leaf enclosed by a red contour is cut away. The partial leaf contour remaining (not including the contour the cut away leaf part makes with the leaf) is the open occluded curve.

For occlusion, we randomly apply 20% to 50% occlusion to the test leaf. By this we mean the percentage of the overall contour that is occluded. We do not consider how much leaf area is occluded. The astute reader will realize that a leaf can have a low amount of leaf contour occlusion and, at the same time, a high amount of leaf area occlusion. For example, a leaf can have 10% contour occlusion but with most of its interior area (say 90%) cut away.

We also assume that the occlusion boundary is known. That is, we have an open curve as input. Occlusion boundary detection is a different problem, and that is not the focus of the paper. Figure 6.7 shows some examples of occluded leaves. The top row shows lower level occlusion, the middle row shows medium level of occlusion, and the last row shows examples when the leaves are highly occluded (up to 50%). Note that the occlusion level is considered at the contour, occlusion in the interior of the leaves are not taken into consideration. A leaf can be 90% occluded in terms of its area, and the occlusion at the contour can be 10%! Our algorithm may be able to handle these cases since we classify the leaves by their contours.

Sometimes, it is difficult to distinguish the leaf species if the amount of occlusion is high. For example, consider Figure 6.8, which shows 4 leaf sets with leaves from different species. Even for small occlusion levels, discriminating among the leaf species is difficult. At 50% occlusion it would be impossible. In our work, we consider any of close leaf species as the

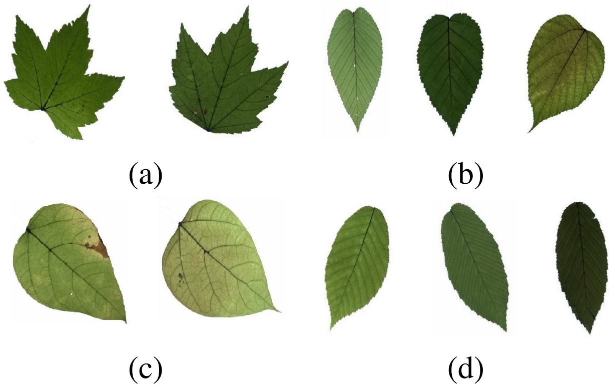


Figure 6.8: Leaves with similar contours for different species: leaves from (a) Sycamore Maple (acer pseudoplatanus) and Red Maple (acer rubrum), (b) Sweet Birch (betula lenta), Whitebark Himalayan Birch (betula jacquemontii) and Paper Mulberry (broussonetxtia papyrifera), (c) Southern Catalpa (catalpa bignonioide) and Northern Catalpa (catalpa speciosa) and (d) Yellow Birch (betula alleghaniensis), Downy Serviceberry (amelanchier arborea) and European Hornbeam (carpinus betulus). Note that we only classify these leaves based on their contour shapes and not on their texture, even though some of the species are clearly distinguishable by their texture.

correct answer to the classifier. Classifying these types of occlusion cases is out of the scope of this paper (the same issue arises for full leaf matching and was handled in the same way [85]). We present and discuss different datasets and corresponding recognition results below.

6.6.1 Swedish dataset

The dataset contains 1125 leaves from 15 species with 75 images per species. Following the protocol of previous work ([85], [48], [65], [148], [162]), we chose 25 images as database images and the rest as the testing images for each species. Also, we have removed the stems from the leaves [65]. As stated previously, we have applied random occlusion to the contour, ranging from 20% to 50%. We present the results of 0% occlusion (full leaf), 25% occlusion, 50% occlusion and the average performance of the algorithm for random occlusion levels from 20% to 50% as precision-recall curves in Figure 6.9. Figure 6.9a shows the performance when there is no occlusion. The performance of our algorithm is the same as state of the art. However, as the occlusion level is increased, our algorithm starts outperforming state of the art. As shown in Figure 6.9d, the average performance of our algorithm is significantly better than the state of the art for higher amounts of occlusion.

6.6.2 Flavia dataset

The dataset contains 32 classes with different numbers of samples per species. We have selected 50 images from each species (because that is the minimum number of samples per species). 25 of these are used as database images and 25 are used as the testing images. We performed similar analysis as previously discussed for the Swedish dataset, and the results are shown in Figure 6.10. Like the results on the Swedish dataset, we outperform state of the art.

6.6.3 Leafsnap dataset

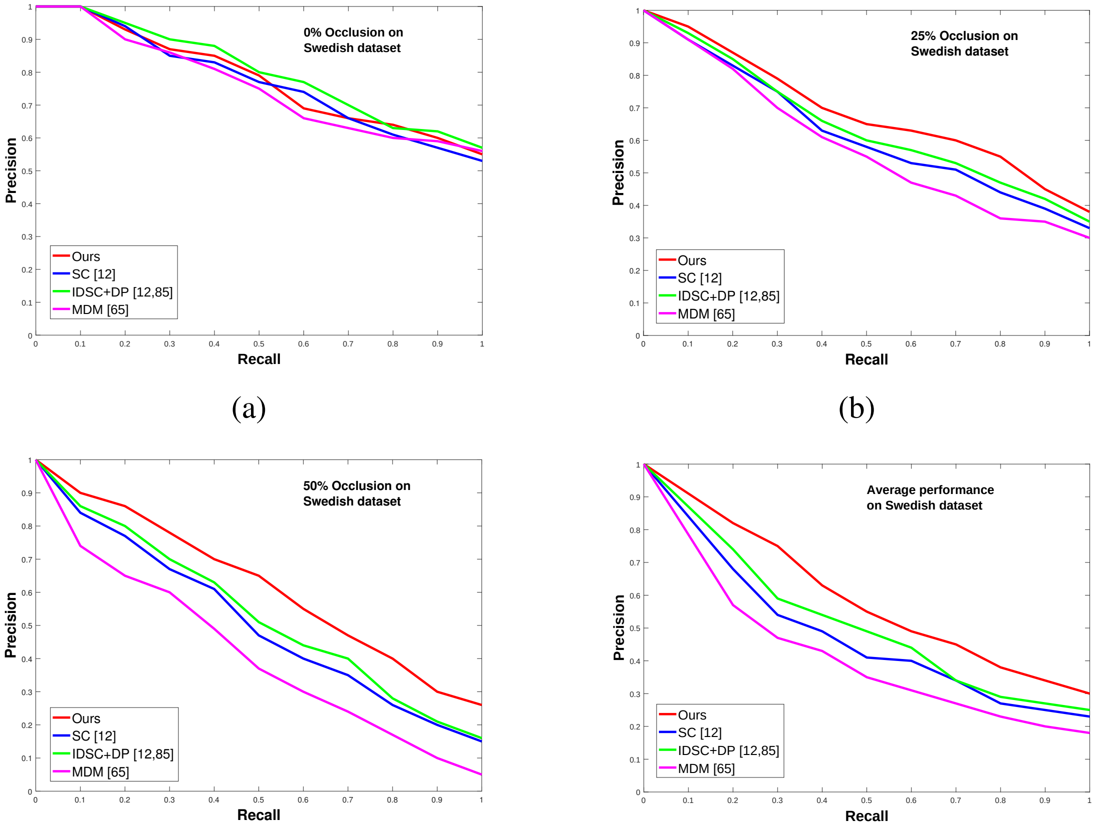
This dataset contains 184 species, having thousands of leaves in total. However, most of the species have a limited number of samples, which are not suitable for testing. Moreover, due to segmentation errors, many samples are not clean. Also, we exclude compound leaves data. Keeping all this in mind, we have selected a subset of 25 species from the dataset. In each species, 20 images are used as database images and the rest (depending on the number of available "clean" images) are used as the testing images. Results, shown in Figure 6.11, are better than state of the art as in Swedish and Flavia dataset results.

6.6.4 Discussion

In all experiments using all the different datasets, we outperform the state-of-the-art. The average recognition rate for all the datasets is about 70.2%. However, from the experimental results, it can be observed that our method performs the best for all occluded cases. When there is no occlusion, IDSC [85] performs slightly better than our method while our performance is quite close to SC [12]. In general, SC and IDSC perform similarly, while both outperform MDM [65]. Although both SC and IDSC are widely used successfully on a large variety of shapes, their performance drops for occluded leaves because both of the methods perform point to point matching, and there is no way of determining which part of a full curve best matches the occluded leaf. A brute force approach would be to perform point-to-point matching for all possible discrete combinations of the full curves in the database and find the best match, which is an NP-hard problem. One of the major contributions of our work is determining the section of the full curve that closely resembles the occluded curve section.

A drawback of the proposed method is that it is somewhat slow. The optimization takes the majority of the time. As stated in Zhao *et al.* [162], the searching step is the main bottleneck in leaf recognition using a large dataset. We use $n = 1000$ points for the curves and the GNCCP algorithm is $O(n^3)$. Even using MatLab's **parfor** to spawn separate computations for each of the 5 curves does not speed things up enough. Reducing the number of contour points negatively affects the recognition accuracy.

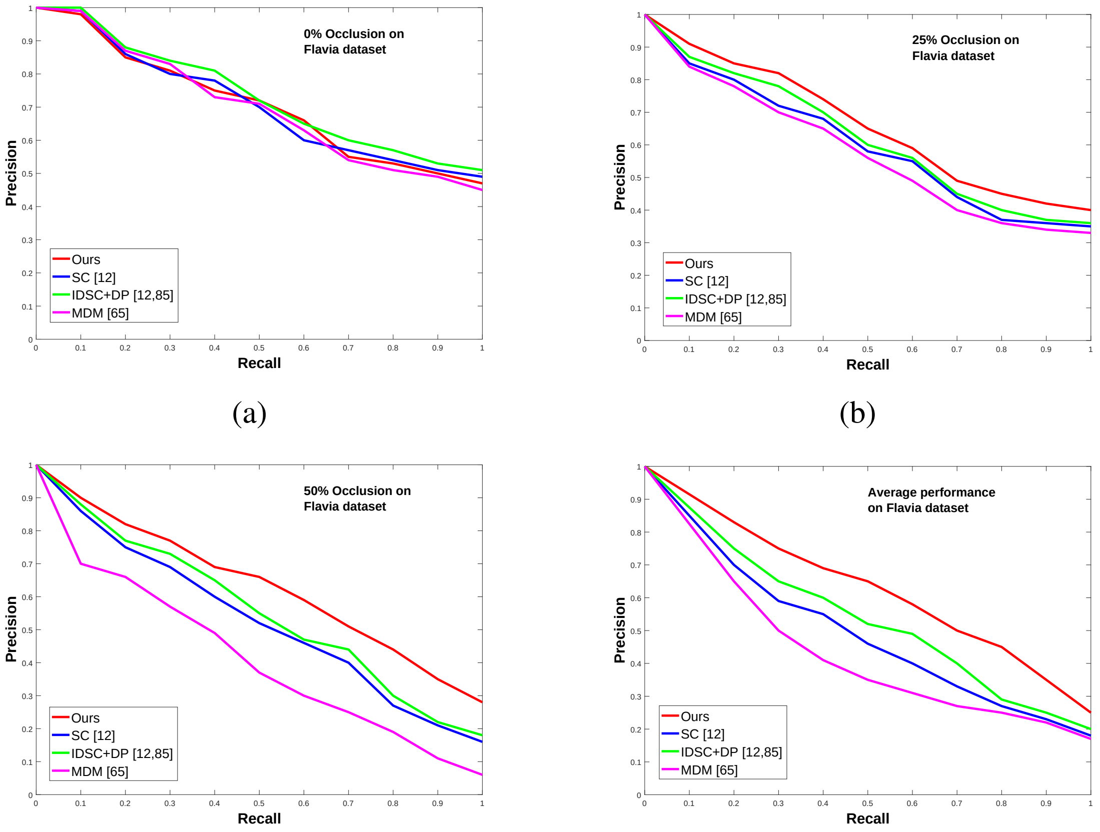
Lastly, note that our method is general enough for use in many different applications of partial shape matching.



(a) (b)

(c) (d)

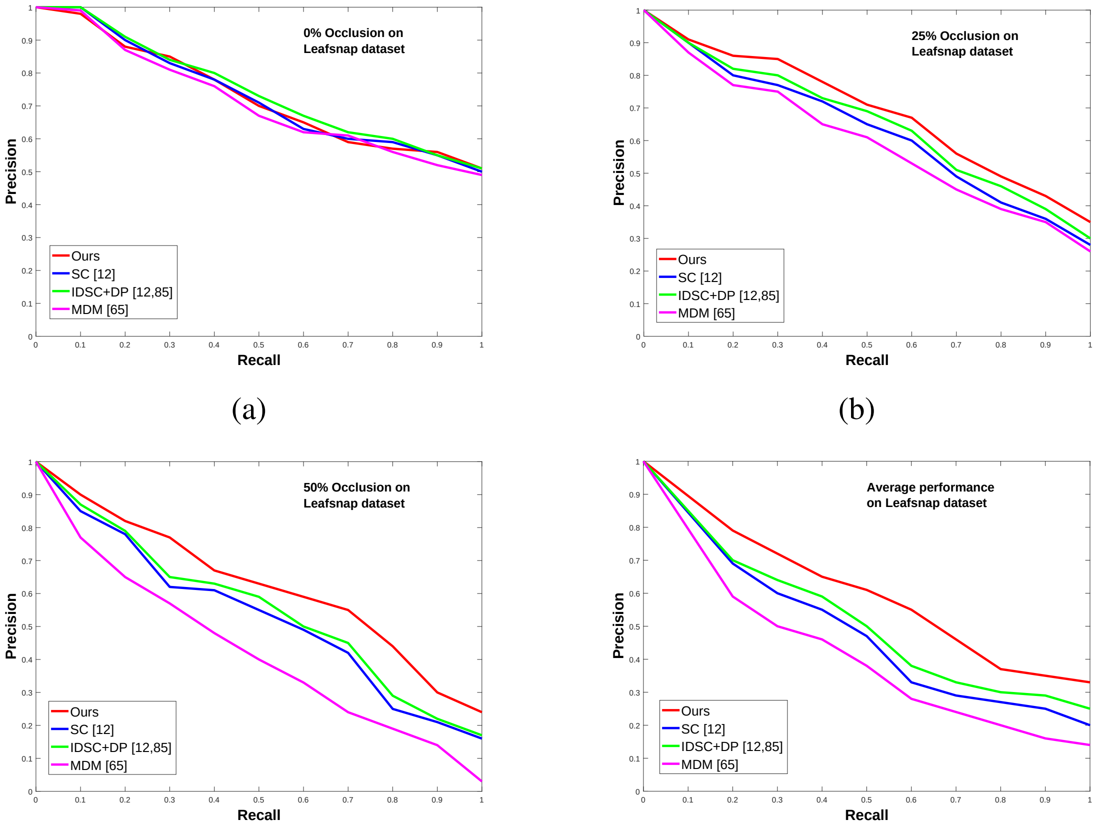
Figure 6.9: Precision-Recall curves for the Swedish dataset with (a) no occlusion, (b) 25% occlusion, (c) 50% occlusion and (d) average performance for random occlusion levels from 20% to 50% respectively.



(a) (b)

(c) (d)

Figure 6.10: Precision-Recall curves for the Flavia dataset with (a) no occlusion, (b) 25% occlusion, (c) 50% occlusion and (d) average performance for random occlusion levels from 20% to 50% respectively.



(a) (b)

(c) (d)

Figure 6.11: Precision-Recall curves for the Leafsnap dataset with (a) no occlusion, (b) 25% occlusion, (c) 50% occlusion and (d) average performance for random occlusion levels from 20% to 50% respectively.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis we have focused on different problems related to automated plant growth analysis. We have shown that 3D surface area and volume can be reliably used in some cases as growth metric for plants. This type of system can be used for comparing growth patterns of different types, varieties and species in different environmental conditions in an autonomous way. Using the junctions as feature points, the 3D reconstruction of the plant is possible without any prior knowledge of the scans. The robot can also be used to perform tracking of different plant organs over time. The partial contour matching algorithm can be extended to estimate the full structure of an occluded leaf, which can be helpful in tracking leaves over time. We believe that the proposed system is general enough to perform various biological relevant experiments in a non-invasive/non-contact and automatic manner.

7.2 Future Work

From the system point of view, one limitation of the system is that, it can process one plant at a time, but this can be extended for future scanning. To increase plant throughput, we can scan multiple plants at a time. Zhao [163] and Yang [156] processed 4 plants at a time, but performed manual cutting (editing) of the range data to separate the data for each plant. One idea can be to change the robot trajectory to scan each plant separately, but some plant overlap may be impossible to avoid. So future work could be changing the sensor trajectory and/or performing automatic editing. Leafy plants will also be problematic for our system as the volume measured for the plant and its actual volume can now be quite different. Another question is whether the mesh enclosing a plant (which may not necessarily be a good estimate of the plant's volume) can be used as a measure of the plant's growth. One idea to handle occlusion can be to exploit the 7 degrees of freedom of our highly flexible robot arm and move it to the occluded areas dynamically to perform scanning more efficiently. Such areas might be found quickly by processing the grayvalue images found by the scanning. Such scanning would have to be performed locally (and not on Sharcnet) to allow dynamic real time re-adjustment of the scanner's trajectory.

From an algorithmic point of view, 3D reconstruction of highly occluded plants will be challenging problem (as discussed in Chapter 4 for the bean plant). Although increasing the number of scans might solve the problem, this would also increase the computational time. The minimum number of scans required to accurately reconstruct the structure of a plant might be a good study which can be used to optimize the computational cost of multiview alignment (trade-off between accuracy and computation cost). Zhao [163] made a start of that kind of work; she had a lot of difficulty in obtaining accurate volumes for plants that she sacrificed. As well, there were difficulties in scanning plastic aquarium plants as the laser beam tends to penetrate the plastic, meaning actual plants could not be replaced by aquarium plants in experiments to precisely determine the minimum number of scans required (may vary for plant types).

The recognition rate and computation time required for partial leaf recognition need to be improved. Also, we haven't studied compound leaves and species which are very close to each other. We have demonstrated partial contour matching in 2D, which needs to be extended for 3D. Another interesting approach might be to use deep learning to solve the partial contour matching problem.

Bibliography

[1] T. Adamek and N. E. O'Connor. A multiscale representation method for nonrigid shapes with a single closed contour. *IEEE Transactions on Circuits Systems for Video Technology*, 14(5):742–753, 2004.

[2] S. Chattopadhyay S. A. Akbar, N. M. Elfiky, H. Medeiros, and A. C. Kak. Measuring and modeling apple trees using time-of-flight data for automation of dormant pruning applications. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[3] E. E. Aksoy, A. Abramov, F. Wörgötter, H. Scharr, A. Fischbach, and B. Dellen. Modeling leaf growth of rosette plants using infrared stereo image sequences. *Computers and Electronics in Agriculture*, 110:78–90, 2015.

[4] Quazi Mah-Zereen Akter. Automation of the registration of range plant images using geomagic studio. Master's thesis, Dept. of Computer Science, University of Western Ontario, 2012.

[5] G. Alenyà, B. Dellen, and C. Torras. 3d modelling of leaves from color and tof data for robotized plant measuring. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[6] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry and Applications*, 5:75–91, 1995.

[7] M. Augustin, Y. Haxhimusa, W. Busch, and W. G. Kropatsch. A framework for the extraction of quantitative traits from 2D images of mature arabidopsis thaliana. *Machine Vision and Applications*, 2015.

[8] X. Bai, C. Rao, and X. Wang. Shape vocabulary: A robust and efficient shape representation for shape matching. *IEEE Transactions on Image Processing*, 23(9), 2014.

[9] J.L. Barron and A. Liptay. Optic flow to measure minute increments in plant growth. *BioImaging (BI1994)*, 2(1):57–61, 1994.

[10] J.L. Barron and A. Liptay. Measuring 3d plant growth using optical flow. *BioImaging (BI1997)*, 5(2):pp82–86, 1997.

[11] B. A. Barsky and J. C. Beatty. Local control of bias and tension in beta-splines. *ACM Transactions on Graphics*, 2(2):109–134, 1983.

[12] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[13] L. Benoit, D. Rousseau, E. Belin, D. Demilly, and F. Chapeau-Blondeau. Simulation of image acquisition in machine vision dedicated to seedling elongation to validate image processing root segmentation algorithms. *Computers and Electronics in Agriculture*, 104, 2014.

[14] P. J. Besl and N. D. McKay. A method for registration of 3d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 1992.

[15] M. Bicego and P. Lovato. A bioinformatics approach to 2d shape classification. *Computer Vision and Image Understanding*, 145:59–69, 2016.

[16] A. Bienert and H.G. Maas. Methods for the automatic geometric registration of terrestrial laser scanner point clouds in forest stands. *Inter. Archives Photogramm., Remote Sens. Spatial Inf. Sci.*, 38(3/W8), 2009.

[17] S. Bouaziz, A. Tagliasacchi, and M. Pauly. Sparce iterative closest point. *Computer Graphics Forum (Symposium on Geometry Processing)*, 32(5):1–11, 2013.

[18] M. Brophy, A. Chaudhury, S. S. Beauchemin, and J. L. Barron. A method for global nonrigid registration of multiple thin structures. In *Proceedings of the 12th Conference on Computer and Robot Vision (CRV)*, pages 214–221, 2015.

[19] Mark Brophy. *Surface Reconstruction from Noisy and Sparse Data*. PhD thesis, Dept. of Computer Science, University of Western Ontario, December 2015.

[20] A. Bucksch and K. Khoshelham. Localized registration of point clouds of botanic trees. *IEEE Geoscience and Remote Sensing Letters*, 10(3), 2013.

[21] A. Bucksch, R.C. Lindenbergh, M. Zulkarnain A. Rahman, and M. Menenti. Breast height diameter estimation from high-density airborne LiDAR data. *IEEE Geoscience and Remote Sensing Letters*, 11(6), 2014.

[22] P. Burger and D. Gillies. *Interactive Computer Graphics*. Addison-Wesley Publishing Company, 1989.

[23] Y. Cao, Z. Zhang, I. Czogiel, I. Dryden, and S. Wang. 2D nonrigid partial shape matching using MCMC and contour subdivision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[24] G. Cerutti, L. Tougne, D. Coquin, and A. Vacavant. Curvature-scale-based contour understanding for leaf margin shape recognition and species identification. In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, 2013.

[25] G. Cerutti, L. Tougne, D. Coquin, and A. Vacavant. Leaf margins as sequences: A structural approach to leaf identification. *Pattern Recognition Letters*, 49:177–184, 2014.

[26] G. Cerutti, L. Tougne, J. Mille, A. Vacavant, and D. Coquin. A model-based approach for compound leaves understanding and identification. In *IEEE International Conference on Image Processing (ICIP)*, 2013.

[27] G. Cerutti, L. Tougne, J. Mille, A. Vacavant, and D. Coquin. Understanding leaves in natural images - A model-based approach for tree species identification. *Computer Vision and Image Understanding*, 117(10):1482–1501, 2013.

[28] A. Chaudhury and J. L. Barron. Partially occluded leaf recognition via subgraph matching and energy optimization. *(in review)*, 2017.

[29] A. Chaudhury, M. Brophy, and J. L. Barron. Junction based correspondence estimation of plant point cloud data using subgraph matching. *IEEE Geoscience and Remote Sensing Letters*, 13(8):1119–1123, 2016.

[30] A. Chaudhury, C. Ward, A. Talasaz, A. G. Ivanov, M. Brophy, N.P.A. Hüner B. Grodzinski, R.V. Patel, and J.L. Barron. Machine vision system for 3D plant phenotyping. *(in review)*, 2017.

[31] A. Chaudhury, C. Ward, A. Talasaz, A. G. Ivanov, N. P. A. Hüner, B. Grodzinski, R. V. Patel, and J. L. Barron. Computer vision based autonomous robotic system for 3D plant growth measurement. In *Proceedings of 12th conference on Computer and Robot Vision(CRV)*, pages 290–296, 2015.

[32] L. Chen, R. Feris, and M. Turk. Efficient partial shape matching using smith-waterman algorithm. In *IEEE Conference on Computer Vision and Pattern Recognition workshops (CVPRW)*, 2008.

[33] H. Chui and A. Rangarajan. A feature registration framework using mixture models. In *Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA)*, 2000.

[34] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, 2003.

[35] F. S. Cohen, Z. Huang, and Z. Yang. Invariant matching and identification of curves using B-splines curve representation. *IEEE Transactions on Image Processing*, 4(1), 1995.

[36] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 313–320, 2007.

[37] T. Darom and Y. Keller. Scale invariant features for 3D mesh models. *IEEE Transactions on Image Processing*, 21(5):2758–2769, 2012.

[38] B. Dellen, H. Scharr, and C. Torras. Growth signatures of rosette plants from time-lapse video. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 12(6):1470–1478, 2015.

[39] K. Deng, J. Tian, J. Zheng, X. Zhang, X. Dai, and M. Xu. Retinal fundus image registration via vascular structure graph matching. *International Journal of Biomedical Imaging*, 2010:14:1–14:13, 2010.

[40] C. Domokos, J. Nemeth, and Z. Kato. Nonlinear shape registration without correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4), 2012.

[41] M. Donoser, H. Riemenschneider, and H. Bischof. Efficient partial shape matching of outer contours. In *Asian Conference on Computer Vision (ACCV)*, 2009.

[42] T. Dornbusch, S. Lorrain, D. Kuznetsov, A. Fortier, R. Liechti, I. Xenarios, and C. Fankhauser. Measuring the diurnal pattern of leaf hyponasty and growth in arabidopsis a novel phenotyping approach using laser scanning. *Functional Plant Biology*, 39(11):860–869, 2012.

[43] O. Duchenne, F. Bach, I. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(33), 2011.

[44] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, 1983.

[45] M. Ester, H-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD), Portland, Oregon, USA*, 1996.

[46] J. Fan, N. Zhou, J. Peng, and L. Gao. Hierarchical learning of tree classifiers for large-scale plant species identification. *IEEE Transactions on Image Processing*, 24(11):4172–4184, 2015.

[47] C. Feinen, C. Yang, O. Tiebe, and M. Grzegorzek. Shape matching using point context and contour segments. In *Asian Conference on Computer Vision (ACCV)*, 2014.

[48] P. F. Felzenszwalb and J. D. Schwartz. Hierarchical matching of deformable shapes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[49] V. Ferrari, T. Tuytelaars, and L. Gool. Object detection by contour segment networks. In *European Conference on Computer Vision (ECCV)*, 2006.

[50] M. A. T. Figueiredo, J. M. N. Leitao, and A. K. Jain. Unsupervised contour representation and estimation using b-splines and a minimum description length criterion. *IEEE Transactions on Image Processing*, 9(6):1075–1087, 2000.

[51] F. Fiorani and U. Schurr. Future scenarios for plant phenotyping. *Annual Review of Plant Biology*, 64, 2013.

[52] A. W. Fitzgibbon. Robust registration of 2d and 3d point sets. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2001.

[53] T. Fourcaud, X. Zhang, A. Stokes, H. Lambers, and C. Körner. Plant growth modelling and applications: The increasing importance of plant architecture in growth models. *Annals of Botany*, 101:1053–1063, 2008.

[54] R. T. Furbank1 and M. Tester. Phenomics - technologies to relieve the phenotyping bottleneck. *Trends in Plant Science*, 16(12):635–644, 2011.

[55] E. Garnier and A. Berger. Effect of water stress on stem diameter changes of peach trees growing in the field. *Journal of Applied Ecology*, 23(1):193–209, 1986.

[56] T. Genard, S. Fishman, G. Vercambre, J-G. Huguet, C. Bussi, J. Besset, and R. Habib. A biophysical analysis of stem and root diameter variations in woody plants. *Plant Physiology*, 126(2), 2001.

[57] D. Gerogiannis, C. Nikou, and A. Likas. Registering sets of points using bayesian regression. *Neurocomputing*, 89:122–133, 2012.

[58] C. Godin and P. Ferraro. Quantifying the degree of self-nestedness of trees: Application to the structural analysis of plants. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(4):688–703, 2010.

[59] F. Golbach, G. Kootstra, S. Damjanovic, G. Otten, and R. Zedde. Validation of plant part measurements using a 3D reconstruction method suitable for high-throughput seedling phenotyping. *Machine Vision and Applications*, 2015.

[60] T. N. T. Goodman. Properties of beta splines. *Journal of Approximation Theory*, 44(2):132–153, 1985.

[61] T. N. T. Goodman and K. Unsworth. Manipulating shape and producing geometuic contnuity in beta spline curves. *IEEE Computer Graphics and Applications*, 6(2):50–56, 1986.

[62] J.A. Hartigan and P.M. Hartigan. The dip test of unimodality. *Annals of Statistics*, 13(1):70–84, 1985.

[63] A. Hartmann, T. Czauderna, R. Hoffmann, N. Stein, and F. Schreiber. HTpheno: An image analysis pipeline for high-throughput plant phenotyping. *BMC Bioinformatics*, 2011.

[64] J.G. Henning and P.J. Radtke. Multiview range-image registration for forested scenes using explicitly-matched tie points estimated from natural surfaces. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(1), 2008.

[65] R. Hu, W. Jia, H. Ling, and D. Huang. Multiscale distance matrix for fast plant leaf recognition. *IEEE Transactions on Image Processing*, 21(11):4667–4672, 2012.

[66] R. Hu, W. Jia, H. Ling, Y. Zhao, and J. Gui. Angular pattern and binary angular pattern for shape retrieval. *IEEE Transactions on Image Processing*, 23(3), 2014.

[67] Z. Huang and F. S. Cohen. Affine-invariant B-spline moments for curve matching. *IEEE Transactions on Image Processing*, 5(10):1473–1480, 1996.

[68] D. F. Huber and M. Hebert. Fully automatic registration of multiple 3d data sets. *Image and Vision Computing*, 21(7):637–650, 2003.

[69] B. Jian and B.C. Vemuri. Robust point set registration using gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8), August 2011.

[70] A. R. Jiménez, R. C. Ruíz, and J. L. P. Rovira. A vision system based on a laser range-finder applied to robotic fruit harvesting. *Machine Vision and Applications*, 11(6):321–329, 2000.

[71] T. Jin, X. Hou, P. Li, and F. Zhou. A novel method of automatic plant species identification using sparse representation of leaf tooth features. *PLoS ONE*, 10, 10 2015.

[72] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5), 1999.

[73] D. Kelly, A. Vatsa, W. Mayham, and T. Kazic. Extracting complex lesion phenotypes in zea mays. *Machine Vision and Applications*, 27(1):145–156, 2016.

[74] M. Klodt, K. Herzog, R. Töpfer, and D. Cremers. Field phenotyping of grapevine growth using dense stereo reconstruction. *BMC Bioinformatics*, 16:143, 2015.

[75] C. Klukas, D. Chen, and J. M. Pape. Integrated analysis platform: An open-source information system for high-throughput plant phenotyping. *Plant Physiology*, 165(2):506–518, 2014.

[76] P. Kontschieder, M. Donoser, and H. Bischof. Beyond pairwise shape similarity analysis. In *Asian Conference on Computer Vision (ACCV)*, 2009.

[77] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. Lopez, and J. V. B. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *The 12th European Conference on Computer Vision (ECCV)*, October 2012.

[78] S. Lamprecht, A. Hill, J. Stoffels, and T. Udelhoven. A machine learning method for co-registration and individual tree matching of forest inventory and airborne laser scanning data. *Remote Sensing*, 9(5), 2017.

[79] J. K. Latecki and R. Lakämper. Convexity rule for shape decomposition based on discrete contour evolution. *Computer Vision and Image Understanding*, 73(3):441–454, 1999.

[80] L. J. Latecki, R. Lakämper, and U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.

[81] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *Proc. of CVPR*, June 2007.

[82] L. Li, Q. Zhang, and D. Huang. A review of imaging techniques for plant phenotyping. *Sensors*, 14(11), 2014.

[83] Y. Li, X. Fan, N. J. Mitra, D. Chamovitz, D. Cohen-Or, and B. Chen. Analyzing growing plants from 4d point cloud data. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2013)*, 32, 2013.

[84] Point Cloud Library. `http://pointclouds.org/`.

[85] H. Ling and D. W. Jacobs. Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):286–299, 2007.

[86] A. Liptay, J. L. Barron, T. Jewett, and I. V. Wesenbeeck. Oscillations in corn seedling growth as measured by optical flow. *J. Amer. Soc. Hort. Sci.*, 120(3), 1995.

[87] H. Liu, D. Coquin, L. Valet, and G. Cerutti. Leaf species classification based on a botanical shape sub-classifier strategy. In *22nd International Conference on Pattern Recognition (ICPR)*, 2014.

[88] M. Liu, B. C. Vemuri, S. Amari, and F. Nielsen. Shape retrieval using hierarchical total bregman soft clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12), 2012.

[89] Z. Liu and H. Qiao. GNCCP - graduated nonconvexity and concavity procedure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6), 2014.

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

[91] J. Ma, J. Zhao, and A. L. Yuille. Non-rigid point set registration by preserving global and local structures. *IEEE Transactions on Image Processing*, 25(1):53–64, 2016.

[92] T. Ma and L. J. Latecki. From partial shape matching through local deformation to robust global shape similarity for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[93] J. Maciel and J. P. Costeira. A global solution to sparse correspondence problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):187–199, 2003.

[94] T. Masuda and N. Yokoya. A robust method for registration and segmentation of multiple range images. *Computer Vision and Image Understanding (CVIU)*, 61(3):295–307, 1995.

[95] J. J. Mcauley and T. S. Caetano. Fast matching of large point sets under occlusions. *Pattern Recognition*, 45(1):563–569, 2012.

[96] A. Mian, M. Bennamoun, and R. Owens. On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes. *International Journal of Computer Vision*, 89(2-3), 2010.

[97] F. Mokhtarian and S. Abbasi. Matching shapes with self-intersections: application to leaf classification. *IEEE Transactions on Image Processing*, 13(5):653–661, 2004.

[98] F. Mokhtarian, S. Abbasi, and J. Kittler. Robust and efficient shape indexing through curvature scale space. In *British Machine Vision Conference (BMVC)*, 1996.

[99] A. Mosig and M. Clausen. Approximately matching polygonal curves with respect to the Fréchet distance. *Computational Geometry: Theory and Applications*, 30(2):113–127, 2005.

[100] S. Mouine, I. Yahiaoui, and A. Verroust-Blondet. Plant species recognition using spatial correlation between the leaf margin and the leaf salient points. In *IEEE International Conference on Image Processing (ICIP)*, 2013.

[101] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12), December 2010.

[102] A. Myronenko, X. Song, and M. A. Carreira-Perpinan. Non-rigid point set registration: Coherent point drift. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.

[103] Y. Nam, E. Hwang, and D. Kim. A similarity-based leaf image retrieval scheme: Joining shape and venation features. *Computer Vision and Image Understanding*, 110(2):245–259, 2008.

[104] P. J. Navarro, C. Fernández, J. Weiss, and . Marcos. Development of a configurable growth chamber with a computer vision system to study circadian rhythm in plants. *Sensors*, 12(11):15356, 2012.

[105] M. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[106] D. A. Nusinow, A. Helfer, E. E. Hamilton, J. J. King, T. Imaizumi, T. F. Schultz, E. M. Farré, and S. A. Kay. The ELF4-ELF3-LUX complex links the circadian clock to diurnal control of hypocotyl growth. *Nature*, 475(7356):398–402, 2011.

[107] A. Opelt, A. Pinz, and A. Zisserman. A boundary fragment model for object detection. In *European Conference on Computer Vision (ECCV)*, 2006.

[108] A. Paproki, X. Sirault, S. Berry, R. Furbank, and J. Fripp. A novel mesh processing based technique for 3d plant analysis. *BMC Plant Biology*, 12(1), 2012.

[109] S. Paulus, J. Dupuis, A. K. Mahlein, and H. Kuhlmann. Surface feature based classification of plant organs from 3d laserscanned point clouds for plant phenotyping. *BMC Bioinformatics*, 14(238), 2013.

[110] S. Paulus, J. Dupuisemail, S. Riedelemail, and H. Kuhlmann. Automated analysis of barley organs using 3D laser scanning: An approach for high throughput phenotyping. *Sensors*, 14(7):12670–12686, 2014.

[111] S. Paulus, H. Schumann, H. Kuhlmann, and J. Léon. High-precision laser scanning system for capturing 3d plant architecture and analysing growth of cereal plants. *Biosystems Engineering*, 121, 2014.

[112] M. P. Pound, A. P. French, J. A. Fozard, E. H. Murchie, and T. P. Pridmore. A patch-based approach to 3D plant shoot phenotyping. *Machine Vision and Applications*, 2016.

[113] M. P. Pound, A. P. French, E. H. Murchie, and T. P. Pridmore. Surface reconstruction of plant shoots from multiple views. In *Proceedings of ECCV Workshops*, 2014.

[114] K. Pulli. Multiview registration for large data sets. In *Proc. Second International Conference on 3-D Digital Imaging and Modeling*, pages 160–168, 1999.

[115] A. Punjani, M. A. Brubaker, and D. J. Fleet. Building Proteins in a Day: Efficient 3D Molecular Structure Estimation with Electron Cryomicroscopy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[116] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, 2001.

[117] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[118] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz. Aligning point cloud views using persistent feature histograms. In *Proceedings of the 21st IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[119] T. T. Santos, L. V. Koenigkan, J. G. A. Barbedo, and G. C. Rodrigues. 3D plant modeling: Localization, mapping and segmentation for plant phenotyping using a single hand-held camera. In *Computer Vision - ECCV 2014 Workshops - Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part IV*, pages 247–263, 2014.

[120] T. T. Santos and G. C. Rodrigues. Flexible three-dimensional modeling of plants using low-resolution cameras and visual odometry. *Machine Vision and Applications*, 27(5):695–707, 2016.

[121] scanalyzer-HTS. `http://www.lemnatec.com/products/hardware-solutions/scanalyzer-hts/`, 2016.

[122] H. Scharr, M. Minervini, A. P. French, C. Klukas, D. M. Kramer, X. Liu, I. Luengo, J. Pape, G. Polder, D. Vukadinovic, X. Yin, and S. A. Tsaftaris. Leaf segmentation in plant phenotyping: a collation study. *Machine Vision and Applications*, 27(4):585–606, 2016.

[123] C. D. Scott and R. D. Nowak. Robust contour matching via the order-preserving assignment problem. *IEEE Transactions on Image Processing*, 15(7):1831–1838, 2006.

[124] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):550–571, 2004.

[125] W. Shen, Y. Jiang, W. Gao, D. Zeng, and X. Wang. Shape recognition by bag of skeleton-associated contour parts. *Pattern Recognition Letters*, 83:321–329, 2016.

[126] J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1270–1281, 2008.

[127] K. Simek, R. Palanivelu, and K. Barnard. Branching gaussian processes with applications to spatiotemporal reconstruction of 3d trees. In *In proc. of European Conference on Computer Vision (ECCV)*, 2016.

[128] T. Simonneau, R. Habib, J-P. Goutouly, and J-G. Huguet. Diurnal changes in stem diameter depend upon variations in water content: Direct evidence in peach trees. *Journal of Experimental Botany*, 44(3):615–621, 1992.

[129] D. Smeets, P. Bruyninckx, J. Keustermans, D. Vandermeulen, and P. Suetens. Robust matching of 3D lung vessel trees. In *MICCAI workshop on pulmonary image analysis*, 2010.

[130] O. J. O. Soderkvist. Computer vision classication of leaves from swedish trees. Master's thesis, Linkoping University, Sweden, 2001.

[131] S. Somayajula, A. A. Joshi, and R. M. Leahy. Non-rigid image registration using gaussian mixture models. In *Proceedings of the 5th international conference on Biomedical Image Registration*, 2012.

[132] R. Sozzani, W. Busch, E. P. Spalding, and P. N. Benfey. Advanced imaging techniques for the study of plant growth and development. *Trends in Plant Science*, 19(15), 2014.

[133] E. P. Spalding and N. D. Miller. Image analysis is driving a renaissance in growth measurement. *Current Opinion in Plant Biology*, 16(1):100–104, 2013.

[134] B. Steder, R.B. Rusu, K. Konolige, and W. Burgard. Point feature extraction on 3D range scans taking into account object boundaries. In *Proc. of ICRA*, 2011.

[135] Y. Su, Y. Liu, B. Cuan, and N. Zheng. Contour guided hierarchical model for shape matching. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[136] R. Subramanian, E. Spalding, and N. Ferrier. A high throughput robot system for machine vision based plant phenotype studies. *Machine Vision and Applications*, 24(3):619–636, 2013.

[137] N. Thakoor, J. Gao, and S. Jung. Hidden markov model based weighted likelihood discriminant for 2-d shape classification. *IEEE Transactions on Image Processing*, 16(11), 2007.

[138] TheArabidopsisGenomeInitiative. Analysis of the genome sequence of the flowering plant arabidopsis thaliana. *Nature*, 408(6814), 2000.

[139] J. H. M. Thornley. Modelling stem height and diameter growth in plants. *Annals of Botany*, 84:195–205, 1999.

[140] R. Toldo, A. Beinat, and F. Crosilla. Global registration of multiple point clouds embedding the generalized procrustes analysis into an ICP framework. In *Proceedings of 3DPVT'10*, May 2010.

[141] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *In proc. of 8th European Conference on Computer Vision (ECCV)*, 2004.

[142] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proc. of SIGGRAPH*, 1994.

[143] M. Uhercik, J. Kybic, H. Liebgott, and C. Cachard. Model fitting using RANSAC for surgical tool localization in 3D ultrasound images. *IEEE Transactions on Biomedical Engineering*, 57(8):1907–1916, 2010.

[144] VascuSynth. `http://vascusynth.cs.sfu.ca`.

[145] M. Wahabzada, S. Paulus, K. Kersting, and A-K. Mahlein. Automated interpretation of 3d laserscanned point clouds for plant organ segmentation. *BMC Bioinformatics*, 16:248, 2015.

[146] B. Wang, D. Brown, Y. Gao, and J. L. Salle. Mobile plant leaf identification using smart-phones. In *IEEE International Conference on Image Processing, (ICIP)*, 2013.

[147] B. Wang, D. Brown, Y. Gao, and J. L. Salle. MARCH: multiscale-arch-height description for mobile retrieval of leaf images. *Information Sciences*, 302:132–148, 2015.

[148] B. Wang and Y. Gao. Hierarchical string cuts: A translation, rotation, scale, and mirror invariant descriptor for fast shape retrieval. *IEEE Transactions on Image Processing*, 23(9):4101–4111, 2014.

[149] S. Wang, T. Kubota, and T. Richardson. Shape correspondence through landmark sliding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[150] Y. Wang and E. K. Teoh. 2D affine-invariant contour matching using b-spline model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1853–1858, 2007.

[151] S. G. Wu, F. S. Bao, E. Y. Xu, Y. Wang, Y. Chang, and Q. Xiang. A leaf recognition algorithm for plant classification using probabilistic neural network. In *IEEE International Symposium on Signal Processing and Information Technology*, 2007.

[152] M. Xia and B. Liu. Image registration by "Super-Curves". *IEEE Transactions on Image Processing*, 13(5):720–732, 2004.

[153] C. Xu, J. Liu, and X. Tang. 2D shape matching by contour flexibility. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):180–186, 2009.

[154] G. Xu, F. Zhang, S. G. Shah, Y. Ye, and H. Mao. Use of leaf color images to identify nitrogen and potassium deficient tomatoes. *Pattern Recognition Letters*, 32(11):1584–1590, 2011.

[155] J. Yang, H. Wang, J. Yuan, Y. Li, and J. Liu. Invariant multi-scale descriptor for shape representation, matching and retrieval. *Computer Vision and Image Understanding*, 145:43–58, 2016.

[156] Pu Yang. Quantitative 3d plant growth of arabidopsis thaliana using 3d range images. Master's thesis, Dept. of Computer Science, University of Western Ontario, 2010.

[157] X. Yang, H. Qiao, and Z. Liu. Partial correspondence based on subgraph matching. *Neurocomputing*, 122:193–197, 2013.

[158] A. L. Yuille and N. M. Grzywacz. A computational theory for the perception of coherent visual motion. *Nature*, 333(6168), 1988.

[159] C. Zhang and T. Chen. Efficient feature extraction for 2D/3D objects in mesh representation. In *IEEE International Conference on Image Processing (ICIP)*, pages 935–938, 2001.

[160] J. Zhang, Z. Huan, and W. Xiong. An adaptive gaussian mixture model for non-rigid image registration. *Journal of Mathematical Imaging and Vision*, 44(3):282–294, 2012.

[161] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision (IJCV)*, 13(2):119–152, 1994.

[162] C. Zhao, S. S. F. Chan, W. Cham, and L. M. Chu. Plant identification using leaf shapes - A pattern counting approach. *Pattern Recognition*, 48(10):3203–3215, 2015.

[163] Chen Zhao. 3d plant growth measurement using the shapegrabber laser scanner. Master's thesis, Dept. of Computer Science, University of Western Ontario, 2009.

[164] D. Zhao and J. Chen. Affine curve moment invariants for shape recognition. *Pattern Recognition*, 30(6):895 – 901, 1997.

[165] F. Zhao and R. Bhotika. Coronary artery tree tracking with robust junction detection in 3D CT angiography. In *Proceedings of the 8th IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 2066–2071, 2011.

[166] Z. Q. Zhao, Y. Hong, P. Zheng, and X. Wu. Plant identification using triangular representation based on salient points and margin points. In *IEEE International Conference on Image Processing (ICIP)*, 2015.

[167] B. Zhong, K. Ma, and J. Yang. Curvature scale-space of open curves: Theory and shape representation. In *IEEE International Conference on Image Processing (ICIP)*, 2013.

[168] F. Zhou and F. De la Torre. Deformable graph matching. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2922–2929, 2013.

[169] G. Zhou, B. Wang, and J. Zhou. Automatic registration of tree point clouds from terrestrial lidar scanning for reconstructing the ground scene of vegetated surfaces. *IEEE Geoscience and Remote Sensing Letters*, 11(9):1654–1658, 2014.

Curriculum Vitae

Publications:

1. A. Chaudhury and J.L. Barron, *"Partially Occluded Leaf Recognition via Subgraph Matching and Energy Optimization"*, IEEE Transactions on Image Processing (in review)

2. A. Chaudhury, C. Ward, A. Talasaz, A.G. Ivanov, M. Brophy, B. Grodzinski, R.V. Patel, N.P.A. Hüner and J.L. Barron, *"Machine Vision System for 3D plant phenotyping"*, IEEE/ACM Transactions on Computational Biology and Bioinformatics (in review).

3. A. Chaudhury, M. Brophy, and J.L. Barron, *"Junction Based Correspondence Estimation of Plant Point Cloud Data using Subgraph Matching"*, IEEE Geoscience and Remote Sensing Letters, Vol. 13, No. 8, pp. 1119-1123, August 2016.

4. A. Chaudhury, C. Ward, A. Talasaz, A.G. Ivanov, N.P.A. Hüner, B. Grodzinski, R.V. Patel, and J.L. Barron, *"Computer Vision Based Autonomous Robotic System for 3D Plant Growth Measurement"*, In Proceedings of the 12th Conference on Computer and Robot Vision (CRV), Halifax, Canada, 3-5 June 2015.

5. M. Brophy, A. Chaudhury, S. Beauchemin, and J.L. Barron, *"A method for global non-rigid registration of multiple thin structures"*, In Proceedings of the 12th Conference on Computer and Robot Vision (CRV), Halifax, Canada, 3-5 June 2015.

Name:	Ayan Chaudhury
Post-Secondary Education and Degrees:	PhD in Computer Science University of Western Ontario, Canada 2012-2017
	M.Tech in Computer Science and Engineering University of Calcutta, India 2010-2012
	Master of Computer Application University of Calcutta, India 2007-2010
	B.Sc. in Physics (Honours) Serampore College, University of Calcutta, India 2004-2007
Honours and Awards:	Best Teaching Assistant award University of Western Ontario, 2016
	Western Graduate Research Scholarship (WGRS) University of Western Ontario, 2012-16
	University of Calcutta Foreign Travel Scholarship, 2012-13
	Rank 6th in M.Tech. examination in University of Calcutta, 2012
	Rank 2nd in MCA examination in University of Calcutta, 2010
Related Work Experience:	Graduate Teaching Assistant The University of Western Ontario, 2012-2017
	Instructor for Computer Graphics course, The University of Western Ontario, Fall 2016
	Internship at Indian Statistical Institute, 2011-’12