Western Graduate&PostdoctoralStudies

Western University
Scholarship@Western

Electronic Thesis and Dissertation Repository

December 2015

Surface Reconstruction from Noisy and Sparse Data

Mark A. Brophy
The University of Western Ontario

Supervisor
Dr. John Barron
The University of Western Ontario

Joint Supervisor
Dr. Steven Beauchemin
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

© Mark A. Brophy 2015

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

Part of the Other Computer Sciences Commons

Recommended Citation

Brophy, Mark A., "Surface Reconstruction from Noisy and Sparse Data" (2015). *Electronic Thesis and Dissertation Repository*. 3375.
https://ir.lib.uwo.ca/etd/3375

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.

Surface Reconstruction

from

Noisy and Sparse Data

by

Mark Brophy

Faculty of Science
Department of Computer Science

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

© Mark Brophy 2015

Abstract

We introduce a set of algorithms for registering, filtering and measuring the similarity of unorganized 3D point clouds, usually obtained from multiple views.

We contribute a method for computing the similarity between point clouds that represent closed surfaces, specifically segmented tumours from CT scans. We obtain watertight surfaces and utilize volumetric overlap to determine similarity in a volumetric way. This similarity measure is used to quantify treatment variability based on target volume segmentation both prior to and following radiotherapy planning stages.

We also contribute an algorithm for the drift-free registration of thin, non-rigid scans, where *drift* is the build-up of error caused by sequential pairwise registration, which is the alignment of each scan to its neighbour. We construct an average scan using mutual nearest neighbours, each scan is registered to this average scan, after which we update the average scan and continue this process until convergence. The use case herein is for merging scans of plants from multiple views and registering vascular scans together.

Our final contribution is a method for filtering noisy point clouds, specifically those constructed from merged depth maps as obtained from a range scanner or multiple view stereo (MVS), applying techniques that have been utilized in finding outliers in clustered data, but not in MVS. We utilize kernel density estimation to obtain a probability density function over the space of observed points, utilizing variable bandwidths based on the nature of the neighbouring points, Mahalanobis and reachability distances that is more discriminative than a classical Mahalanobis distance-based metric.

Acknowledgements

A special thank you to my supervisor Dr. John Barron, you've been infinitely helpful, patient and caring. Having someone to learn from and bounce ideas off has been invaluable. You are simultaneously pragmatic and creative in your approach to problem solving. Likewise, thank you to my co-supervisor Dr. Steven Beauchemin, you gave me guidance and room to experiment from the very beginning, allowing me to explore and truly understand and own the process of my postgraduate studies. Penning publications with both of you has been an honour. Thank you to Spencer Martin and Ayan Chaudhury, you both have been fantastic collaborators.

Thank you to my partner Erika, you have always been loving, supportive and proud. Even at the lowest points of my research, you have never entertained the idea of giving up. Similarly, thank you to my mother and father, you have been incredible throughout this entire process. All three of you have overlooked my glaring shortcomings and helped me move forward on more occasions than I can count. Thank you to Dr. Paul Azzopardi, Sam Hinds, Jonathan Hornell-Kennedy and everyone at the London Western Track and Field Club for making London my home for many years.

Thank you to all of the great scientists at Western who have inspired me, but especially those in my area: in addition to supervisors John and Steven, thank you to Drs. Yuri Boykov, Olga Veksler and Mahmoud R. El-Sakka. You all continue to show me what ambitious, top notch vision research looks like.

Thank you to the members of my committee, Drs. John Zelek, Ken A. McIsaac, Mahmoud El-Sakka and Charles Ling: I know that your schedules are absurdly busy at this time of year and I appreciate your commitment to both my personal growth and academe at large.

Contents

ABSTRACT ii

ACKNOWLEDGEMENTS iii

CONTENTS iv

LIST OF TABLES vii

LIST OF FIGURES viii

1 Introduction 1
- 1.1 Contributions . . . 2
- 1.2 Publications . . . 3
- 1.3 Overview of Thesis . . . 4

2 Energy based surface reconstruction for point clouds 7
- 2.1 Preliminaries: Minimization, Smoothing and Segmentation . . 9
- 2.2 Level Sets and Snakes . . . 11
  - 2.2.1 Geodesic Active Contours . . . 12
- 2.3 Graph Cuts . . . 13
  - 2.3.1 Multiview Graph Cuts . . . 14
- 2.4 Total Variation . . . 16
  - 2.4.1 Total Variation Surface Reconstruction . . . 17
- 2.5 Chan-Vese: Active Contours without Edges . . . 18
  - 2.5.1 Shape Priors in Level Set Formulations . . . 19
- 2.6 Segmentation of Point Clouds with Level Sets . . . 21
- 2.7 Flux-based Functionals . . . 21
  - 2.7.1 Continuous Global Optimization . . . 24
  - 2.7.2 Global Minimization for Shape Fitting . . . 25
  - 2.7.3 Power Watersheds . . . 25
- 2.8 Summary of Methods . . . 27

3 **A Method for Quantitative Comparison of 4D-CT Lung Tumour Segmentations 28**
3.1 Introduction 28
3.2 Background 30
3.3 Problem 31
3.4 Methodology 32
3.5 Point Cloud Construction 35
  3.5.1 Surface Fitting 36
  3.5.2 Computing Normals 38
  3.5.3 Minimal Surface Fitting to Segmented Point Cloud . . 39
3.6 Similarity Quantification 41
3.7 Results 42
3.8 Conclusions and Future Work 47

4 **Global Registration of Multiple Thin Structures 50**
4.1 Previous Work 54
4.2 Proposed Method 57
  4.2.1 Approximate Alignment 60
  4.2.2 Global Non-Rigid Registration via MNN 60
    Global Registration 60
4.3 Results 61
  4.3.1 Synthetic Vascular Data 61
  4.3.2 Plant Data 62
4.4 Conclusions and Future Work 69

5 **Kernel Density Filtering for Noisy Point Clouds in One Step 70**
5.1 Introduction 71
5.2 Previous Work 73
5.3 Obtaining a Probability Density Function from Measured Data 77
  5.3.1 Kernel Density Functions 78
  5.3.2 Mahalanobis Distance 78
  5.3.3 Local Bandwidth Estimation 80
5.4 Methodology 80
  5.4.1 Reachability Distance 81
5.5 Kernel Density Filtering on Data with Additive Noise 82
5.6 Results 83
5.7 Conclusions and Future Work 88

6 Conclusions and Future Work **90**

6.1 Conclusions 90

6.2 Future Work 91

BIBLIOGRAPHY **92**

VITA **100**

List of Tables

3.1 Coefficient of Variation for VOE. 47

4.1 Quantitative results for different algorithms and data sets. . . 57

5.1 Confusion matrix describing the different classifications of inliers and outliers. 83

List of Figures

2.1 Level set segmentation of a 2D point cloud. 22
2.2 Level set segmentation of a 3D point cloud of a cube. 23

3.1 Two different views of the same segmented primary GTV. Note the top of the surface is flat and this made the volume fitting more challenging because the flux width has to be wider to ensure that the entire surface is contained. The flat top is visible in (a). 33
3.2 Slices of physician provided segmentations for patient E. The GTVs on different axes are shown at top left, top right and bottom left, while the bottom right image shows the superimposed 3D meshes for all segmentations. Top left shows a slice at $z = 52.7$, top right shows slice $x = 5.62$ and bottom left shows slice $y = -18.485$. 34
3.3 Point cloud extracted from physician segmentations. 35
3.4 Approximate meshed surface for point cloud seen in Figure 3.3, as fitted using the ball pivoting algorithm. 37
3.5 Normals calculated for an approximate surface. 38
3.6 Three different segmentations of patient 3's nodal tumour by different physicians. 43
3.7 Six segmentations of patient 3's primary tumour by different physicians. 44
3.8 Graph of volumetric overlap for gated tumours for each patient. 45
3.9 Graph of RMS distance for gated tumours for each patient. . . 45
3.10 Slices of physician provided segmentations for patient B GTV on different axes are shown at top left, top right and bottom left, while bottom right shows the superimposed 3D meshes for all segmentations. Top left shows a slice at $z = 53.400$, top right shows slice $x = -5.313$ and bottom bottom left shows a slice at $y = -17.808$. 49

4.1 A subset of the views of the synthetic vascular data. We take one scan and then apply a set of random, non-rigid deformations to obtain our data set. 52
4.2 Pairwise registration results for two scans of the Stanford Bunny. Top row: original two views, Bottom row: results obtained using Rusu et al. (FPFH) [61], Fitzgibbon (LMICP) [23], Jian and Vemuri (GMMReg) [33] and Myronenko and Song (CPD) [49]. 57
4.3 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. [61], Fitzgibbon [23] and Myronenko and Song [49]. 58
4.4 12 scans of the Arabidopsis plant, prior to registration, but with rotation and and translation pre-applied. 64
4.5 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. 65
4.6 20 synthetic scans of vascular data merged into a single point cloud using MNN, as seen from two viewpoints, front and back. 66
4.7 MNN versus sequential pairwise registration on vascular data, registering the first scan to each of the subsequent scans. . . . 67
4.8 MNN versus sequential pairwise registration on the plant data. 68

5.1 The *Buddha* point cloud with 5:1 ratio of noise to inliers added. 75
5.2 The *Bunny* point cloud with 5:1 ratio of noise to inliers added. 76
5.3 ROC curve for anisotropic filters on the *Bunny* point cloud with 5:1 noise. The reachability distance/adaptive method is in red. 84
5.4 ROC curve for anisotropic filters on the *Buddha* point cloud with 5:1 noise. The reachability distance/adaptive method is in red. 84
5.5 Buddha, filtered. The reachability distance-based filtering method leaves almost no outliers. 85
5.6 The *Bunny* point cloud with 5:1 noise, filtered. 87

Chapter 1

Introduction

This thesis began its life as a theoretical work on capturing the shape of 3D objects. To test the surface fitting method that I was working on, I began constructing point cloud data by sampling spheres and cubes. I constructed an approximate initial surface, evolved the shape of this surface until it fit the points well, and recovered a good facsimile of the originally sampled surface.

Two problems arose. First, the process of evolving the surfaces requires high precision. The surface could only be evolved extremely slowly due to the numerical instability exhibited by the method when making large jumps in the solution space. Second, surface evolution methods in general are notoriously difficult to minimize when there is noise present. Any sort of perturbation of points or the presence of outliers impacted negatively on the ability of the method to converge.

As I tried to utilize surface evolution based methods on real problems, these limitations became abundantly clear. The results from the depth maps that I reconstructed from multiple images and attempted to mesh were poor due to the prominence of outliers. Results were extremely reliant on parameterization, with the surface "leaking" through gaps between points with excessive spacing.

Similarly, my attempts to register depth maps and unorganized point clouds using surface-based methods were filled with folly. Capturing the shape of point clouds using statistical methods is the basis of this thesis. Whether it is fitting watertight surfaces to tumours and quantifying their similarity, registering multiple point clouds into a single point cloud, or explicitly removing outliers from an extremely noisy point cloud prior to meshing, the consistent theme is that a good model is required to solve real world problems.

This thesis suggests some algorithms to help ameliorate the inherent problems surrounding shape recovery for both static and moving scenes and objects. We deal with the recovery of non-rigid shape from both partial range scans from multiple views and segmented point clouds from MRI data, as well as the Stanford Bunny [71] and the Happy Buddha [39] data sets.

1.1 Contributions

The contributions of this thesis are as follows:

- A global method [10, 15] for non-rigid registration of multiple thin 3D structures from range scans. Its efficacy is demonstrated on both synthetic vascular data and real scans of a mature Arabidopsis plant. The positions of both the leaves and the branches change from scan to scan (as the plant rotates on a turntable underneath a ceiling fan). Our global method utilizes the pair based algorithm of these scans based on the Coherent Point Drift (CPD) algorithm of Myronenko and Song [49].
- A novel method [46, 45] for quantifying physician performance in seg-

menting tumours from MRI data. For each tumour, we construct an approximate surface to obtain normals, we use Boykov and Lempitsky's [38] flux-based method for obtaining watertight surfaces, and then we compare these surfaces using volumetric overlap estimation (VOE).

- A method [9] for filtering point clouds using variable bandwidth in conjunction with an anisotropic kernel density filter. We filter noisy point clouds generated from merged, noisy depth maps in a single step, allowing us to fit a surface to the resultant point cloud. This algorithm is useful for removing outliers from range scans and other noisy 3D data, and can work as a useful preprocessing step before registration or surface fitting.

All work herein was performed by the author, with the exception of the pair based registrations of the Bunny and Arabidopsis, as seen in Figures 4.2, 4.3 and Table 4.1 (performed by Ayan Chaudhury) as well as the statistical analyses seen in Figures 3.8, 3.9 and Table 3.1 (performed by Spencer Martin.)

1.2 Publications

Three journal papers (one is in submission), along with two conference papers have been published based on this work.

1. M.A. 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. IEEE, June 2015.

2. M.A. Brophy, S.S. Beauchemin, and J.L. Barron. Kernel density filtering

for noisy point clouds in one step. In *Proceedings of the Irish Machine Vision and Image Processing Conference*, pages 27 – 34. August 2015.

3. A. Chaudhury, M.A. Brophy, and J.L. Barron. Junction point detection in 3D plant point clouds using multi-modal analysis and tls approximation. In *Submitted*, August 25, 2015.

4. S. Martin, C. Johnson, M.A. Brophy, D.A. Palma, J.L. Barron, S.S. Beauchemin, A.V. Louie, E. Yu, B. Yaremko, B. Ahmad, G.B. Rodrigues, and S. Gaede. Impact of target volume segmentation accuracy and variability on treatment planning for 4d-ct-based non-small cell lung cancer radiotherapy. *Acta Oncologica*, 54(3): 322 – 332, March 2015.

5. S. Martin, M. Brophy, G. Rodrigues, D. Palma, A. Louie, B. Yaremko, B.A.E. Yu, S.S. Beauchemin, J.L. Barron, and S. Gaede. A proposed framework for consensus-based lung tumour volume auto-segmentation in four- dimensional computed tomography imaging. *Physics in Medicine and Biology*, 60(4):1497 - 1518, February 2015.

1.3 Overview of Thesis

This thesis deals with algorithms for constructing meshed 3D models of objects from various data sources, be it from range scans, multiple 2D images, physician-segmented CT data, etc.

The following chapters are unified in their application to unorganized point clouds. Chapter 2 provides a review of energy-based methods for fitting surfaces to point clouds, while the chapters that follow are largely for registering, filtering and comparing point clouds.

In Chapter 3, we describe our method for comparison of pairs of unorganized point clouds, specifically for the comparison of tumour volumes for the purpose of quantifying physician performance in segmenting tumours from CT scans. We utilize Boykov and Lempitsky's [38] method for obtaining watertight meshes for the tumour volumes, and perform both point-based (mutual nearest neighbours) and volumetric (Jaccard index) to quantify the variability of a set of physicians' performance of lung tumour volumes over the various breathing phases.

In Chapter 4, we present a global algorithm for drift-free alignment of multiple range scans of "thin" data into a single point cloud (suitable for further processing, such as triangular meshing and volume calculation). We consider two sets of non-rigid data: synthetic vascular data and real Arabidopsis plant data. Our method builds on the coherent point drift algorithm, and aligns multiple point clouds into a single 3D point cloud. The plant data was acquired in a growth chamber, where the fan caused jittering in both the branch and leaf data. For each scan, we construct a target scan from the centroids of its Mutual Nearest Neighbours (MNN) in all other scans and iteratively register to this, as opposed to registering pairwise scans sequentially. We have have adapted MNN for use in non-rigid scenarios, producing a method that will not degrade as more scans are registered, and produces better results than sequential pairwise registration.

Chapter 5 describes our method for filtering very noisy point clouds. We utilize techniques that have roots in clustering data and apply it to the filtering of point clouds. The probability density function (PDF) is approximated via a method called kernel density estimation (KDE). Our filter uses Mahalanobis distance, variable bandwidth and reachability distance, which gives more discriminative power than a strictly Mahalanobis distance-based KDE filter.

Finally, Chapter 6 presents the conclusions of the thesis and gives future directions of research.

Chapter 2

Energy based surface reconstruction for point clouds

With modern range sensor technology, the ability to capture high quality structural information has never been easier: obtaining an accurate depth map for almost any opaque object is as simple as pushing a button as range scanners and high quality stereo algorithms are available to the masses.

Despite this, many of the problems surrounding the accurate fitting of a surface to point cloud and merging multiple point clouds into a single scan remain difficult. Problems in this domain are often ill-posed and thus we must provide additional constraints on our search domain in order to find a suitable solution.

This chapter gives an overview of methods for reconstructing 3D object models from unorganized point clouds, initially born out of a desire to solve this problem elegantly using evolving surfaces guided by partial differential equations. Most of our work is both discrete and statistical in nature, using techniques that rely on graph cuts, expectation maximization, etc. We review both earlier work and modern techniques.

As we mentioned above, most of the problems we solve are ill-posed, and thus we need to understand exactly what this means and the ramifications that this will have on our work. We begin by reviewing the fundamentals of energy-based methods before we deal with the details of surface fitting. Discrete, graph cut-based methods and continuous total variation methods are the state of the art. However the utilization of flux, curvature and shape priors that began with level sets is also illuminating.

Though this is a review of 3D methods, we'll often introduce and illustrate methods via 2D image segmentation, as this is where this work is rooted. In short, though 2D and 3D (MRI or CT, for example) image segmentation involves segmenting dense intensity data instead of sparse binary data, the techniques for performing either one have influenced techniques for performing the other. In short, understanding image segmentation is core to obtaining a deeper understanding of surface fitting,

2.1 Preliminaries: Minimization, Smoothing and Segmentation

Many problems in Computer Vision are intricately tied: an optimal solution of one problem often provides meaningful insight into other seemingly unrelated problems. Most vision problems of interest are *ill-posed*, and thus require some sort of restriction in terms of the solution space of its parameters in order to find a good solution. This is true of fitting a surface to a scattered set of points [3], where there are an infinite number of surfaces that fit said points [27]. As shown in the sections that follow, these constraints often involve finding optimally denoised, diffused or smoothed regions to avoid getting stuck in local minima while propagating a contour. One desires the solution space of a function to be strict in terms of its shape: Ideally, a function is *monotonically* or *strictly* increasing, resulting in a *convex* or *semi-convex* space to search for the minimum.

When a solution space u is convex or semi-convex, one can guarantee a global minimum will be obtained via gradient descent. In one dimension, such a solution space will resemble a parabola while in two dimensions the solution space resembles a paraboloid. The direction of the step is the one of *steepest descent*, so such a scheme is guaranteed to find a global minimum since the "steepest" path will eventually lead to a minimal value. This method of energy minimization is extremely common in Computer Vision and can be found using *level sets*. Level sets are the subject of Section 2.2.

The idea behind the piecewise-smooth Mumford-Shah functional [48] was to provide both a smooth approximation u of an image I, and a partitioning that divided the image along its discontinuous areas [11]. It's an intuitive notion: One imagines that the segmentation of an image that obtains the maximal smoothness for each area would result in very good separation of objects that exist in the image. The functional, given by Brox and Cremers [11]:

$$E(u,K) = \int_{\Omega} (u - I)^2 dx + \lambda \int_{\Omega - K} |\nabla|\nabla u|^2 \mathbf{dx} + v|K| \tag{2.1}$$

operates on the image $I : \Omega \to R$ with domain $\Omega \subset \mathbb{R}^2$ and seeks the edge set K that partitions function u into smooth sections, where u is an approximation of I. Thus, K segments Ω into n (which is unknown prior to minimization) disjoint regions, $\Omega_{1 \cdots n}$. The functional seeks both the optimal u and the optimal K. $v \geq 0$ and $\lambda \geq 0$ are weighting parameters.

As pointed out by Brox and Cremers [11], Mumford and Shah [48] wrote nearly 100 pages of theory without ever suggesting how one would actually find a minimizer of Equation (2.1) or its simplified (piecewise-constant) form, the latter of which is obtained when $\lambda \to \infty$ as:

$$E(u,K) = \sum_i \int_{\Omega_i} (u_i - I)^2 dx + v_o|K|, \tag{2.2}$$

where v_0 is the re-scaled version of v, and each u_i is now a single value, as opposed to in Equation (2.1), where it was a function. In other words, the image is approximated by a piecewise constant function, as opposed to a piecewise smooth one. This is why it is referred to as the *cartoon limit*. It has been the focus of a large body of work that has followed.

2.2 Level Sets and Snakes

Kass et al.'s [36] seminal paper on snakes in the late 80's served as a source of inspiration for a huge body of work that includes geodesic active contours, active contours without edges, etc. Variational methods like snakes are still applied to problems in image segmentation, stereo, multiview reconstruction, and more recently to multi-region segmentation.

The original formulation of snakes evolved a contour according to a combination of image forces and external constraint forces

$$E(C) = \alpha \int_0^1 |C'(q)^2 \, dq + \beta \int_0^1 |C''(q)|^2 \, dq - \lambda \int_0^1 |\nabla I(C(q))| dq \tag{2.3}$$

where C is a curve that is parameterized by q, yielding a point in R^2. In the original implementation, a term was included to allow for user interactivity to avoid the contours tendency to get stuck in local minima. The first two terms (α and β) control the first- and second-order smoothness of the contour's internal energy. The first-order term makes the snake behave like a "membrane", while the second-order term makes it act like a "thin plate" [36]. The attractiveness of the contour to edges (the external energy) is controlled by λ. This external force ensures that a local minimum is found at an image edge, while the internal force ensures piecewise smoothness of the contour.

This energy is then converted to its Euler-Lagrange equation and solved via gradient descent. The success of such a scheme is highly dependent on a good initialization, however. As was noted by Kass et al. [36], the contour must be placed near the solution to avoid getting stuck in local minima.

2.2.1 Geodesic Active Contours

Caselles et al.'s [14] scheme for image segmentation, while inspired by snakes, was based on curve evolution via mean curvature and geometric flows, not energy minimization. Their method involved contour propagation (deformation) according to a velocity function defined in terms of curve regularity and closeness to a boundary. By embedding an evolving contour C into a level set formulation, the representation of C is both implicit and parameter free. Further, different topologies can be captured by the zero level set, and thus contours are able to split and join without requiring any sort of heuristics, a definite improvement on the original snakes implementation. Multi-object segmentation may be performed without knowing the number of objects in advance.

Caselles et al.'s functional:

$$E_{GAC}(C) = \int_0^{L(C)} g(|\nabla I_0(C(s))|)ds, \tag{2.4}$$

the so-called *geodesic active contour*, removes the parameterization of the curve C that was present in Equation (2.3). It is replaced with the total Euclidean length of the curve, $L(C)$, as defined by the Euclidean element of length, ds. The boundary function:

$$g(|\nabla I_0|) = \frac{1}{1 + \beta|\nabla I_0|^2}, \tag{2.5}$$

is essentially an edge detector with a weighting function β. The corresponding Euler-Lagrange equation allows one to utilize gradient descent to find a minimizer of Equation (2.4):

$$\partial_t C = (\kappa g - \langle \nabla g, \mathcal{N} \rangle)\mathcal{N},$$

or in level set form as:

$$\partial_t \phi = \left(\kappa g + \langle \nabla g, \frac{\nabla \phi}{|\nabla \phi|} \rangle \right), \tag{2.6}$$

where the evolving curve C is embedded as the zero level set of ϕ such that $C(t) = \{x \in \mathbb{R}^N | \phi(x,t) = 0\}$.

2.3 Graph Cuts

For many vision problems that can be approximated by level sets, graph cuts can find a global minimum for many binary labeling problems as well as a good approximation for multi-labeling problems. Instead of evolving a surface based on forces related to topology, weighted by the data, solving problems with graph cuts instead involves solving problems by finding a way to embed them on an appropriate graph, performing various modifications and then finding a minimum cut along the edges of the graph. Graph cuts came to prominence for image segmentation, specifically for separating a foreground object from its background. This is a classic binary labeling problem, solving the same problem as Chan-Vese did in Equation (2.15).

A graph is built with two types of links, t-links (to terminal nodes) and n-links (to neighbouring nodes). The nodes typically represent a discrete grid. In the case of an image, each pixel would have a node, and it would have n-links to its neighbouring pixels/nodes, possibly the neighbours of its neighbours, etc. A node will also have a t-link to either an s- or t-node, i.e., a source or a sink. The assignment of these nodes depends on initialization strategy, and will change based on the strong moves that are performed during optimization.

Prior to finding the max-flow/min-cut, the labels for nodes are found using α-expansion and α–β-swap.

In iterative α-expansion, any subset of pixels can be changed to a label α that is fixed; the algorithm goes through each possible label until no more improvements can be made. At each step, a region P_α can only expand. α-expansion allows, for a selected label α to change any node's label to α, so long as it decreases the total energy of the min-cut. Likewise, a α–β-swap allows for any two labels to be swapped so long as they decrease the total energy of the min cut.

Graph cuts energies typically have both a data and a smoothness term. A data term might be intensity if the graph represents an image, or a distance function if a set of points has been discretized and embedded on a graph for the purpose of segmentation. If the smoothness term is metric then we use α-expansion, otherwise, we use α–β-swap.

The smoothness term is specifically for penalizing discontinuities between neighbouring points. In other words, it penalizes differing labels that neighbour one another.

2.3.1 Multiview Graph Cuts

Vogiatzis et al. [74] construct a dense graph from photo-consistency values, which is different from fitting surfaces to point clouds, but They use graph cuts to find the minimum of a dense grid where each node represents the centre of a voxel. The weights between two nodes is

$$w_{ij} = \frac{4\pi h^2}{3}\rho\left(\frac{x_i + x_j}{2}\right) \tag{2.7}$$

where x_i and x_j are connected voxels, and $\rho(\cdot)$ is a photo-consistency / matching function. A ballooning force $w_b = \lambda h^3$ connects each point to the sink, while the voxels on the perimeter of the volume have value ∞.

Sormann et al. [66] take a set of depth maps and construct a signed distance function for each, which are then combined. They dilate the zero level set to form a watertight “crust” band, and calculate the confidence at each voxel from an unsigned distance function using the method of Hornung and Kobbert [29], who find the minimum cut of the unsigned distance function around a “dilated” crust of input points, the output of which is a closed / watertight surface. They perform volumetric diffusion on the unsigned distance function. The use of unsigned distance removes the need for orientation data and is more robust to noise. The points are placed / rasterized on the grid and then an unsigned dilation operator is utilized.

The diffusion simply happens iteratively over a 6-neighbourhood until all non-empty voxels v on the grid are 6-connected.

$$\phi(v) = \frac{1}{|N(v)| + 1}\left(\phi(v) + \Sigma_{u \in N(v)}\phi(u)\right), \tag{2.8}$$

with the constraint that $\phi(v)$ remains zero for all initial sparse surface points. The number of diffusion steps is not overly important, in fact, even with no diffusion steps, a min cut on ϕ will yield a surface.

2.4 Total Variation

Total variation, for a continuous signal is computed as integral of the absolute gradient, or for a discrete signal, as $V(u) = \sum_n |\nabla u|$. Total variation lessens oscillations, but allows for discontinuities. It is used as a regularization term.

As was noted above, the success of the geodesic active contour model is also contingent on a good initialization due to the presence of local minima in the solution space. Bresson et al. [24] unified the Rudin, Osher and Fatemi (ROF) model

$$E_{ROF}(u, \lambda) = \int_\Omega |\nabla u| dx + \lambda \int_\Omega (u - f)^2 dx, \tag{2.9}$$

with the geodesic active contour model [7]. The ROF model denoises images while preserving edges. Instead of embedding the contour as the zero level set of a higher-order function, they instead chose to iteratively minimize an energy u and segment the result using some threshold $\mu \in [0, 1]$, typically chosen to be $\mu > 0.5$.

Bresson et al. replaced the total variation norm in Equation (2.9) with their own *weighted* total variation norm $TV_g(u)$, with the previously-defined weight function, $g(x)$. This weighted norm

$$TV_g(u) = g \int_\Omega |\nabla u| dx \tag{2.10}$$

introduces a link between the ROF model and the geodesic active contour functional from Equation (2.4). When one sets the input u of the weighted total variation norm to be a characteristic function, one obtains the geodesic active contour

$$TV_g(u = 1_{\Omega_C}) = \int_\Omega g(x)|\nabla 1_{\Omega_C}|dx \tag{2.11}$$

$$= \int_C g(x)ds = E_{GAC}(C). \tag{2.12}$$

Note that the gradient of the characteristic function is the boundary of the closed region, $\partial\Omega$. Thus, the only values integrated over are the diffusivities g values on the boundary C, $\partial\Omega$.

They also replaced the L^2-norm with the L^1-norm, noting that in the regularization process, the disappearing of features is determined entirely in terms of geometric characteristics, specifically the area and length of the contours and not contrast (as it is when using the L^2-norm).

Their non-strictly convex energy functional

$$E_{GGAC}(u, \lambda) = \int_\Omega g(x)|\nabla u|dx + \lambda \int_\Omega |u - f|dx,$$

is very similar to the ROF model. Using the calculus of variations, one obtains a partial differential equation

$$\frac{\partial u}{\partial t} = g\text{div}\left(\frac{\nabla u}{|\nabla u|}\right) + < \nabla g, \frac{\nabla u}{|\nabla u|} > +\lambda\frac{u - f}{|u - f|},$$

from which the resultant energy u is thresholded, and a global minimum is obtained. This is proved in Bresson et al.'s [7] appendix.

2.4.1 Total Variation Surface Reconstruction

In addition to creating a solver for the Chan Vese functional, Jian et al. [34] also introduce a method for fitting 3D surfaces to point clouds.

Zach et al. [76] utilize the total variation (TV) $|\nabla u|$ introduced by Rudin et al. [24] on robust range image integration. They minimize an energy functional consisting of both the TV term for regularization and an L1 data term for data fidelity. Their energy function is fairly straightforward

$$E = \int \left\{ |\nabla u| + \lambda \Sigma_{i \in I(x)} w_i |u - f_i| \right\} dx. \qquad (2.13)$$

2.5 Chan-Vese: Active Contours without Edges

Chan and Vese [73] proposed to solve the segmentation problem posed by the piecewise-constant Mumford-Shah functional by further simplifying the problem to a two-phase segmentation. While active contours rely heavily on a strong gradient separating the foreground and background of the image, Chan and Vese find a boundary that minimizes both the interior pixels' variance from the interior mean, and the exterior pixels' variance from the exterior mean. To define these regions, they utilize the Heaviside function

$$H(\phi) = \begin{cases} 1 & \phi \geq 0 \\ 0 & \phi < 0 \end{cases} \qquad (2.14)$$

Contour C is embedded as the zero level set of a 3D function ϕ. Their functional

$$E_{CV}(\phi, c_1, c2) = \int_{\Omega} \{H_\epsilon(\phi)(I(w) - c_1)^2 + (1 - H_\epsilon(\phi))(I(w) - c_2)^2 + \lambda|\nabla H_\epsilon(\phi)|\} dw, \qquad (2.15)$$

attempts to find the best approximation of I as a pair of unconnected regions with only two values, with respect to the L^2 norm [7]. c_1 and c_2 represent the mean values of I in the interior and exterior, and are formally defined as

$$c_1 = \frac{\int I(x)H(\phi)dx}{\int H(\phi)dx} \tag{2.16}$$

$$c_2 = \frac{\int I(x)(1 - H(\phi))dx}{\int (1 - H(\phi))dx}. \tag{2.17}$$

In practise, both the Heaviside function and the Dirac function are regularized [73]. Using these regularizations in Equation (2.15) and computing the subsequent Euler-Lagrange equations results in a PDE

$$\begin{aligned}
\frac{\partial \phi}{\partial t} &= \delta_\varepsilon(\phi)(\lambda_{in}(I(x) - c_in)^2 - \lambda_{out}(I(x) - c_{out})^2 + \mu) \text{ in } \Omega \\
\frac{\partial \phi}{\partial n} &= 0 \text{ on } \partial\Omega \\
\phi &= \phi_0 \text{ in } \Omega,
\end{aligned}$$

where $\frac{\partial \phi}{\partial n}$ is the partial derivative of ϕ in the direction of the outward normal to the boundary $\partial\Omega$.

Chan Vese suffers from the fact that it is highly dependent on its initialization. Like snakes, the model is not convex – it suffers from being a two valued function [26].

2.5.1 Shape Priors in Level Set Formulations

When most objects are projected onto a 2D plane, their resultant $2d$ boundaries vary depending on the orientation of the plane with respect to the object. Therefore, when using a shape prior, it is necessary to estimate the projective transformation parameters along with the best fit of the shape prior.

Rousson and Paragios [58] proposed embedding a shape prior into a level set formulation by generating a signed distance function from a pre-defined 2D contour. They defined *equivalence* between two objects of equal shape by their contours $\Omega_1, \Omega_2 \subset \mathbb{R}^2$ by their signed distance functions ϕ_1, ϕ_2

$$\phi_2(x,y) = r\phi_1\left[\frac{(x-a)\cos\theta + (y-b)\sin\theta}{r}, \frac{-(x-a)\sin\theta + (y-b)\cos\theta}{r}\right],$$

noting a unique solution to $|\nabla\phi| = 1$, constrained by

$$\phi = \begin{cases} > 0 & x \in \Omega \backslash \partial\Omega \\ = 0 & x \in \partial\Omega \\ < 0 & x \in R^2 \backslash \Omega \end{cases}, \tag{2.18}$$

where (a, b) is the centre of ϕ_2, and r and θ are the scaling factor and angle of rotation, respectively. One can generate an object's equivalent shapes by choosing different values for (a, b, r, θ).

Chan and Zhu added a shape prior to the Chan-Vese functional (2.15)

$$E_{CV+SP}(\phi) = E_{CV}(\phi) + E_{shape}(\phi),$$

where

$$E_{shape}(\phi) = \int_\Omega (H(\phi) - H(\psi_0))^2 dydx.$$

Instead of only iteratively solving for $\frac{\partial\phi}{\partial t}$, one iterates back and forth between solving for $\frac{\partial\phi}{\partial t}$ and the gradient descents for the shape priors transformational parameters: translational ($\frac{\partial a}{\partial t}$, $\frac{b}{\partial t}$), scale ($\frac{\partial r}{\partial t}$), and rotational ($\frac{\partial\theta}{\partial t}$).

2.6 Segmentation of Point Clouds with Level Sets

Once a set of disparity maps have been calculated and perspectively projected into a 3D volume, a surface can be evolved to fit the points using the convection method of Zhao et al. [79]. This is demonstrated on a sparse 2D point cloud in Figure 2.1, although the evolution equation used:

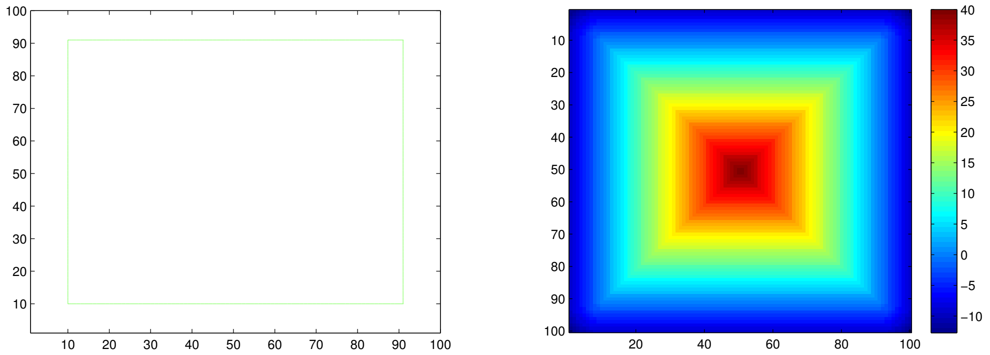
$$\frac{\partial \phi}{\partial t} = \nabla d(\mathbf{x}) \cdot \nabla \phi + d(\mathbf{x}) \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}, \tag{2.19}$$

is different than the one used by Zhao et al. [79] in that it incorporates mean curvature in the regularization.

2.7 Flux-based Functionals

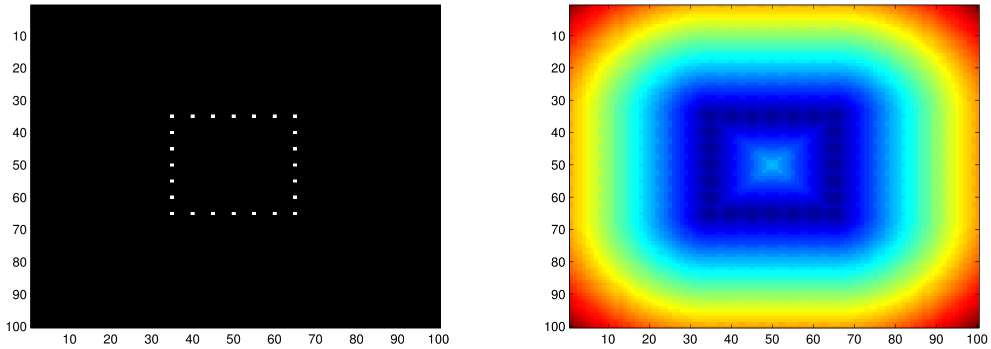
Unlike Zhao et al. [79] who utilize distance from sparse surface points, Savadjiev et al. [62] reconstruct the surface by utilizing a field of vectors that are normal to the surface. They assume that for the points in a small neighbourhood there exists a surface that the points have been sampled from. They find normals and smooth them based on local curvature, and then solve for the flux. Flux takes a vector field as input and produces a scalar which is essentially the amount of flow through a unit area of a surface. A minimizer for the functional

$$flux(t) = \int_0^1 < V, N > ||C_p|| dp = \int_0^{L(t)} < V, N > ds \tag{2.20}$$



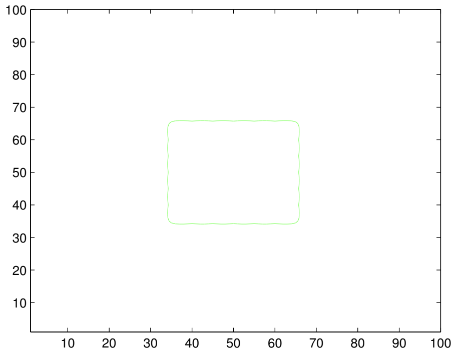
(a) Initial contour.

(b) Initial level set function.



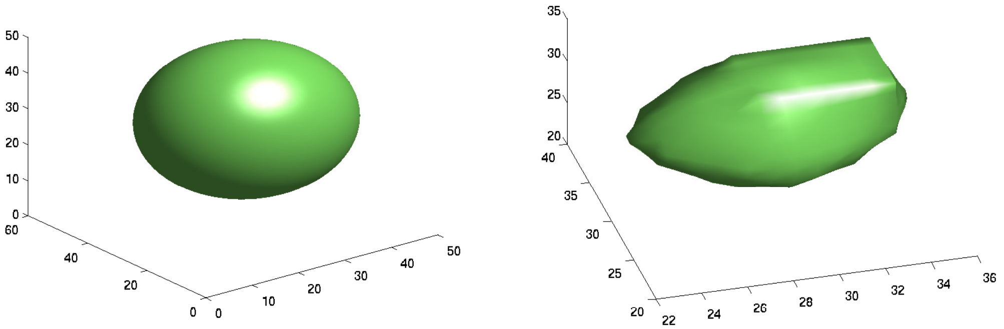
(c) Point cloud to segment (S).

(d) Distance function to S.



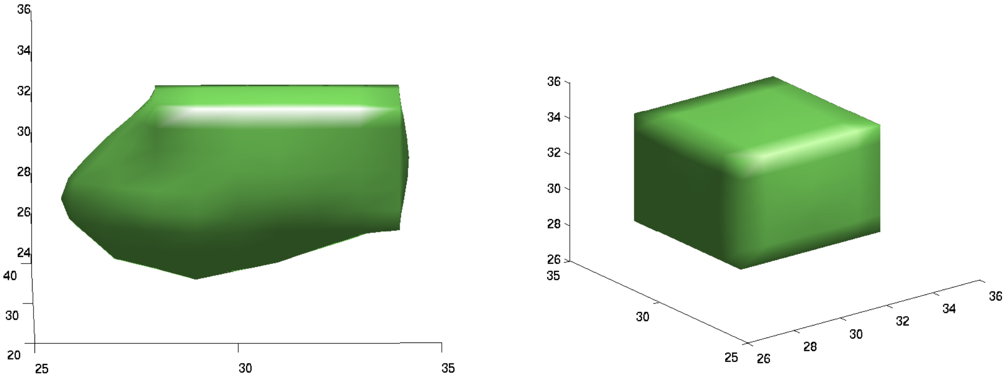
(e) Recovered contour.

Figure 2.1: Level set segmentation of a 2D point cloud.



(a) Initial contour.

(b) After 600 iterations.



(c) After 700 iterations.

(d) After 800 iterations

Figure 2.2: Level set segmentation of a 3D point cloud of a cube.

is found using level sets as described in 2.2. The normal vector field of a level set ϕ is $N = \delta\phi/|\delta\phi|$. The point cloud to which we wish to fit a surface has normal information for each point, but a dense field is required to evolve the level set. A box filter is used obtain F but propagate smoothed normal information. F drives surface evolution towards the points

$$\phi_t + F\phi = 0. \tag{2.21}$$

The idea of using flux for segmentation was introduced by Vasilevskiy and Siddiqi [72] and applied to 2D and 3D MRA images of blood vessels. They take an image I and compute the flux by using its gradient ∇I as the vector field V, wishing to maximize the flux through its boundary, which is either a curve or a surface.

2.7.1 Continuous Global Optimization

Pan and Skala [52] start with a set of points with weak orientation data (like [38]) and use total variation to measure the smoothness of function

$$E_s = \int\int\int_S |\nabla f(x, y, z)| dxdydz \tag{2.22}$$

where $|\cdot|$ is the L_1 norm. A smaller norm indicates a more smooth function, which a larger norm means the space is less smooth.

They use flux for their data term. A vector field F approximates the normal directions via a smoothing filter. They design E_d such that the gradient of the implicit function is best aligned with this vector field

$$E_d = \int_\Omega < \nabla f, F > \tag{2.23}$$

i.e. the optimal dot product between ∇f and F is zero.

2.7.2 Global Minimization for Shape Fitting

Lempitsky and Boykov [38] build on both graph cuts and Savadjiev et al.'s [62] functional. They choose to solve for a sparse set of normals where only weak orientation data is necessary. They too utilize a flux-based functional

$$max \overset{S}{\leftarrow} \int_S < v_s, n_s > ds, \tag{2.24}$$

though they solve for a thin band around the discretized points on the grid, which is expanded as as needed using their *TouchExpand* [38] algorithm.

Neighbouring nodes are connected via n-links, these represent the regularizing term. The data term is represented by t-links, where nodes with potentials greater than zero $U(p) > 0$ are connected to the source and nodes with potentials less than zero $U(p) < 0$ are connected to the sink.

2.7.3 Power Watersheds

Couprie et al. [20] also use a graph based solution to find a solution to the total variation problem mentioned above, specifically looking for the solution u to

$$\min_{u \in [0,1]} \int_\Omega w(z)^p |\nabla u(z)|^q dz, \tag{2.25}$$

subject to $u(z) = 0 \forall z \in \Omega_{in}$ and $u(z) = 1 \forall z \in \Omega_{out}$, where Ω_{in} is the set of labels inside of the surface and Ω_{out} is the outside set. They construct a graph G of edges (E) and nodes (V), discretizing the points on the grid and weighting each edge based on how far it is from an actual point.

Each unlabeled sub-graph with maximal weight is merged into a single point, and then for each plateau (a maximal set of connected edges with identical weight), we look in the neighbourhood of each node and assign the node the value

$$\min_{x} \sum_{e_{ij} \;\; inE} w_{ij}^{p} |x_i - x_j|^q \tag{2.26}$$

$$s.t. x(F) = 1, x(B) = 0. \tag{2.27}$$

where

$$w_{ij} = \min(d_P(i), d_P(j)),$$

and $d_P(\cdot)$ is the distance between the nearest point in the set of all nodes P.

The power watersheds algorithm looks for a labeling xx of G, which is initialized with seed labels $x(B)$ and $x(F)$ on some points.

The algorithm is evaluated by iteratively finding an unlabeled node with maximal weight, and then finding the maximal sub-graph of edges with equal weight.

If this graph contains nodes with a known label then we find the surface label based on minimizing Equation (2.26). Otherwise, all nodes are merged and we find the next unlabeled node with maximal weight. Like Boykov and Lempitsky [38], the power watershed algorithm can operate in a thin band, which they define as the subset of the grid where the thresholded distance function at least divides nodes into the foreground and background.

2.8 Summary of Methods

The previous sections provided an overview of energy based methods for surface fitting to point clouds. Some of the reviewed algorithms are in 2D space, but the problems in 2D and 3D are extremely similar, and many of the 3D algorithms described above were inspired by the 2D algorithms mentioned.

Graph cut based methods have proved to be extremely effective for 3D surface fitting, as have TV methods. Surfaces constructed using graph cuts will have small metrification artifacts [5], and level sets based methods will often get stuck in local minima. TV methods lack metrification artifacts, but are difficult to implement when compared with graph cuts. Both are great choices.

Chapter 3

A Method for Quantitative Comparison of 4D-CT Lung Tumour Segmentations

3.1 Introduction

Internal imaging techniques have made it possible to develop treatments that are minimally invasive. It is, however, incredibly difficult for an oncologist to accurately determine the boundaries of a tumour; there is huge variability among a group of oncologists' segmentations of a tumour in a single patient. Inter-observer variability is the biggest source of uncertainty in treatment planning in radiotherapy [31]. We wish to quantify treatment variability based on target volume segmentation both prior to and following radiotherapy planning stages. The gross tumour volume (GTV) is the portion of the tumour that can be imaged: its volume does not necessarily contain the entirety of tumuorous growth [12]. Often the segmentation process of internal tumours from four-dimensional computed tomography (4D-CT) is complicated by tumour motion during the breathing cycle. Oncologists therefore provide segmentations at ten breathing phases $(0\%, 10\%, \cdots, 90\%)$ to ensure that dosages will be applied to the entirety of the tumour. These respiratory phases each represent a percentage of full inspiration.

In addition, tumours are "gated", which essentially means that some adjustment for the tumour's position between these breathing phases is accounted for using some type of motion tracking. Generating a 4D-CT usually involves combining multiple 3D-CTs at various phases of the breathing cycle.

There has been substantial research into auto-segmentation of target volumes from 4D-CT, but currently the most effective method is to segment the gross tumour volume (GTV) on each CT data set (i.e. each respiratory phase), which we then combine these to form a single integrated gross tumour volume (IGTV). To account for low contrast and/or unseen areas of disease, the margin of this volume is expanded by 5-10mm. We refer to this larger volume as the clinical target volume (CTV); it is the area that we wish to treat.

Finally, a similar and related term is the planned treatment volume (PTV). The PTV's purpose is to address the fact that uncertainties exist in both the planning and delivery of radiographic treatment. The dose may be distributed unevenly, and likewise, there areas of the body for which exposure to radiation would have severe effects, and thus the PTV must factor these things in as well. These are referred to as organs at risk (OR) [12].

The data we utilize are unorganized point clouds, segmented tumours from CT data; it is not intensity data. We start with the points that each physician specified as the boundaries on each slice in the CT data. The CT scans and segmentations were obtained at the London Regional Cancer Program. The subsequent statistical analysis, as seen in [45, 46] were performed by the other contributors, and are present to help explain the larger significance of the work.

We contribute a method for computing a similarity measure for the volumetric comparison of sparse point clouds of tumours as segmented from CT data (as well as the actual computation of volumetric overlap and distance-based error) for the compared point clouds in the data set.

3.2 Background

For reasons we described in Chapter 2, we rely on Lempitsky and Boykov's [38] flux-based method for providing a globally optimal surface for each point cloud. Their method solves for a sparse set of normals where only weak orientation data is necessary. They rasterize the points onto a dense 3D volume and solve their functional on a thin band surrounding the points, using the Touch Expand algorithm. Touch Expand is an algorithm for solving max-flow problems on a graph with source and sink arcs.

The surface fitting function

$$max \overset{S}{\leftarrow} \int_S < v_s, n_s > ds \tag{3.1}$$

uses normals and the surface velocity and is specifically for the circumstance when perfect data is available. That is obviously not the case here, and thus the elastic membrane prior is utilized. We want to compute

$$min \overset{S}{\leftarrow} \int_S \lambda ds - \int_S < v, n_s > ds. \tag{3.2}$$

For the graph's set of nodes contains two special terminal nodes, the *source*

s and the *sink* t. We then have non-terminal nodes $p \in P$, each of which represents a voxel and is connected to its neighbouring nodes via n-links and terminal links of the form (s, p) and (p, t), as was explained in Section 2.3.

Neighbouring nodes are connected via n-links, these represent the regularizing term. The data term is represented by t-links, where nodes with potentials greater than zero $U(p) > 0$ are connected to the source and nodes with potentials less than zero $U(p) < 0$ are connected to the sink.

We also utilize the method's closed topology, which works by extending the boundary of the dense grid in each dimension and setting the boundary points' t-links to a large, negative value.

The sparse vector field $\{\bar{v}_i | \bar{p}_i \in P\}$ is blurred by the width of the flux's diffuse radius, a parameter set prior to run-time. In practice, a box filter is used to perform this n times, where n is a preset parameter. The result of this blurring is to obtain a semi-dense field $\bar{v}_p = \Sigma_{i=1}^{N} \bar{v}_p^i$. These vectors are a set of semi-dense averaged normals, obtained by applying said box filter to sparse normals.

3.3 Problem

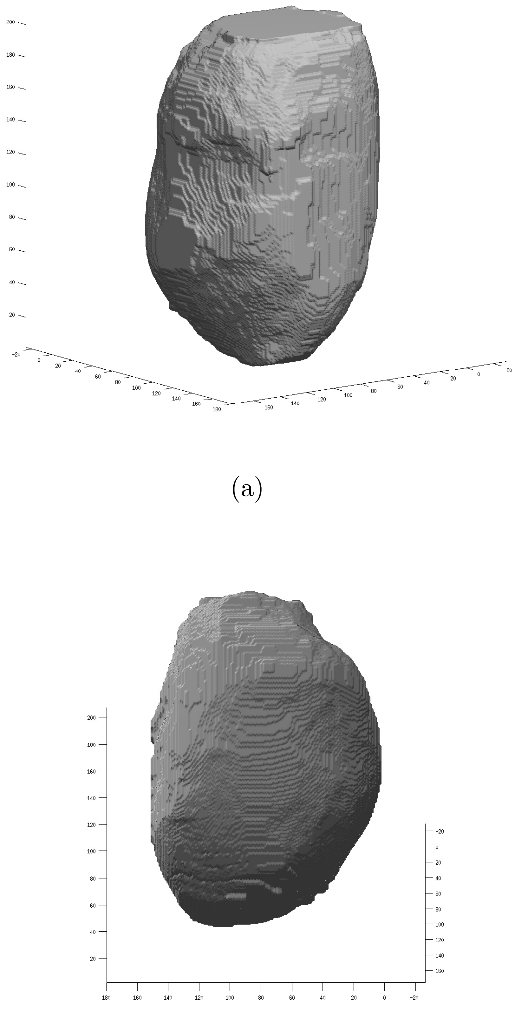
The method by which physicians segment a dense 4d image to produce a 3D point cloud, outlining the perimeter of 2D slices, produces a set of points that incompletely describes the surface. Specifically, the segmentation is performed by a physician in an application called Pinnacle Treatment Planning System [67]. On the periphery of each dimension, the edge is implied, i.e., in one slice there will be a segmentation performed, and then in the adjacent slice there won't be. In some cases, the segmentation in the next-to-last slice will be substantial and large gaps remain to be filled. This is the reality of the current process of segmentation, not a problem with a physician failing to do their job.

Given sparse and incomplete data, as we see on the top of Figure 3.1a, how do we compare the difference between two point clouds in a meaningful way? This makes using volumetric methods for comparison difficult. Similarly, using point based methods don't solve this problem either; we have large swaths which are implied parts of the tumour that also need to be quantified.

3.4 Methodology

Ten patients with Non-Small Cell Lung Cancer (NSCLC) were imaged using 4d-CT using a Big Bore CT scanner, along with the Real-time Position Management (RPM) system for gating. The 4d data has x, y, z and intensity values, where the$z$-axis is the coronal plane, the x-axis is the sagittal plane, the y-axis is the axial plane and their units are in millimetres (mm). RPM involves having markers placed on the patient's abdomen or chest, and tracking these markers using an infrared camera. Using this, the images are generated at the breathing phases and correspond to one another, both for a patient's images captured at different phases, and other patients' images captured at the same phase [46].

Thus, 10 4D-CT images are generated for each patient, an example of which can be seen in Figure 3.2 A 4D-CT can be thought of as a "stack" of intensity images: a loaf of multi-coloured bread is an apt comparison. A physician takes each of these "slices" for a single breathing phase, and must identify the interior of the tumour for this slice from the exterior, typically with a mouse by clicking around the boundary of the tumour. In addition to this being a difficult task – the largest source of uncertainty of treatment



(a)

(b)

Figure 3.1: Two different views of the same segmented primary GTV. Note the top of the surface is flat and this made the volume fitting more challenging because the flux width has to be wider to ensure that the entire surface is contained. The flat top is visible in (a).

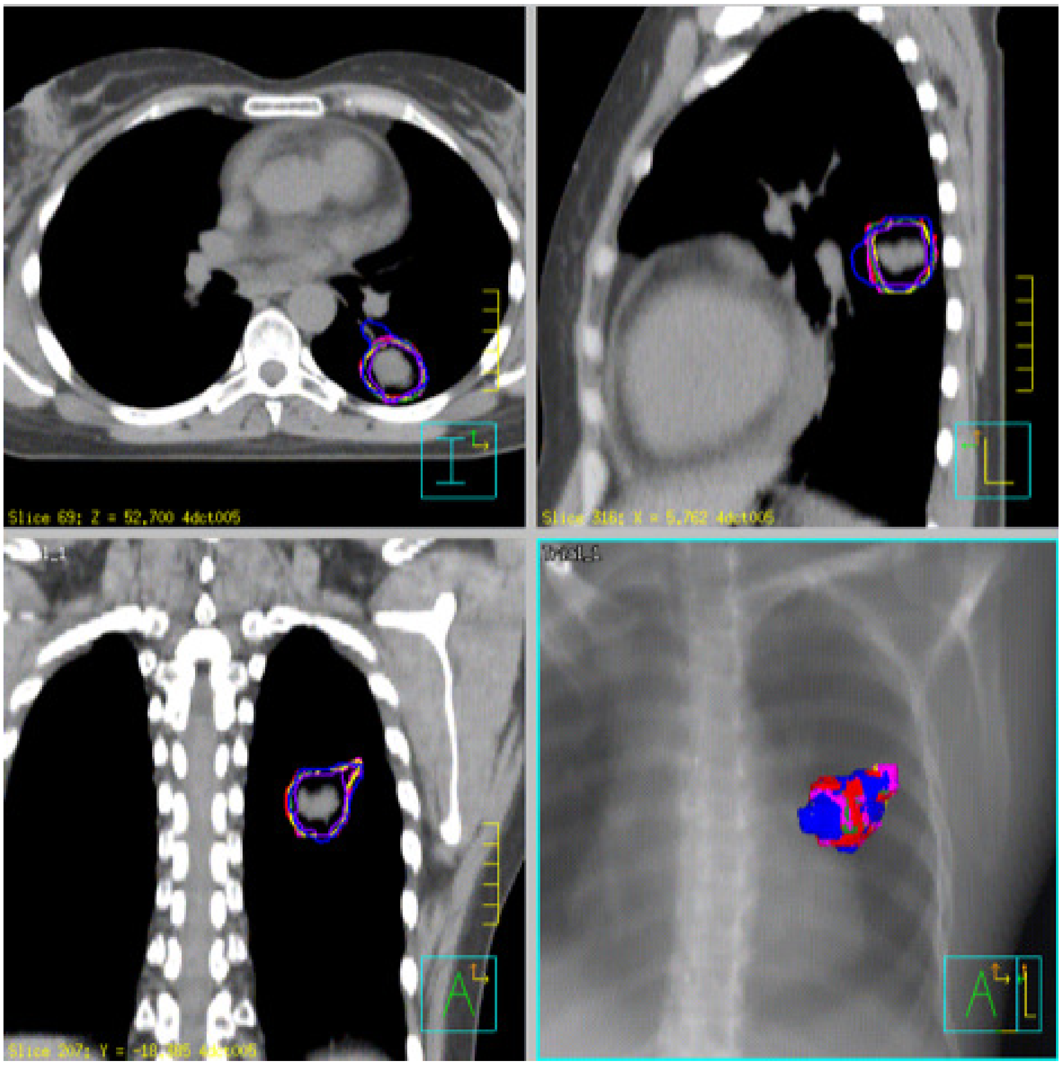

Slice 69; Z = 52,700 4dct005
Slice 316; X = 5,762 4dct005
Slice 207; Y = -18,485 4dct005

Figure 3.2: Slices of physician provided segmentations for patient E. The GTVs on different axes are shown at top left, top right and bottom left, while the bottom right image shows the superimposed 3D meshes for all segmentations. Top left shows a slice at $z = 52.7$, top right shows slice $x = 5.62$ and bottom left shows slice $y = -18.485$.

is inter-observer variation [31] – it is also a highly time consuming task, as a single 4D-CT image often has more than 100 slices.

3.5 Point Cloud Construction

Each 2D point (x_i, y_i) is combined with its z_s, the height (i.e. slice) measurement, and these points are collected to make a single point cloud, an example of which is seen in Figure 3.3. The larger goal of this work is to determine the efficacy of treatment, but the technical goal is to deliver a method that can consistently measure how well the treatment is administered, i.e. how much overlap there is between the tumour and the treated area.

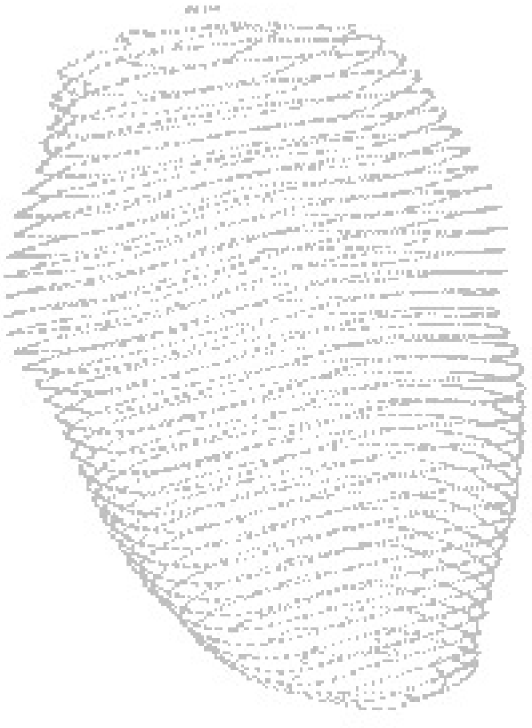


Figure 3.3: Point cloud extracted from physician segmentations.

3.5.1 Surface Fitting

The nature of the point clouds combined with our need for a watertight surface created a significant challenge in providing an accurate surface fitting. If one thinks about the slices of a 3D-CT, the “top” of the tumour will typically be quite wide in terms of the segmentation. The other points are generally fairly evenly-spaced, but the top and bottom often have a large gap and this makes getting a “tight” fitting surface a problem. To generate accurate, automated and watertight segmentations for almost 1000 images is a challenge.

The flux based surface fitting method requires approximate orientation data for computing the max-flow/min-cut. It is, as they note, acceptable to use a single normal for all points in a single range scan. We do not have the benefit of such orientation information, so we first fit an approximate surface and then find vertex normals based on this surface.

Our C++ application (*trimesh_normal*) that imported each of these files sequentially first constructed an approximate surface mesh to get a field of normal vectors that is approximates the true field. We used the ball pivoting algorithm to first fit the approximate surface, and then compute the normals using said surface. Our application uses VCGLIB’s [18] routines for ball pivoting, normal estimation and orientation, and TouchExpand [38] for watertight surface fitting.

Ball pivoting is an advancing front algorithm [1]. A “seed” point is selected, and two neighbours are searched for such that they can be circumscribed in the radius of the ball with an empty interior. If such a pair of vertices are found, a triangle is created [22]. The ball is then pivoted, i.e. it is rotated around the edge until it circumscribes another point, forming another triangle. In the case where multiple points are found, the point with the smallest distance from the

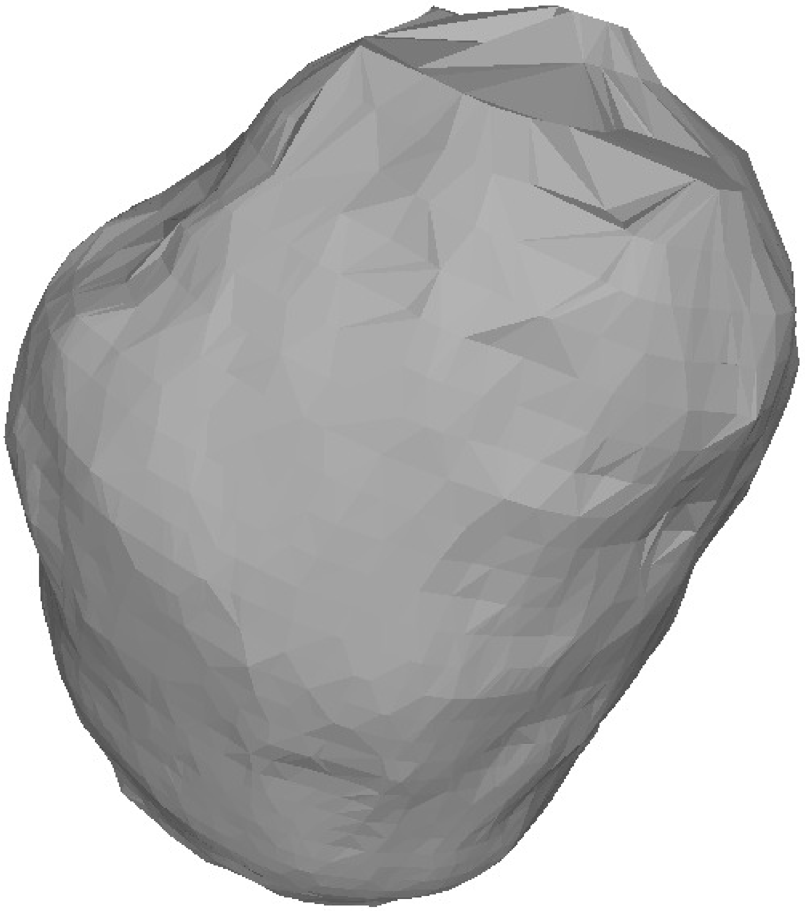


Figure 3.4: Approximate meshed surface for point cloud seen in Figure 3.3, as fitted using the ball pivoting algorithm.

edge is chosen. This process is continued until all edges have been traversed, at which point a new seed triangle is found [1].

The ball pivoting algorithm is uses a sphere of radius r (in our case $r = 1$) and "rolls" it over the point cloud, adding a triangle (i.e. edges and a surface) if the ball contains three points only. In Figure 3.4 we see the approximate surface fit after ball pivoting has been applied, which we then utilize to compute surface normals.

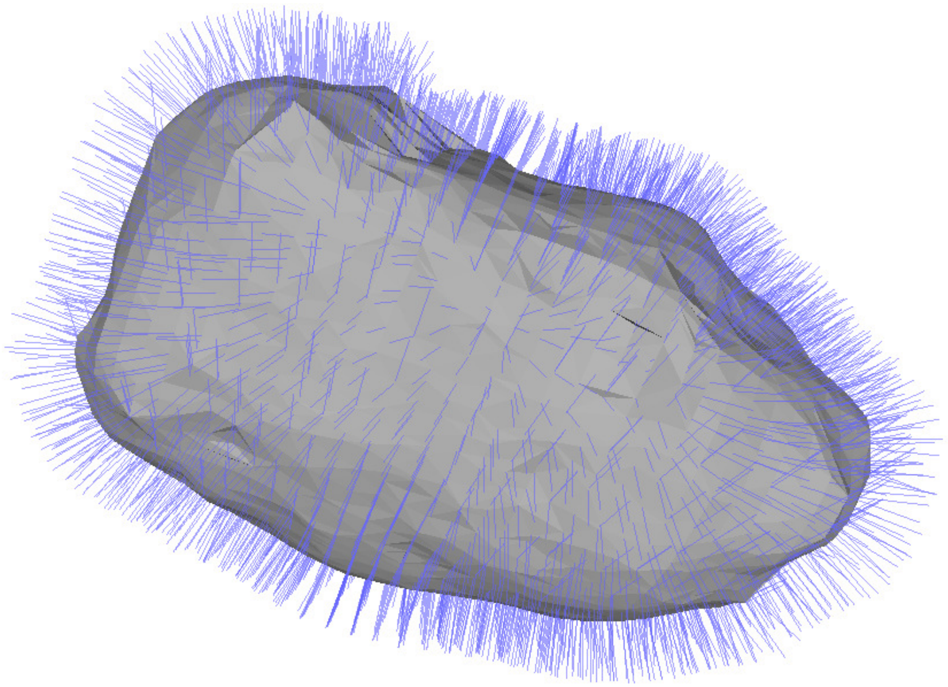


Figure 3.5: Normals calculated for an approximate surface.

3.5.2 Computing Normals

We first calculate the per face normals for our newly-constructed mesh. For each triangle with vertices p_1, p_2, p_3, we calculate vector $u = p_1 - p_2$ and $v = p_3 - p_1$, then the normal $n = u \times v$. The choice of variable assignment determines the orientation of the normal, i.e., whether it points inward or outward, so we wind the normals subsequently.

We then normalize the lengths of these normals and utilize them to compute the normal for each vertex. We use equally weighted averaging of facets, i.e. $n_v || \sum_{i=1}^{3} n_i$, where n_v is a vertex normal and $||$ implies parallelism and thus subsequent normalization, i.e. $n_v = n_v \frac{n_v}{|n_v|}$, we borrow this notation from [35].

As we said above, the computed normals' orientation is strictly based on our assignment of each p_i to a vertex. We the wind the normals to ensure that they are pointing outward, an example of such a field can be seen in Figure 3.5.

Computing accurate surfaces for each patient ended up being heavily computationally expensive. Computing almost 1000 watertight, globally optimal surfaces is expensive, and one of our requirements was that the resultant 3D volumes are comparable. Our bounding box encapsulated every point, but also had a boundary at the maximum and minimum of each dimension, as TouchExpand needs sufficient space around a voxel to approximate the energy using neighbouring voxels: when neighbouring points are boundaries, we experienced trouble with convergence. We settled on a grid size of $150 \times 150 \times 150$, which is quite large and resulted in expensive surface computations, but given that the bounding box encapsulated *all* tumours, it was necessary to have high resolution since in practice some tumour volumes would only be occupying a small amount of the bounding box.

There was an element of trial and error in generating a set of terms that could accurately capture an appropriate surface for our entire set of point clouds. In practice it was very easy to tell when a surface was not watertight: when we computed the volumetric overlap [28] where E_o is zero when overlap is perfect, and one when no overlap exists.

3.5.3 Minimal Surface Fitting to Segmented Point Cloud

As we discussed in Chapter 2, we use the minimal surface as a guiding principle for our surface fitting: we wish to obtain the surface with a volume that encapsulates the points to which we are trying to fit the surface. In the preliminaries, we mentioned that this idea was introduced to the vision world by Caselles et al. [14], though their level sets-based method was prone to local minima due to the level sets implementation.

Instead, we use a graph cuts-based method, specifically the flux-based method as introduced by Boykov and Lempitsky [38]. Again, as we mentioned in the introduction, graph cuts-based methods promise to provide a global minimizer for the proposed functional.

From the ball pivoting algorithm, we now have a mesh, comprised of faces with normals, from which we generate vertex normals. We then rasterize our vertices onto a discrete grid and compute the flux for these points.

For our purposes, we used TouchExpand with a topology constraint (closed surface) with a larger diffuse radius for the flux ($d_f = 2$, versus the default of $d_f = 1$), as well as an extra iteration. This is to ensure that we can guarantee a closed surface.

The result of TouchExpand is a binary, dense 3D volume where the inside of the point cloud is marked by 1's and the outside by 0's. Given our problem of computing volumetric overlap, this is convenient, as we can compute E_o directly by just iterating over all points.

We performed a cursory visual inspection of each generated mesh to ensure a watertight result, however, our method could easily detect when we lacked convergence, i.e. a closed surface. If the volumetric overlap was less than a reasonable threshold (generally 70%), it was usually obvious that *trimesh_normal_surf* had failed to obtain a watertight surface for one (or both) of the point clouds. While each tumour had been segmented by eight doctors, the level of agreement between them was generally at least 70%.

3.6 Similarity Quantification

We chose the method of Lempitsky and Boykov [38] after our evaluation of many methods, as seen in Chapter 2, primarily because it solves both our core problems. Surface evolution methods and local methods cannot guarantee finding a global minimum when fitting a watertight surface to a point cloud. Their TouchExpand library is freely available and has an optional shape prior for closed surfaces.

Definition 3.6.1 (Volumetric Overlap)**.** For two watertight surfaces S and G that exist on the same grid structure, if all occupied voxels s_i are occupied by G and all voxels g_i are occupied by S, we say that a perfect volumetric overlap ($VOE = 0$) occurs

More specifically, the volumetric overlap is

$$E_o(S, G) = 1 - (|S \cap G|)/(|S \cup G|). \tag{3.3}$$

We use E_o, a.k.a. the Jaccard index because, as we point out in [46], using R^2 error is disproportionately punishing as distance increases between a source and target. Moreover, the numbers aren't comparable between tumours of largely varying sizes: we wish to know how well the physician performed independent of the size of the tumour.

Definition 3.6.2 (Nearest Neighbour of x and Y)**.** For a point x, the nearest neighbour in a set of points $Y = \{y_1, \cdots, y_n\}$ is the point y_i with minimum Euclidean distance from x, i.e.

$NN(x, Y) = \min \sqrt{(x - y_i)^2}, \forall y_i \in Y$.

Likewise, we also have the Nearest Neighbours for two sets

Definition 3.6.3 (Nearest Neighbours of X and Y)**.** For a set of points X, the nearest neighbours of a set of points Y is a set such that $NN(X, Y) = NN(x_j, Y) \forall x_j \in X$.

Definition 3.6.4 (Symmetric Nearest Neighbour)**.** For two sets of points X and Y, two points x_m and y_n are considered symmetric (i.e. mutual) nearest neighbours if y_n is the nearest neighbour of x_m and y_n is the nearest neighbour of x_m.

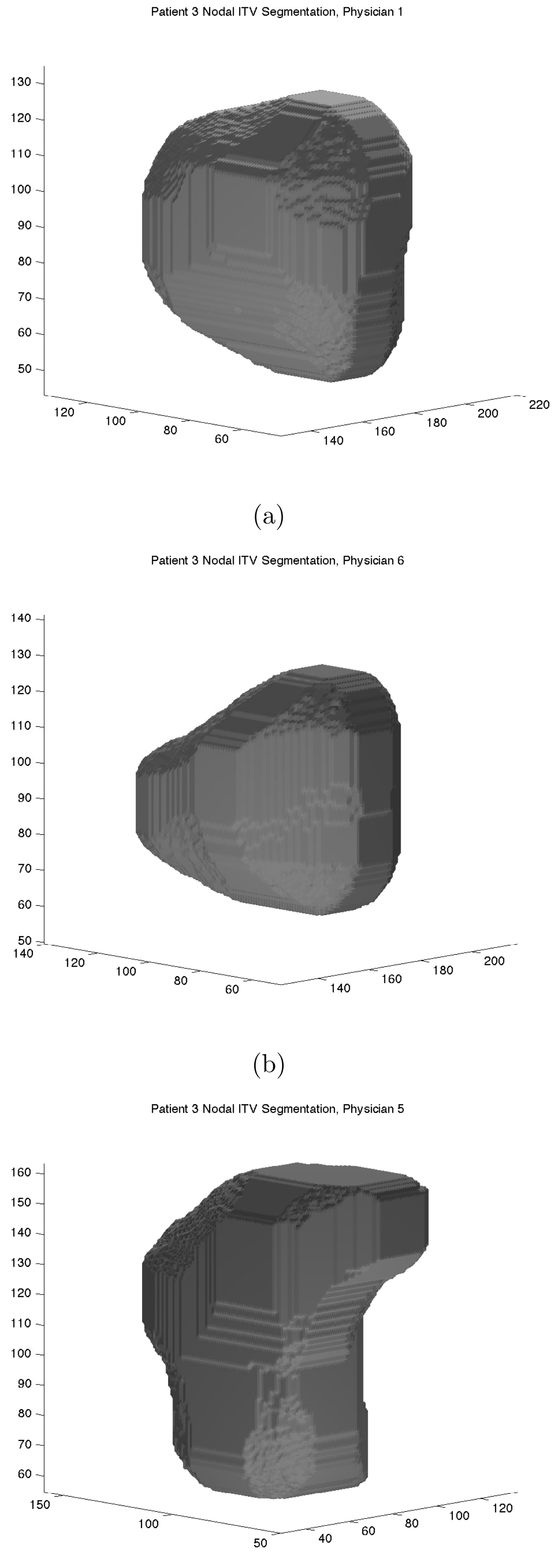
In practice, we store our point clouds in kdtrees to improve the complexity of computing nearest neighbour options, which we use to compute the symmetric $RMS - E$, which is the $RMS - E$ for symmetric nearest neighbours.

3.7 Results

As we see in Figure 3.10, segmentations of patient B's GTV will vary greatly among physicians. Each colour represents a different physician's segmentation. In the top left and bottom left of Figure 3.10, we see a large difference on the y-axis; each physician has a very different view of what does and does not encompass the tumour. Indeed, the reconstructions shown in the bottom right of Figure 3.10 (each has a different colour), are extremely different from one another.

This, in fact, is a moderate variation. Figures 3.6 and 3.7 show how dramatically different a set of physicians' interpretation of what does and does not constitute a tumour can be in both primary and nodal data.

For the VOE of the primary IGTV, the combined GTVs from all 10 breathing phases, in terms of observer (intra-patient) we saw a large variation. Individual patient VOE measures can be seen in Figure 3.8. Overall, VOE was


Patient 3 Nodal ITV Segmentation, Physician 1

(a)

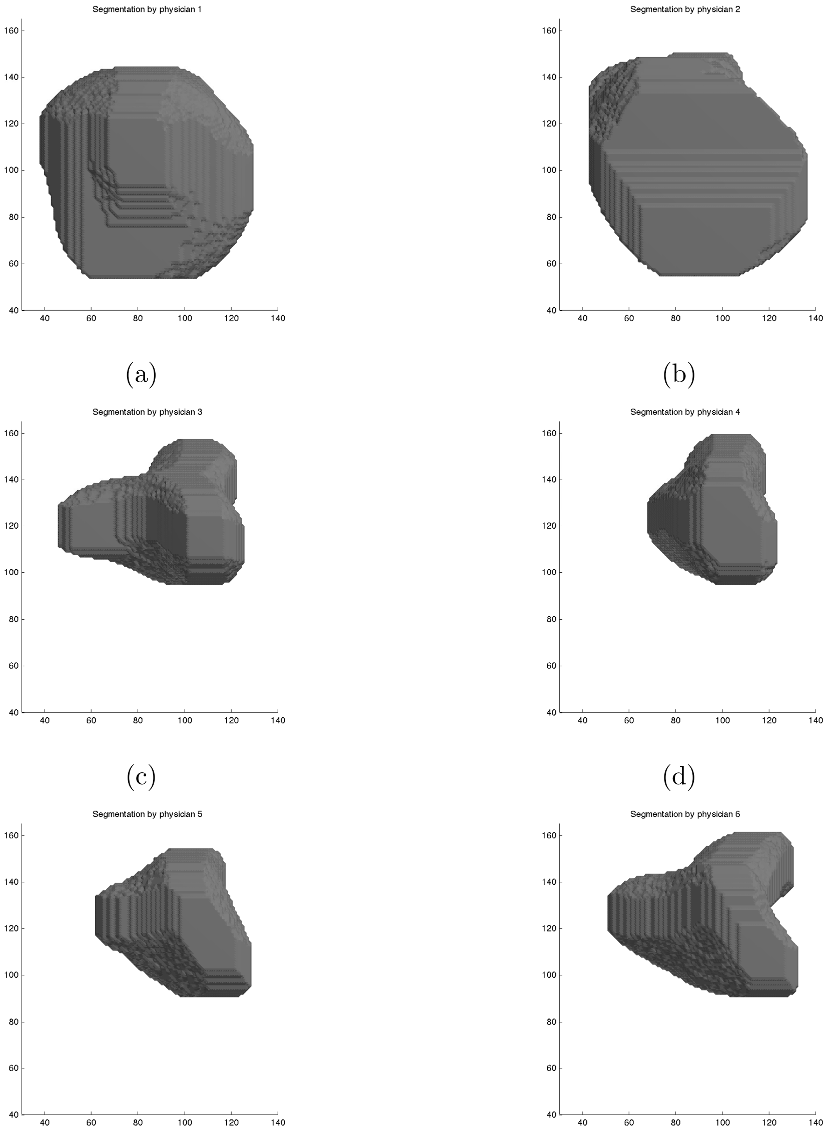
Patient 3 Nodal ITV Segmentation, Physician 6

(b)

Patient 3 Nodal ITV Segmentation, Physician 5

(c)

Figure 3.6: Three different segmentations of patient 3's nodal tumour by different physicians.


Segmentation by physician 1
Segmentation by physician 2
Segmentation by physician 3
Segmentation by physician 4
Segmentation by physician 5
Segmentation by physician 6
40
60
80
100
120
140
160

(a) (b)

(c) (d)

(e) (f)

Figure 3.7: Six segmentations of patient 3's primary tumour by different physicians.

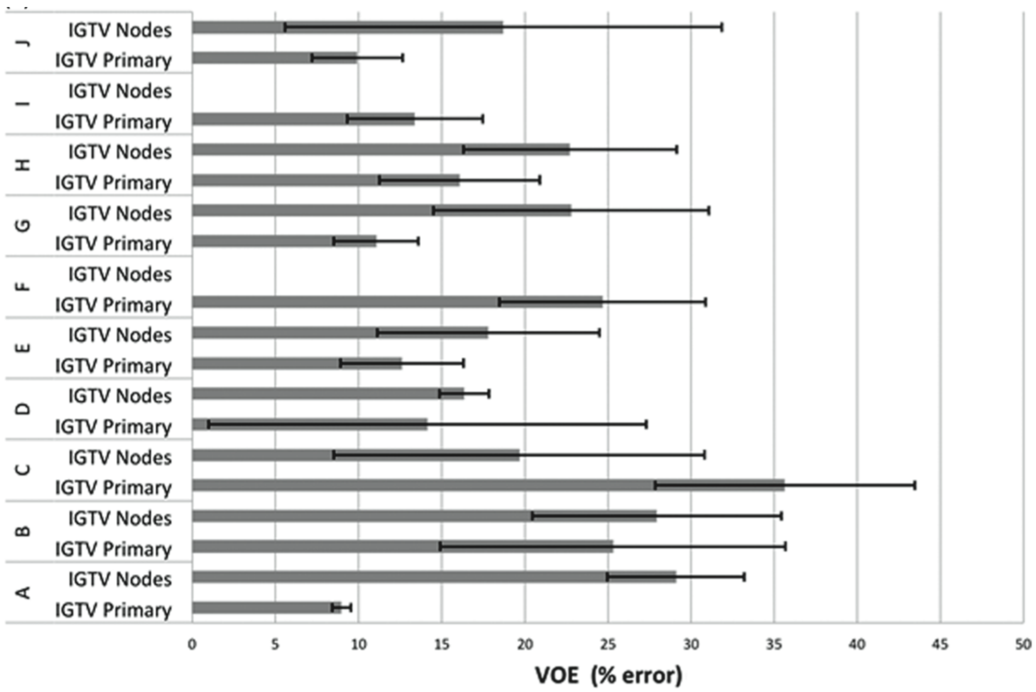


Figure 3.8: Graph of volumetric overlap for gated tumours for each patient.

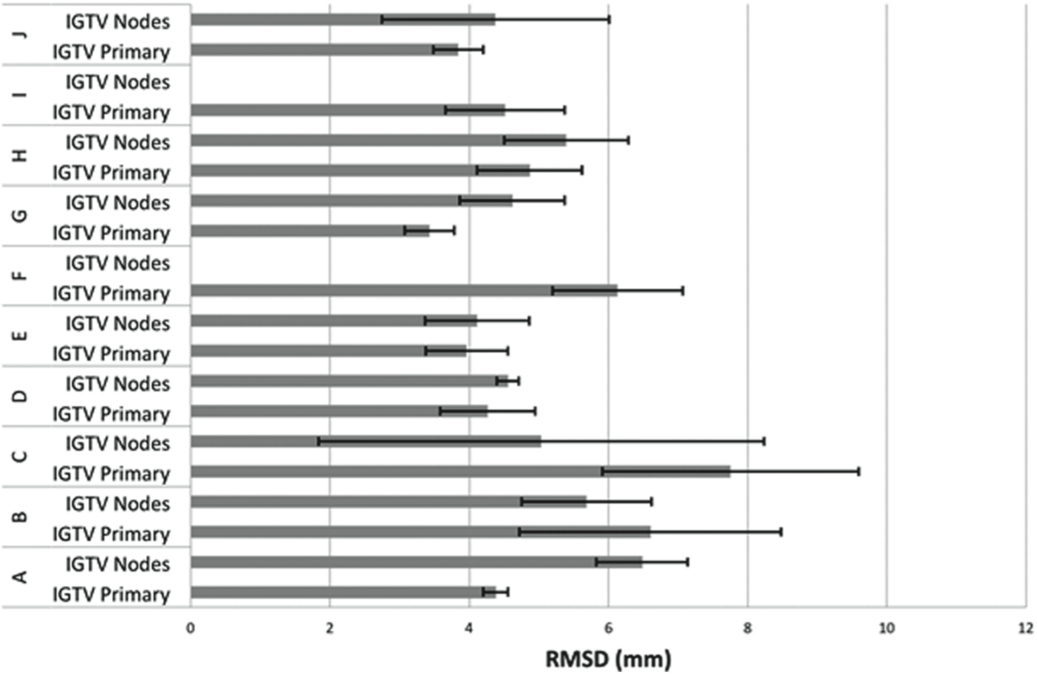


Figure 3.9: Graph of RMS distance for gated tumours for each patient.

$(9.0 \pm .6)$ to (35.6 ± 7.5) and in nodal tumours $(16.4 \pm .5)$ to (29.1 ± 4.1). We saw inter-patient σ from 0.6% to 13.2% for primary tumours and from 1.5% to 13.1% for nodal tumours. For symmetric RMS distances (individual results can be seen in Figure 3.9), we saw $(3.4 \pm .4)$mm to $(7.8 \pm .8)$mm for primary IGTVs with standard deviations in the range $(0.2mm, 1.2mm)$. The nodal IGTVs had mean observer RMS symmetric distances ranging from $(4.1 \pm .8)mm$ to $(6.5 \pm .7)mm$ with σ ranging from 0.2mm to 3.2mm. Note that RMS is a distance but VOE is a ratio, hence the lack of a unit in the former.

The symmetric RMS distances are affected by the size of the tumour volume, and the nodes are smaller than the primary volume. It is also substantially more difficult to accurately segment the nodal volumes, partially because nodes tend to exist in low contrast areas.

Both lymph nodes can be affected, in which case there will be more than one nodal volume. The process of constructing an initial, approximate surface, calculating surface normals and then utilizing [38] handles this constraint effectively, utilizing the same parameters as those used for the GTV to ensure that comparisons are fair.

For each of the hundreds of segmented point clouds, our method successfully fit a watertight figure, some examples of which can be seen in Figures 3.1. This required an element of trial and error, as it was necessary to find parameters that would avoid smoothing out contours and still maintain a watertight surface. Our results demonstrate a clear trend, so in addition to guarantees that TouchExpand provides in terms of optimality, we can feel confident that it successfully captures an accurately modeled, representative shape.

The COVs (coefficient of variation) of physician segmentations of both the primary and nodal IGTVs, as seen in Figure 3.1 are extremely enlightening.

	PTV Primary			PTV Nodes		
Patient	Max VOE (%)	Min VOE (%)	COV (%)	Max VOE (%)	Min VOE (%)	COV (%)
A	9.85	8.36	6.30	33.18	22.83	14.20
B	40.89	10.54	41.07	37.51	21.07	26.87
C	41.71	20.99	21.93	84.10	12.82	105.75
D	40.31	5.96	93.25	18.24	14.21	9.10
E	17.66	7.14	29.28	29.67	12.48	37.59
F	34.02	15.97	25.22	---	---	---
G	15.69	8.27	22.93	34.53	11.16	36.40
H	24.35	10.36	29.98	31.22	13.60	28.22
I	18.59	8.76	30.49	---	---	---
J	13.56	6.14	27.57	38.50	8.41	70.29

Table 3.1: Coefficient of Variation for VOE.

Two patients did not have nodal tumours, but of the five of the eight who did, the COVs of nodal IGTVs were larger than the primary IGTVs, though for patient D, the primary GTV has a very large COV (93.25%). Overall, both nodal and primary GTV segmentations tend to have very high COVs.

3.8 Conclusions and Future Work

One clear conclusion from our research is that there is a high degree of variation among physicians' interpretations of what is and is not tumourous growth and thus their resultant segmentations and dose distributions are highly variable. This is true of both primary and nodal tumours, and though nodal tumour performance is more variable, both suffer from the same problem to a large degree.

Advancements in technology have brought about improved target localization via gating of imaged data, as well as methods to more effectively target tumours with radiation. The process of segmentation remains error prone; quantification of performance and subsequent feedback will only improve the performance of physicians and help ensure that patients are being given appropriate dosages.

Because the human cost is so high, experimentation is limited. The data set that we have produced here was hugely time consuming to compile, meanwhile, limited human and physical capital within the health-care system is the norm. Any steps toward automation and standardization will be a huge win, and the method presented is a step in that direction.

We would like, in future work, to perform a similar analysis on different types of tumours, in hopes of obtaining a quantitative measure of what types of tumours are most difficult to segment.

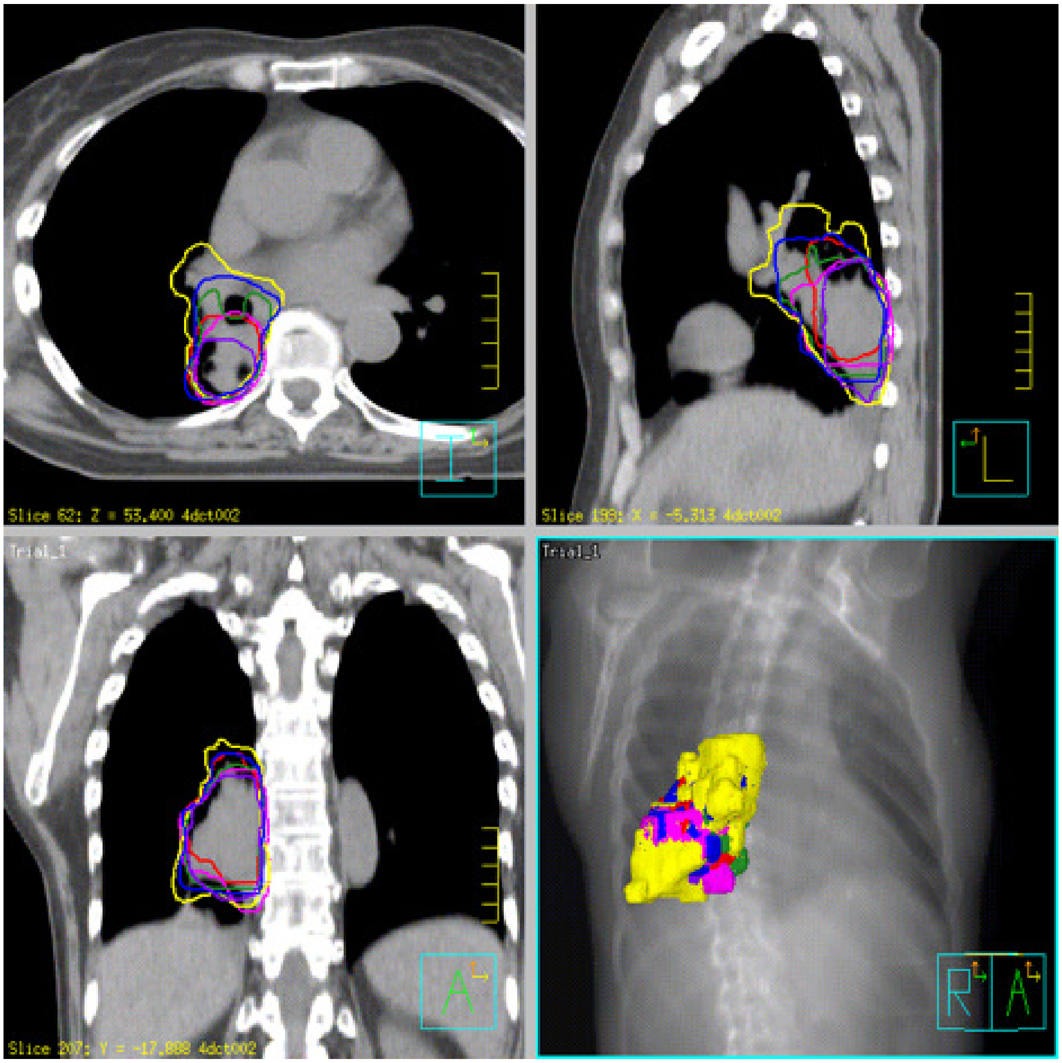

Slice 62: Z = 53.400 4dct002
Slice 138: X = -5.313 4dct002
Trial_1
Slice 207: Y = -17.808 4dct002
Trial_1
I
L
A
R
A

Figure 3.10: Slices of physician provided segmentations for patient B GTV on different axes are shown at top left, top right and bottom left, while bottom right shows the superimposed 3D meshes for all segmentations. Top left shows a slice at $z = 53.400$, top right shows slice $x = -5.313$ and bottom bottom left shows a slice at $y = -17.808$.

Chapter 4

Global Registration of Multiple Thin Structures

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. 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 analyze 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 like Stanford bunny. 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 ([64], [21]), still it remains an open problem in medical imaging community. One motivation of this chapter is to address the problem of aligning multiple views of thin recursive structures like plant or vascular data. We show an example of the latter in Figure 4.1.

We introduce 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 neighbour, followed by the alignment of another neighbouring 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.

Inspired by Toldo et al.'s work [69] 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 Neighbours (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 4.2. To overcome the limitations of 2D image-based plant analysis (lack of accurate spatial and volumetric information [53]), 3D imaging is essential for measuring various plant parameters that indicate plant growth. A significant body of literature has been reported on plant growth analysis. For example, Clark et al. [19] proposed a high throughput software system for analyzing root of a plant. Paproki et al. [53] demonstrated a 3D approach for measuring plant growth in the vegetative stage. They captured data as high resolution 2D images and generated 3D mesh data from these images. Recently, Paulus et al. [55] used 3D laser scanning technology to

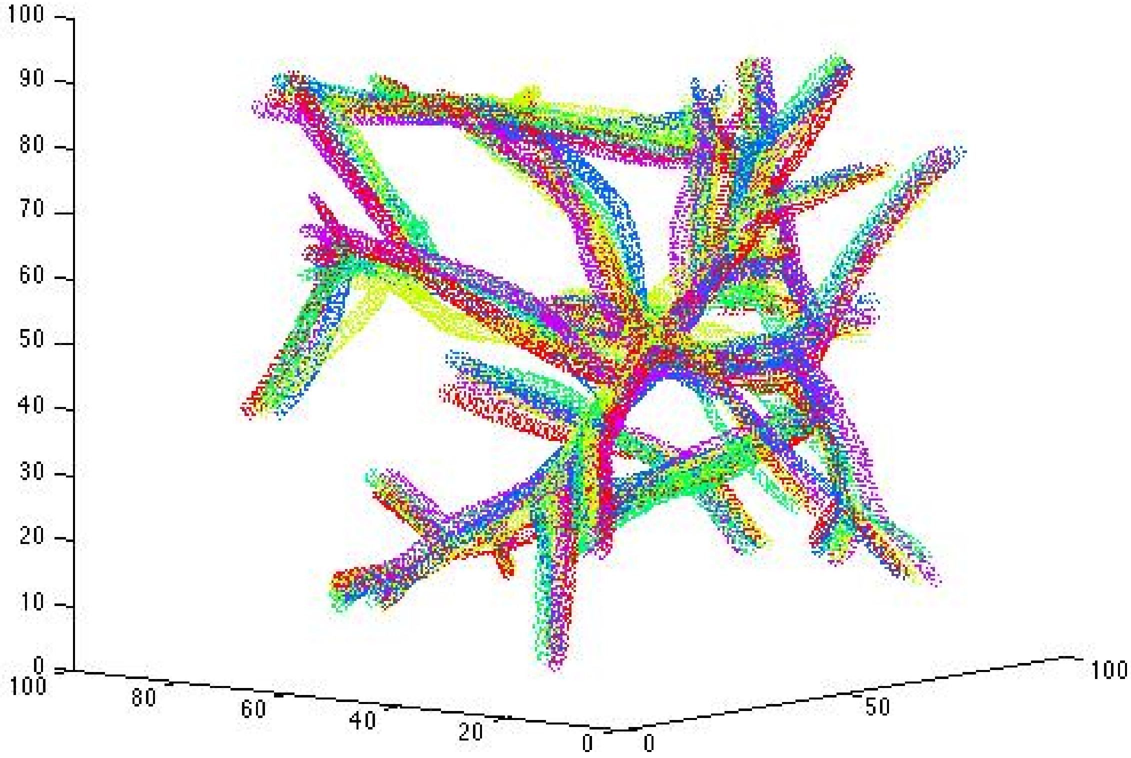


Figure 4.1: A subset of the views of the synthetic vascular data. We take one scan and then apply a set of random, non-rigid deformations to obtain our data set.

perform organ classification of plants. Li et al. [41] proposed a technique to track budding and bifurcation of plants from raw point cloud data.

In order to obtain the 3D structure of an object, a range scanner is used to capture point cloud data from multiple viewpoints. These data sets are merged together and then triangulated to build a polygonal mesh of the object. However, most of the current methods for doing this concentrate on the optimization involved in reconstructing standard 3D models, with the constraint that the calculation be computationally inexpensive. Here, we are interested mainly in modeling complex plant objects and, as well, have to consider the non-rigidity of the plant caused by the inter-scan movement of the object due to weak and uncontrollable wind currents in the growth chamber. We are investigating how well a plant's volume is a function of its growth as a long term goal.

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) [2] algorithm finds a rigid transformation that registers two views using local optimization.

Several variants and applications of the classical ICP were reported in the literature [71, 78, 47, 57, 59, 30, 4]. Registration of the set of plant range scans, using the existing methods whose implementations are available, were poor because of the inter-scan movements between different parts of the plant range images (the plant is not rigid). Further, due to the way our (Arabidopsis) plant's branches are interleaved, we found that ICP could not find a reasonable registration and would frequently get stuck.

Recently, Gaussian Mixture Models (GMM) have been used for non-rigid registration. Our plant data 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. We have concluded from our experimental work, that Myronenko et al.'s Coherent Point Drift (CPD) [50, 49] method works best for aligning two point clouds of the plant data. However, Myronenko et al. didn't consider aligning more than two views. We propose an algorithm based on CPD which can align many views with minimal error. 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 rigid pairwise registration and then use our global method to create the final point cloud.

4.1 Previous Work

Among several robust methods for registration, some notable work can be found in [16, 70, 77, 65]. These algorithms belong to a similar class of approaches. We found that various non-rigid registration methods are reasonably effective in registering adjacent scans, but any attempt to merge multiple registrations into a single point cloud was problematic.

Fitzgibbon [23] 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 data set (which satisfies this constraint). Applying the algorithm to our plant data yielded poor results.

The Point Cloud Library (PCL) [42] provides some libraries for point cloud alignment and is based on Rusu et al.'s method [61, 60] for Fast Point Feature Histogram (FPFH). Rusu et al.'s contribution is in finding efficient matching of features between two sets of point clouds. Each point in a data set 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. We have implemented Rusu et al.'s algorithm using the point cloud library. The results are shown later in this section.

Some robust rigid registration methods have used GMM: Jian and Vemuri [33] followed this approach (GMMReg) 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 data set. The algorithm does not work for our plant data set (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. 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. [50, 49]. 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

---

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

function will be $p(x) = \sum_{i=1}^{M+1} P(i)p(x|i)$, where:

$$p(x|i) = \frac{1}{(2\pi\sigma^2)^{D/2}} \, exp[-\frac{||x - y_i||^2}{2\sigma^2}]. \tag{4.1}$$

Instead of maximizing the GMM posterior probability, the negative log-likelihood function can be minimized to obtain the optimal alignment:

$$E(\theta, \sigma^2) = -\sum_{j=1}^{N} log \sum_{i=1}^{M+1} P(i)p(x|i). \tag{4.2}$$

They iteratively use the Expectation Maximization algorithm to optimize the cost function. The algorithm is also robust to noise. We have used this algorithm and have run the same 2 adjacent point clouds of our plant data as for the other algorithms we tested.

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). We show our results on real Arabidopsis plant data. The results are shown in Figure 4.2 for the Stanford Bunny data set and Figure 4.3 for the real Arabidopsis plant.

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 manually acquired 18 and 44 correspondences respectively and used them as ground truth. 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 4.1. We can observe that for Bunny data sets, 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

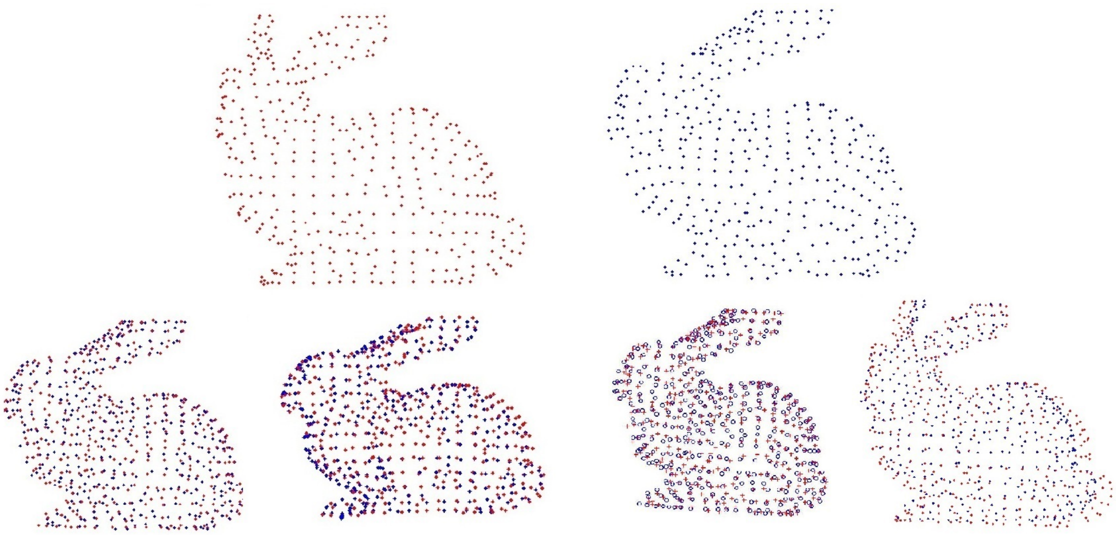


Figure 4.2: Pairwise registration results for two scans of the Stanford Bunny. Top row: original two views, Bottom row: results obtained using Rusu et al. (FPFH) [61], Fitzgibbon (LMICP) [23], Jian and Vemuri (GMMReg) [33] and Myronenko and Song (CPD) [49].

	Algorithms			
Data sets	FPFH	LMICP	GMMReg	CPD
Bunny	0.149	0.046	0.012	0.014
Arabidopsis Plant	9.661	8.302	N/A	3.254

Table 4.1: Quantitative results for different algorithms and data sets.

been some errors. Although 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.

4.2 Proposed Method

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

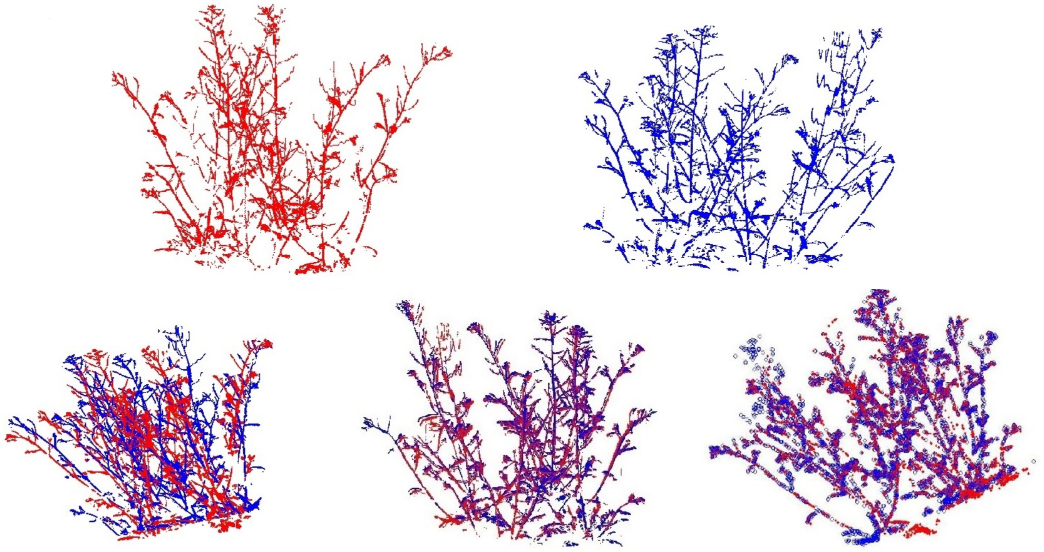


Figure 4.3: Pairwise registration results two scans of the real Arabidopsis plant data. Top row: the original two views. Bottom row: results obtained using Rusu et al. [61], Fitzgibbon [23] and Myronenko and Song [49].

X_i to an "average" shape, which we construct using the centroids of the *mutual nearest neighbours* (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 (4.1) to perform global registration

$$p(x|i) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp[-\frac{||x - \hat{y}_i||^2}{2\sigma^2}], \tag{4.3}$$

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

$$x_{i_n} = \min(|x_p - y_j|), \forall x_p \in X, \tag{4.4}$$

and

$$y_{j_n} = \min(|y_q - x_i|), \forall y_q \in Y. \tag{4.5}$$

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

$$x_{cent} = \sum_{i}^{n} \frac{x_k}{n}. \tag{4.6}$$

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 the “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.

4.2.1 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:

$$\hat{X}_i = R_i X_i + \vec{t_i}, \tag{4.7}$$

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 4.4.

4.2.2 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 neighbour 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 (4.6). For each point in scan X_i, we find the single nearest neighbour 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 $tol_{max}$.
  $tol = \sum_{i=1}^{N}$ error_$L^2(X_i)/N$
  while $err > tol_{max}$ do
    for $i = 1$ to $N$ do
      for $j = 1$ to $N$ do
        if $j \neq i$ then
          $X_{i_{cent}} = \mathrm{MNN}(X_j, X_i)$
        end if
      end for
      $X_{cent} = \mathrm{centroids}(X_{1_{cent}}, \cdots, X_{N_{cent}})$
      $X_i = \mathrm{register\_cpd}(X_{cent}, X_i)$
      err = norm($X_{cent}$, $X_i$)
    end for
  end while
```

4.3 Results

We show our results for the synthetic and plant data in the following sections.

4.3.1 Synthetic Vascular Data

We first demonstrate the efficacy of our method on synthetic vascular data. We take a 3D point cloud of synthetic veins, as generated by VascuSynth [32] 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 **sampling** $= 0.1$, **power** $= 3$ and $\lambda = 4$, the latter of which represents the relationship between fit and regularization. Initially and $\sigma = 5$ 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 4.6.

The magnitude of this transformation is substantially larger than those created by the wind in the Arabidopsis 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.

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 4.7 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 4.6. Quantitatively, we have shown that it easily outperforms sequential pairwise registration, that our method limits drift.

4.3.2 Plant Data

Figure 4.5 shows all 12 scans, merged into a single point cloud after subsampling each scan. Each colour 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 data-sets will bias the motion of a scan that is being registered. Figure 4.8 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.

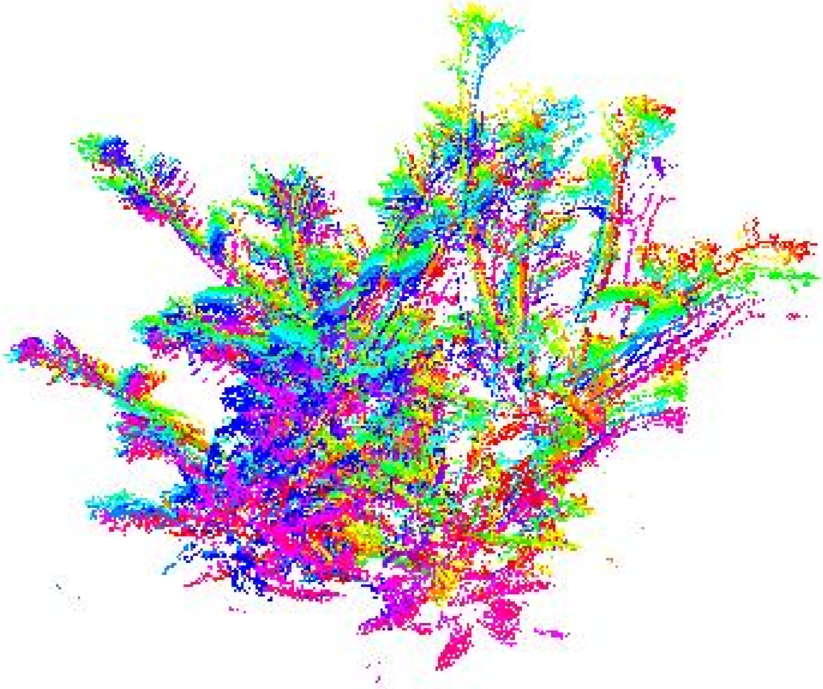
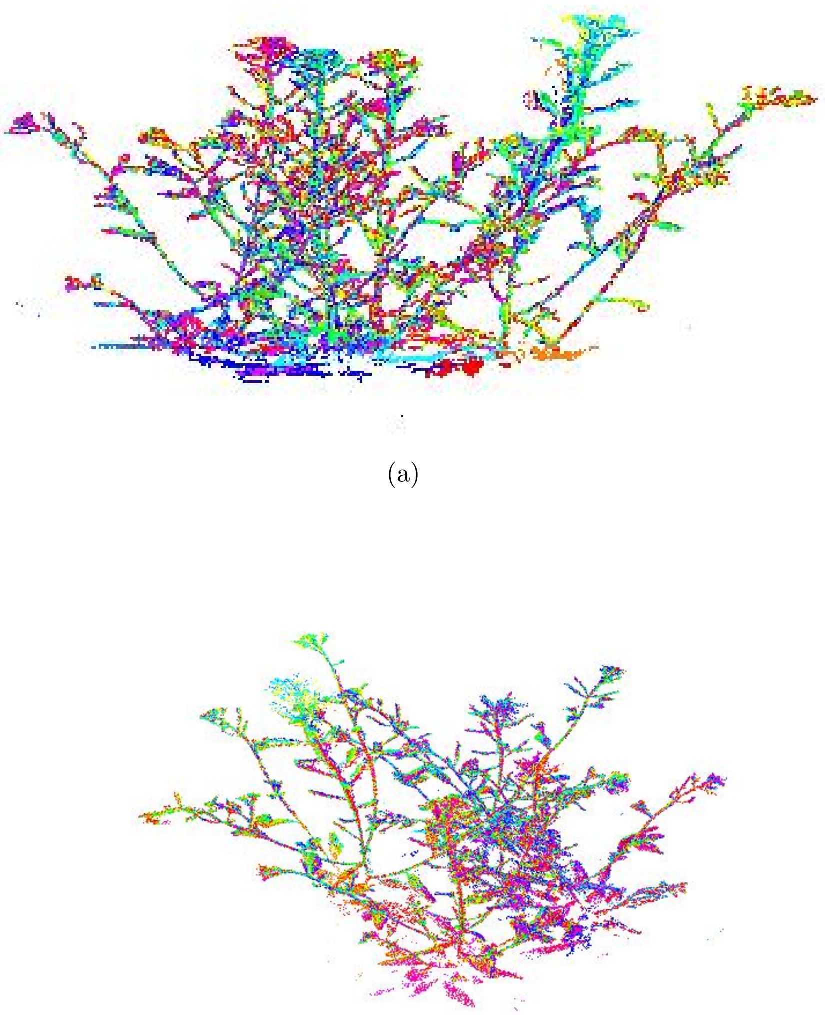


Figure 4.4: 12 scans of the Arabidopsis plant, prior to registration, but with rotation and and translation pre-applied.



(a)

(b)

Figure 4.5: 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.

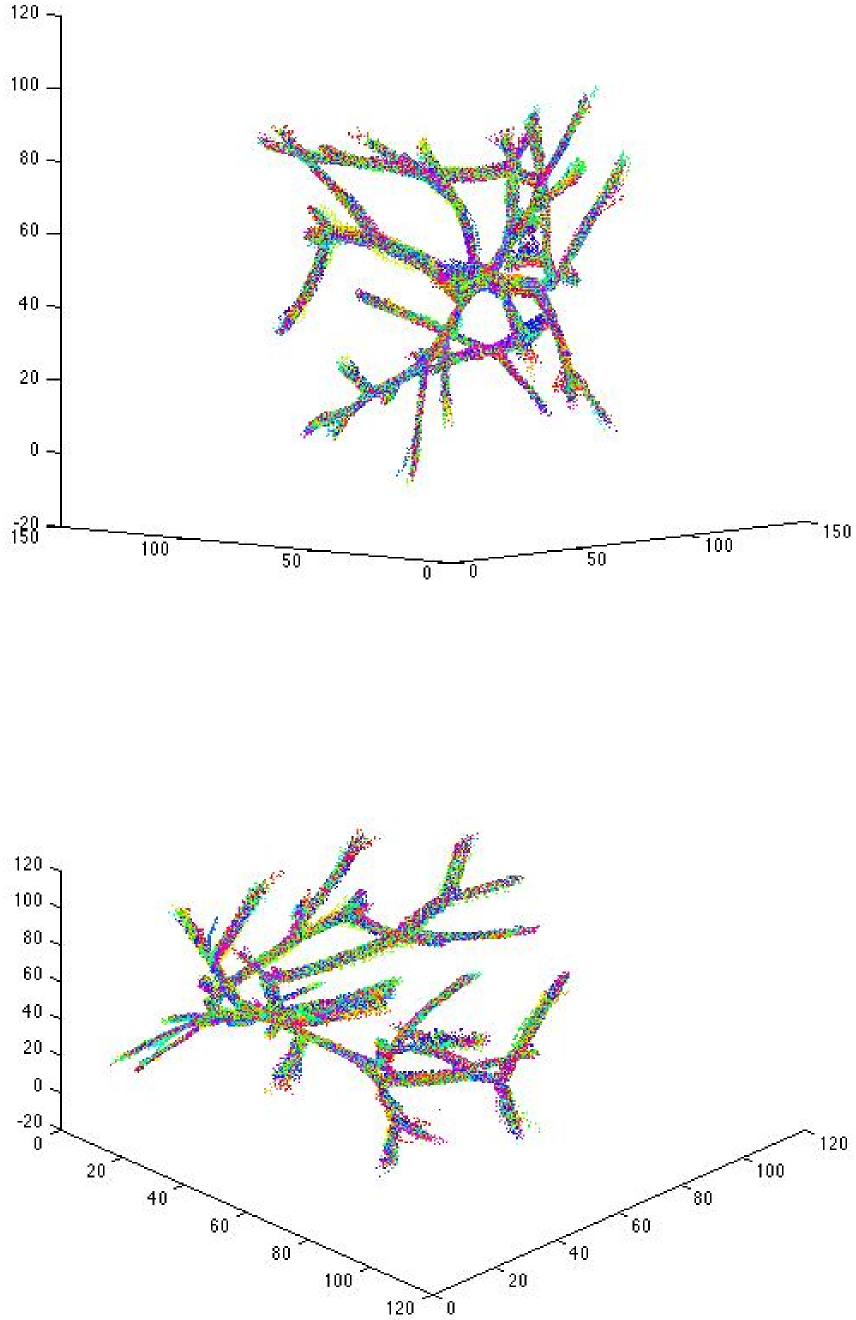


Figure 4.6: 20 synthetic scans of vascular data merged into a single point cloud using MNN, as seen from two viewpoints, front and back.

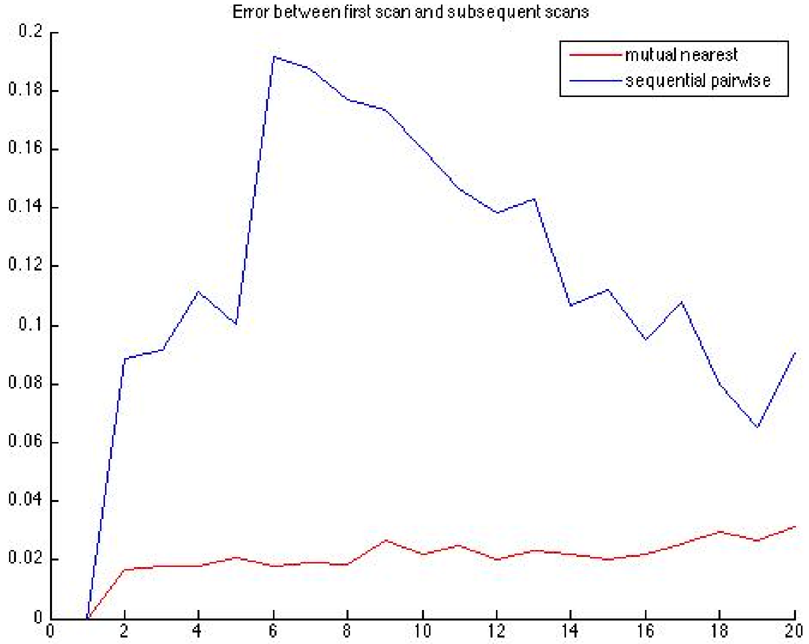


Figure 4.7: MNN versus sequential pairwise registration on vascular data, registering the first scan to each of the subsequent scans.

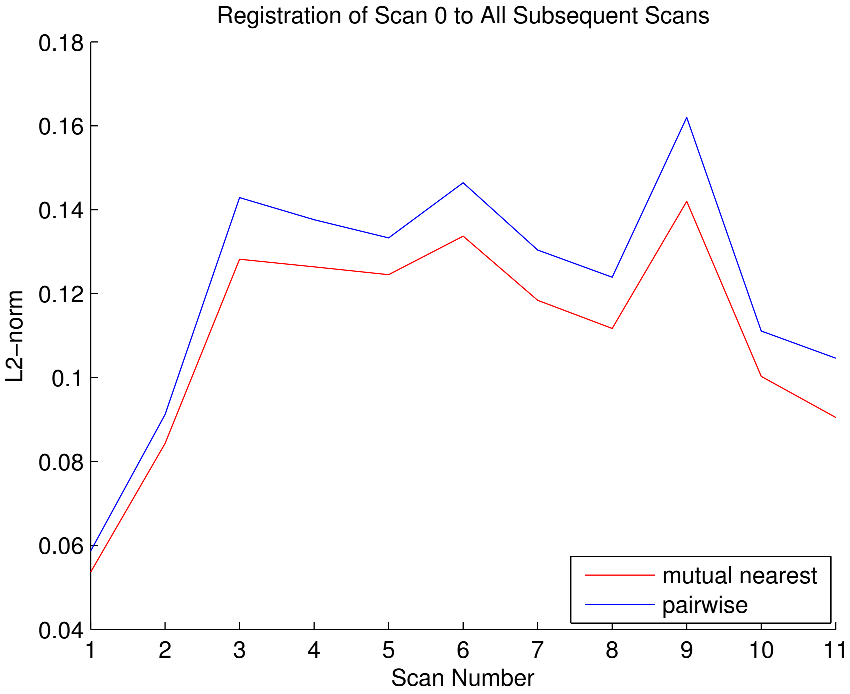


Figure 4.8: MNN versus sequential pairwise registration on the plant data.

4.4 Conclusions and Future Work

We have presented an approach to merge multiple scans into a single point cloud. Our plan is to triangulate this cloud into a triangular mesh and measure the volume of the plant. We hope that the quantitative volumes of the plant over time comprise a growth metric for the plant that is both non-invasive and non-contact. We have chosen an effective algorithm to work on sparse plant data and built our model on top of it: it can handle thin, non-rigid objects. One possible future research direction may be computing a quantitative analysis of how many scans should be sufficient to reconstruct the plant within tolerable range (for example, using 8, 6 or 4 scans instead of 12 scans), and we are also working on faster methods for registration using approximations of CPD or even a coarse-to-fine scheme, as our current solution will take hours to register a pair of Arabidopsis scans. We also intend to use the junction points proposed by Chaudhury et al. [15] to obtain an initial alignment more efficiently.

Chapter 5

Kernel Density Filtering for Noisy Point Clouds in One Step

We present a method for filtering noisy point clouds, specifically those constructed from merged depth maps as obtained from a range scanner or multiple view stereo (MVS), applying techniques that have previously been used in finding outliers in clustered data, but not in MVS or range scanning. This work initially fit well with the work presented in Chapter 4, when we were using an earlier version of the registration algorithm and this filtering aided in the convergence of the calculation. However, when we switched to a global method (as presented in Chapter 4) the filtering only added minimal improvement to the registration at significant computational cost. That said, computing a GMM for a large point cloud is very CPU intensive, taking 12 hours for a single Arabidopsis scan. Using a kernel density method like the one described in this chapter is much simpler and will yield results in a more efficient fashion than traditional GMMs.

We estimate the probability density function (PDF) over the space of observed points via a technique called kernel density estimation. We utilize Mahalanobis distance and a variable bandwidth for weighting kernels accordingly, based on the nature of neighbouring points. Further, we incorporate a distance metric called the Reachability Distance that, as we show in our results, gives better discrimination than a classical Mahalanobis distance-based metric. With the addition of this nearest neighbour metric, we can produce results that are ready for meshing without any post-processing of the cloud. We mesh our filtered point clouds using a traditional surface fitting technique that is unequipped to deal with noise to demonstrate the efficacy of our method.

5.1 Introduction

Following a renaissance in energy-based MVS methods in the literature, there has been a return to methods that merge depth maps from multiple views to generate a representative point cloud. This change in methodology can be traced back to the work of Goesele et al. [25], whose simple method generated depth maps for each camera using adjacent views, and then merged those depth maps using third-party software designed to merge range images into a complex model. Prior to this, the MVS literature was dominated by methods that evaluated photo-consistency over a dense grid and then used an energy minimization technique to extract a representative surface, which is often the minimal surface.

A cursory glance at recent results on the Middlebury multi-view stereo data sets [25] indicates incredible improvement of modern stereo matching algorithms over their predecessors. Such advancements are possible because of the improvements in disparity map generation like ordering constraints, bi-directional image matching, etc. That said, merging multiple depth maps and fitting a surface to the resultant point cloud remains a challenging endeavour, at least partially because of the presence of outliers and the general location of these outliers. These outliers can be very difficult to filter, as outlier clusters can occur both near to and far away from the true surface. Mahalanobis distance-based density estimation cannot correctly identify points that are close to the centre of these outlier clusters as not being part of the surface.

The goal of this algorithm is to demonstrate that it is possible to fit a surface to a point cloud with very large quantity of outliers (a ratio of 5:1 outliers to inliers), by filtering using anisotropic kernel density estimation with variable bandwidth, and subsequently fitting a surface to this filtered cloud using standard surface meshing software [17]. We take clean, merged point clouds and populate them with noise. This way, we know the ground truth and can thus quantify this method's ability to discriminate inliers from outliers with the Receiver Operating Characteristic (ROC) curves [56].

Our method uses a process called kernel density estimation to construct a probability density function over the space of discrete data that we obtain from measured data. The target application is filtering point clouds obtained from MVS data. Xi et al. [75] and Schall et al. [63] utilize a similar process, but our method differs from theirs in that we use a variable bandwidth based on the nearest neighbour of each point that contributes to a density estimate, as inspired by [37, 43, 68]. These works utilize variable bandwidths in finding outliers in clustered data. Our key observation is that these metrics are also very effective in filtering MVS and range scan data. Mahalanobis distance works well for discrimination of points near the true surface, in our experience better than Euclidean distance-based methods, but is prone to accept false positives near the centre of a cluster of outliers. Xi et al. [75] note that, when necessary, they apply their filter repeatedly until they get a result that is visually acceptable. Our method provides sufficient discriminatory ability such that it can be applied to a noisy point cloud once and the result can be meshed as is. We quantify our results and compare the discriminatory ability of our method with the Mahalanobis distance-based method used in [75].

5.2 Previous Work

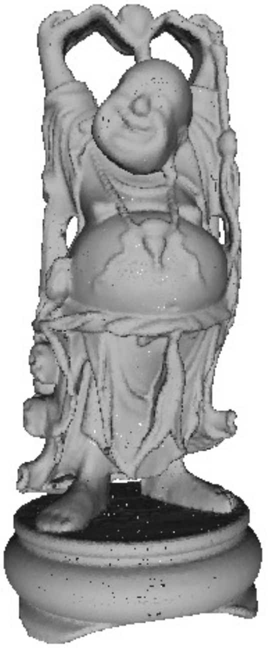
Lu et al. [44] utilize tensor voting (TV) with a minimal surface-based fitting scheme to reconstruct surfaces from highly noisy (1:1 signal-to-noise ratio) point clouds. They use the level set formulation of Zhao et al. [79] to evolve an initial implicit surface to fit the points, but add an extra term to influence the motion in the direction of the tensor.

When working with noisy data, one way to deal with outliers is to average them out. This is the method utilized by Goesele et al. [25]. The software they use, *VRIP*, converts each depth map into a signed distance function (SDF), and then merges these SDFs using a weighted averaging based on the angle between the observed points and the sensor in each depth map. The area over which a point can be averaged with another point is referred to as its “ramp”. Such a method results in smoothing of observations, and subsequent depth based methods that explicitly filter out a subset of the point cloud construct a surface from actual, unsmoothed observations.

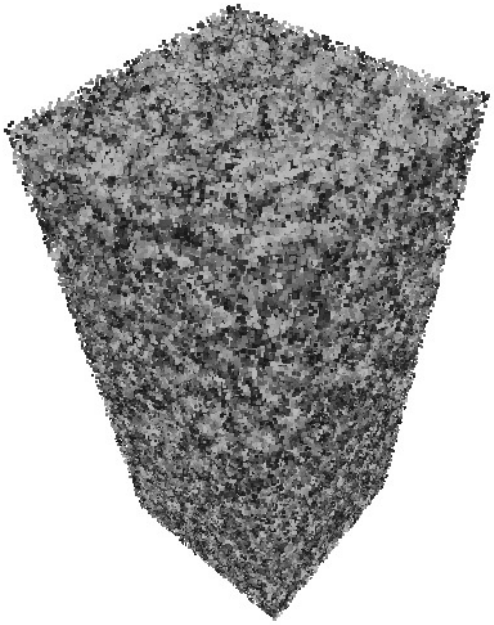
Li et al. [40] identify “tracks”, matched features that are found in at least three different views of an object, and then use bundle adjustment to recover the 3D point. If the reprojection error is above a certain threshold in one of the images, the point in question is discarded.

Campbell et al. [13] keep multiple hypotheses for prospective matches, and then use a spatial consistency measure in a Markov Random Field minimization scheme to recover better matches. If a point's hypotheses are not spatially consistent with its neighbours, it is discarded. Bradley et al. [6] construct point clouds using multi-scale matching and then use an iterative filtering method for outlier detection on the resultant point cloud. They compute the projection of a point and its neighbours to a plane and then evaluate the fit using a density function.

Xi et al.'s [75] work is the reference point for our method. They merge depth maps constructed from multiple views and use an anisotropic kernel density estimation method combined with a projected line search to obtain the maximum along each normal to find the maximum area of density on each normals path. We forego the use of reprojection error and "ramps" and instead use a density estimator to determine the quality of an observation. Schall et al. [63] also use an anisotropic kernel for filtering. Their method is similar to Xi et al. [75] in that they use an iterative method to move points along the normal direction to areas of maximal density. Further, they eliminate noise from high quality scans, and generate smooth surfaces from very high quality scans. Our focus is somewhat different, in that we study the circumstance where the quantity of outliers equals or exceeds the number of inliers, testing the ability to discern between inliers and the types of outliers that we see in photometric stereo, i.e. those that are both clustered near and far from the true surface. Further, we wish to simply filter the point cloud, as opposed to iteratively shifting the points to the area of highest density along the normal.

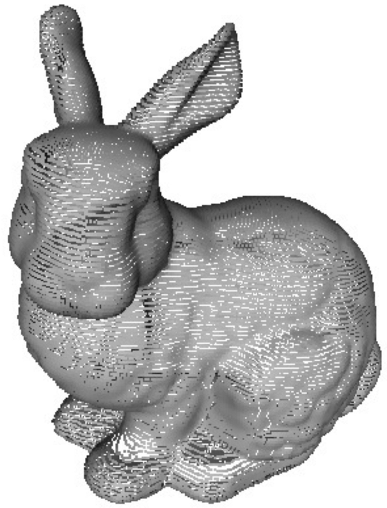


(a) Buddha statue.

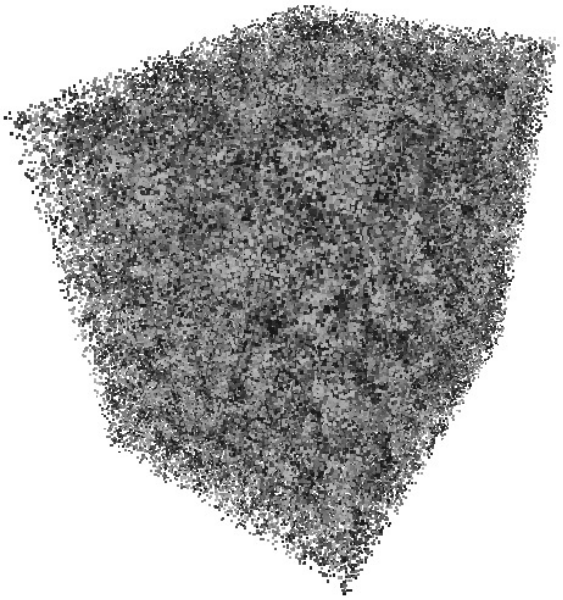


(b) Statue, noise added.

Figure 5.1: The *Buddha* point cloud with 5:1 ratio of noise to inliers added.



(a) Bunny.



(b) Bunny, noise added.

Figure 5.2: The *Bunny* point cloud with 5:1 ratio of noise to inliers added.

We assert that if a slightly better density estimator is used, one that uses both Mahalanobis distance, an adaptive bandwidth and the reachability distance utilized previously for clustering by Breunig et al. [8], we can mesh resultant point clouds without any further complication. We use a more advanced nearest neighbour metric, point clouds from range scans and MVS can be filtered without the projected line search while yielding an output cloud that can be meshed using an off-the-shelf method without any other pre-processing. We test this method on both the *Bunny* and *Buddha* data sets, as seen in Figures 5.2 and 5.1 respectively, where we can see that five randomly generated outliers have been added to the bounding volume for every point on the surface.

5.3 Obtaining a Probability Density Function from Measured Data

The easiest way to construct a probability density function from a set of points is a binning approach, similar to the construction of a histogram. Consider the problem in 1D: Say we have a set of measured data $X = [x_1, x_2, \ldots, x_n]$ where each data point x_i is a scalar-valued observation. If we construct a set of k bins and simply count the number of items that fall within each bin, we can easily construct a histogram.

A number of questions arise:

- How many bins should we use?
- Where should our bins start and end?
- How does this strategy scale in higher dimensions, i.e., how do we determine the orientation of the bins in multiple dimensions?

The next section introduces a better method for constructing a PDF.

5.3.1 Kernel Density Functions

Kernel density functions propose to solve the problem of obtaining a probability density function in a different way. Instead of creating arbitrary bins of data, the density is instead evaluated at each point, using the distance to neighbouring points as input to a kernel function, the most commonly used of which is the Gaussian kernel [75]

$$K(x) = \frac{1}{(2\pi)^d} \exp\left(-\frac{x}{2}\right), \tag{5.1}$$

where d is the dimension. This is referred to as the Parzen Window technique [54]. For each point $x_i \in X$, we rely on all points within a predefined radius to calculate the density of x_i using the kernel $K(x)$

$$f(x_i) = \frac{C_{k,d}}{n \cdot h} \sum_{j=1}^{n} K\left(\frac{||x_i - x_j||^2}{h}\right), \tag{5.2}$$

where the points x_j are the n neighbouring points of x_i within radius r, $C_{k,d}$ is a weight constant and h is the bandwidth.

Intuitively, $f(x_i)$ will be close to 1 if the sum of distances between x_i and its neighbouring points is small compared to the bandwidth. In other words, areas that contain a large number of points inside of their radius will yield a large density estimate and thus are more likely to be considered inliers than outliers.

5.3.2 Mahalanobis Distance

Xi et al. [75] utilize a more advanced method for filtering, based on the observation that the distribution of noise in point clouds tends to be anisotropic in nature. Thus, they evaluate an anisotropic kernel f of fixed radius r and shape at each point x to estimate its density utilizing its neighbouring points within distance r. Instead of using the L^2 distance between points within r, they find the distance to the centre of mass by making use of

$$f(x) = \frac{C_{k,d}}{n} \sum_{i=1}^{n} K(d_\Sigma(x, x_i)), \tag{5.3}$$

where the kernel K is as defined previously. $d_\Sigma(\cdot, \cdot)$ is the Mahalanobis distance, which is defined as

$$d_\Sigma(x, x_i) = \left((x - x_i)^T H^{-1} (x - x_i)\right)^{1/2},$$

where the covariance matrix

$$H = DD^T,$$

can be constructed using

$$D = (x_1 - x, x_2 - x, \cdots, x_n - x).$$

They find the location of highest density within the neighbourhood (r) and then use the distance to this point as the distance for the kernel to evaluate. This method discriminates between inliers and outliers when near the “true” surface much more robustly than the basic kernel density estimation method that we defined in Equation (5.2). We found that this method still had good discriminatory power when the signal-to-noise ratio was 1:10, i.e. we added 10 randomly generated outliers for each inlier in the point cloud’s bounding volume. Their method is prone to errors when there are clusters of outliers in a small area, a common occurrence.

This method isn't new, it has been used in outlier estimation when dealing with clusters of data in the past. Likewise, our work is based on well-founded principles that are known in the literature. It hasn't been applied to this domain, and its discriminatory ability is notable.

5.3.3 Local Bandwidth Estimation

The idea to use a local estimate of bandwidth $h(x_i)$ comes from Latecki et al. [37] who applied it to detecting outliers in clusters of data. It effectively unweights points whose n-th nearest neighbour is any larger than a very small distance from the point itself. Applying this to surface fitting makes a lot of sense, as points that lay near the true surface should have a neighbour in its very near vicinity.

5.4 Methodology

Based on the density of the nearest neighbour, we can weight each kernel accordingly. In other words, if a neighbouring point x_i itself has a low density, the bandwidth $h(x_i)$ will be lower and thus the contribution to the magnitude of $f(x)$ will be smaller than an equally distant point that exists in an area of higher density.

$$f(x_i) = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{h(x_j)^d}K\left(\frac{d_{\Sigma}(x_j, x_i)}{h(x_j)}\right) \tag{5.4}$$

The bandwidth is the distance of the nearest neighbour to x_i, the dimension d is 3, $n = size(NN(x_i))$, where $NN(\cdot)$ is the set containing the nearest neighbours of x_i, the points within the radius r. The Gaussian kernel is as defined previously. The bandwidth,

$$h(x_j) = \min(d_\Sigma(x_j, x_k))$$

where $x_k \in NN(x_j)$, i.e., x_k is the nearest neighbour to x_j, when Mahalanobis distance is used to determine the "closeness" of two points. This method differs from the anisotropic kernel density method described in the previous section in that the density of a point *relies on the density of its neighbouring points.* In other words, we could have a point x_i and its nearest k points, and in the previous method, its kernel density estimate $q(x_i)$ would be the same irrespective of the points surrounding these neighbours.

5.4.1 Reachability Distance

We can extend this idea further by replacing the numerator of Equation (5.4) with a more robust metric called the *reachability distance*, where

$$\mathbf{rd}(x_i, x_j) = \max(d_\Sigma(x_j, x_i), d_\Sigma(x_k, x_j)). \tag{5.5}$$

This yields

$$f(x_i) = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{h(x_j)^d} K\left(\frac{\mathbf{rd}(x_i, x_j)}{h(x_j)}\right), \tag{5.6}$$

and is thus composed of both the distance from x_i to its neighbours x_j, and distances of neighbouring points x_j to their nearest neighbour (x_k). If $rd(x_i, x_j) = d_\Sigma(x_i, x_j)$ then the inside of K is $-\frac{d_\Sigma(x_i,x_j)}{2d_\Sigma(x_j,x_k)}$ where the numerator is greater than the denominator, and thus yields an increasingly smaller value as this difference increases when evaluated by the exponential. If $rd(x_i, x_j) = d_\Sigma(x_j, x_k)$ then the inside of the exponential is $-\frac{1}{2}$ and thus gives the minimal value.

If this neighbour x_j's nearest neighbour is quite far away, i.e., x_k is a somewhat "isolated" point, it gives us very little information about the nature of point x_i, as being in the same neighbourhood as a likely outlier is not a great clue that x_i is an inlier. If, however, the distance to x_j is larger than the distance to x_j's nearest neighbour, we can say that having such a point in the neighbourhood is good evidence, and it is thus more likely that x_i is indeed an inlier!

5.5 Kernel Density Filtering on Data with Additive Noise

We approach it first using anisotropic kernel density estimation, where we evaluate the density of the data within a set radius of each data point. Equation (5.3) evaluates the contribution of each point within this area, and takes into account the local density of each of these points as well.

We filter the signal by removing points whose density are below some threshold $\tau \in [0, 1]$. In practice, the median density is used. The method struggles with clusters of outliers though, and a substantial percentage of inliers are removed before the outliers fall below τ.

Interestingly, when we use the nearest-neighbour kernel density estimate, even in conjunction with the less discriminative L^2-norm, we still recover a much more accurate signal. That said, the performance of the anisotropic kernel density near the signal is better.

5.6 Results

We test our algorithm on the *Bunny* and *Buddha* data sets from Stanford and add random noise of a 5:1 ratio to determine the ability of our density-based method to discriminate between inliers and outliers. This means that for each point on the surface, we add five randomly generated outliers in the bounding box. To quantify the filter's ability to discriminate between an inlier, i.e. a member of the normal class ("NC") and an outlier ("OC"), we generate a ROC curve, which plots the detection rate (r_D) versus the false alarm rate (r_{FA})

$$r_D = \frac{TP}{TP+FN} \tag{5.7}$$

$$r_{FA} = \frac{FP}{FP+TN} \tag{5.8}$$

where *TP* is the number of true positives, *FN* is the number of false negatives, and *FP* is the number of false positives and *TN* is the number of true negatives.

The nature of these terms is explained the confusion matrix seen in Table 5.1. A perfect ROC curve has an area of one beneath said curve.

	Predicted Outlier Class (OC)	Predicted Normal Class (NC)
Actual Outliers	True Positive	False Negative
Actual Normal Class	False Positive	True Negative

Table 5.1: Confusion matrix describing the different classifications of inliers and outliers.

As can be seen from the ROC curves in Figures 5.4 and 5.3, our method of-

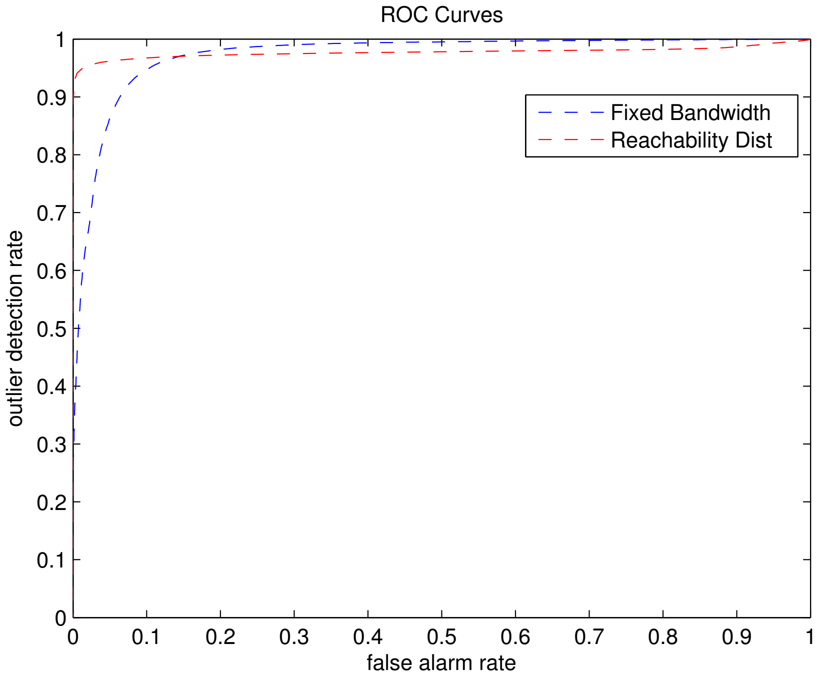


Figure 5.3: ROC curve for anisotropic filters on the *Bunny* point cloud with 5:1 noise. The reachability distance/adaptive method is in red.

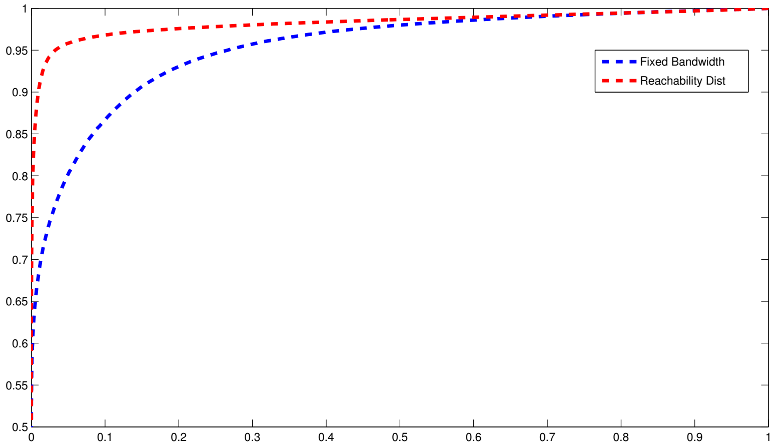
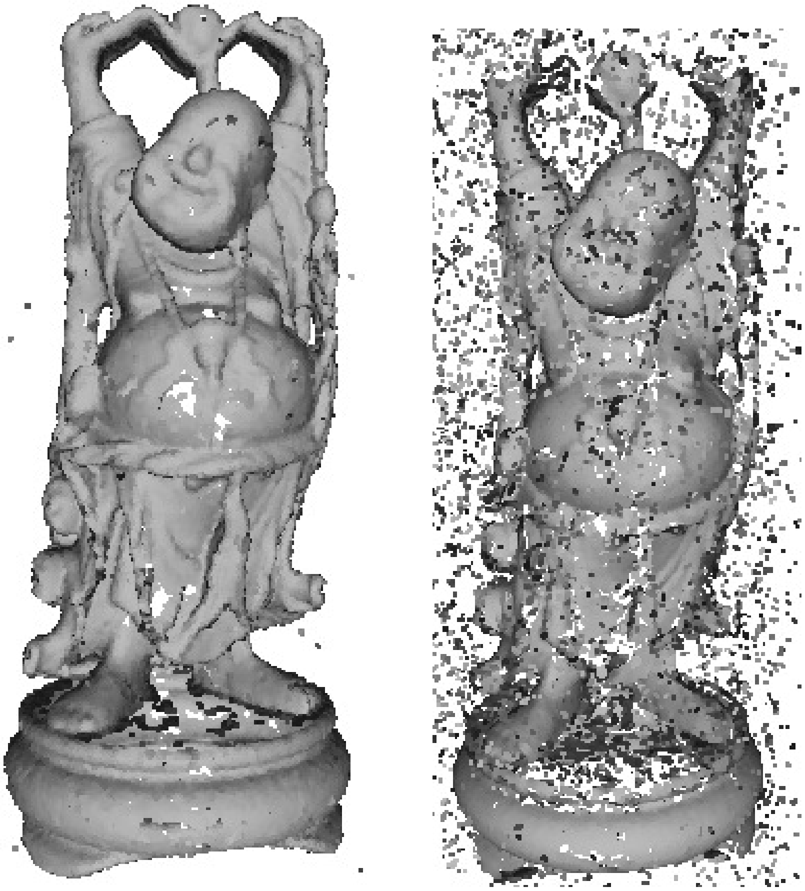


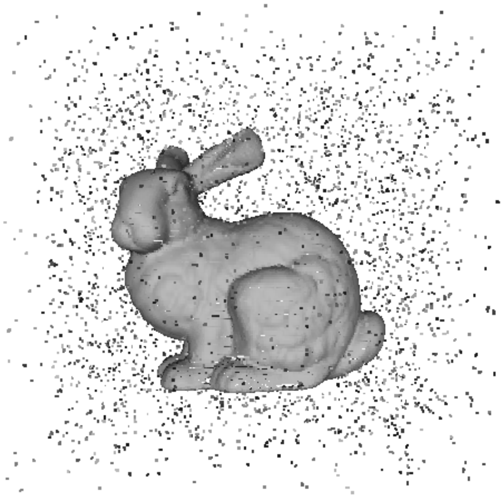
Figure 5.4: ROC curve for anisotropic filters on the *Buddha* point cloud with 5:1 noise. The reachability distance/adaptive method is in red.



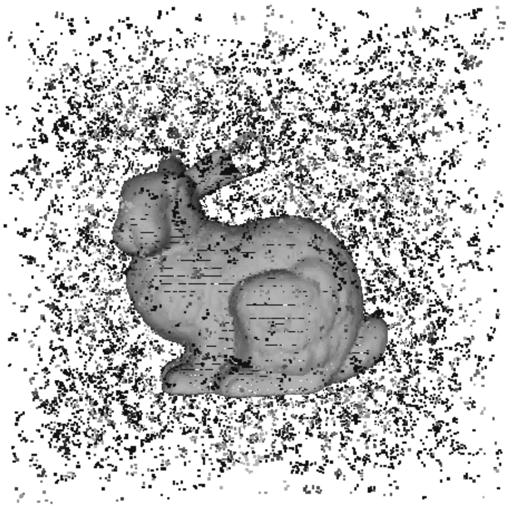
(a) Reachability distance. (b) Mahalanobis distance.

Figure 5.5: Buddha, filtered. The reachability distance-based filtering method leaves almost no outliers.

fers excellent discrimination between inliers and outliers in both circumstances. When one uses a strictly Mahalanobis-based density estimator on the *Bunny* data, a substantial portion of outliers remain. To make this more clear, we display the remaining points in the resultant cloud in Figure 5.6. There are far too many outliers in Figure 5.6b to allow for the fitting of a surface points, and true surface points are being thresholded as we remove more outliers. In Figure 5.6a, a few outliers remain, but they occur in such low densities that they do not interfere with the subsequent surface fitting. We meshed the points from Figure 5.6a in Figure 5.6c with the ball pivoting algorithm of Bernardini et al. [1].



(a) Reachability distance.



(b) Fixed Mahalanobis.

Figure 5.6: The *Bunny* point cloud with 5:1 noise, filtered.

Likewise, we see that our method works extremely well on the *Buddha* data set. Most impressive is its ability to handle the thin part of the statue above the head. As we see in Figure 5.5, despite the addition of 5:1 noise, we can still perform an accurate reconstruction of the surface with our method.

We also experimented with using the k nearest neighbours of x_i for estimation of the Mahalanobis distance in the above examples, but we found that the slight increase in discrimination was not worth the added time complexity.

Adjusting the area of support, r, has an effect on the nature of the filtering. If the algorithm is having trouble removing outliers near the surface, it may be useful to *decrease* the radius of the points that contribute to the density estimate. Increasing r will include more points with a larger distance to a point x_i if it is an outlier, but it will do the same for an inlier. The radius should be large enough to contain a sufficient number of points (for our purposes, ≥ 50), but small enough that the density estimates are excessively “smooth”. Ideally, the estimation of f for any outlier near the surface will include a large sampling of inliers (i.e. actual surface points) to weight the centre of mass correctly, yielding a small distance to points on the surface, and a large distance to outliers. Further, if there are thin areas on the surface, a small r can be useful to ensure that a density estimate at x is only influenced by its neighbours on the surface, not close by points that belong to a different part of the surface.

5.7 Conclusions and Future Work

We have demonstrated that our filtering method performs well on both the Buddha and Bunny data sets when they are corrupted by large amounts of noise. In reality, point clouds obtained from MVS or range scanning are not even near as noisy as our two corrupted point clouds. That said, the nature of the noise may be such that noise resides near the true surface, and it will thus be more difficult for the method to decipher whether a point is an inlier or an outlier.

In the future, we would like to automate the process of fitting a surface to our filtered cloud, possibly by including our density estimate in a surface evolution scheme, similar to the level set-based method of Zhao et al. [79], with an extra term for density. It might be effective to include the confidence measure of each point in the point cloud from the stereo matching process. In the end, the goal is to obtain extremely accurate multi-view surface reconstructions of objects from multiple views, and a filtering method like the one we've presented is a step in that direction.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Most meaningful problems in computer vision are under-constrained, and this is true of surface fitting to point clouds: there are an infinite number of possible surfaces that fit to a set of scattered points [27]. It is thus extremely important to leverage the best tools available for solving the optimization problems that arise. We frame problems in a global context when it is possible to do so, and we attempt to utilize the appropriate models for capturing the true nature of the data with which we work. The methods that we described in the preceding chapters for filtering, registering and comparing 3D point clouds adhere to these ideas.

Chapter 3 outlined our method for quantifying performance of physicians in segmenting tumours from CT data. We based our method on Boykov and Lempitsky's [38] method, and by having a guarantee that our watertight surfaces are minimal, we can make meaningful insights into the variability of tumour segmentation.

Chapter 4 described our method for registration of multiple thin point clouds. By iteratively registering each scan to a target scan that captures all scans, we removed the issues with drift that are endemic to sequential, local registration methods for multiple scan registration.

Finally, Chapter 5 described a method for filtering very noisy point clouds. By using reachability distance for estimating kernel density, we are able to remove far more outliers in a single step than we would be able to using a regular Mahalanobis-based kernel density estimate.

6.2 Future Work

We wish to continue our work in point cloud registration, specifically registration of point clouds of thin structures. We will be exploring use of the bounded generalized Gaussian mixture model [51] to capture thin data structures more effectively. We would like to investigate methods for faster registration of large point clouds, either through more efficient computation of mixture models or by utilizing kernel density methods. It should also be noted that many of these algorithms are parallelizable to a degree, and this strategy should be explored as well.

We also wish to quantify physician performance on different types of tumours, in hopes of obtaining a measure of what types of tumours are the most difficult to segment. Finally, we would like to automate the process of fitting a surface to highly noisy point clouds by incorporating our density estimate in a surface evolution based scheme.

Bibliography

[1] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5, 10 1999.

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

[3] V. Blanz, A. Mehl, T. Vetter, and H. p. Seidel. A statistical method for robust 3d surface reconstruction from sparse data. In *In Int. Symp. on 3D Data Processing, Visualization and Transmission*, pages 293–300, 2004.

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

[5] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 26–33, Washington, DC, USA, 2003. IEEE Computer Society.

[6] D. Bradley, T. Boubekeur, T. Berlin, and W. Heidrich. Accurate multiview reconstruction using robust binocular stereo and surface meshing. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[7] X. Bresson, S. Esedolu, P. Vandergheynst, J.P. Thiran, and S. Osher. Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and vision*, 28(2):151–167, 2007.

[8] M.M. Breunig, H.P. Kriegel, R.T. Ng, and J. Sander. Lof: Identifying density based local outliers. In *Proceedings of the ACM SIGMOD Conference*, Dallas, TX, May 2000.

[9] M.A. Brophy, S.S. Beauchemin, and J.L. Barron. Kernel density filtering for noisy point clouds in one step. In *Proceedings of the Irish Machine Vision and Image Processing Conference*, pages 27–34, August 2015.

[10] M.A. 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. IEEE, June 2015.

[11] T. Brox and D. Cremers. On local region models and a statistical interpretation of the piecewise smooth mumford-shah functional. *International journal of computer vision*, 84(2):184–193, 2009.

[12] N.G. Burnet, S.J. Thomas, K.E. Burton, and S.J. Jefferies. Defining the tumour and target volumes for radiotherapy. *Cancer Imaging*, 4:153–161, October 2004.

[13] N.D.F. Campbell, G. Vogiatzis, C. Hernandez, and R. Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *Proceedings of the European Conference on Computer Vision ECCV*, volume 1, pages 766–779, 2008.

[14] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22:61–79, 1995.

[15] A. Chaudhury, M.A. Brophy, and J.L. Barron. Junction point detection in 3d plant point clouds using multi-modal analysis and tls approximation. In *submitted*, 2014.

[16] 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.

[17] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. Meshlab: an open-source mesh processing tool. In *Eurographics Italian Chapter Conference*. Eurographics, 2008.

[18] P. Cignoni and F. Fabio Ganovelli. Vcg library. http://vcg.isti.cnr.it/vcglib, 2013.

[19] R.T. Clark, R.B. MacCurdy, J.K. Jung, J.E. Shaff, S.R. McCouch, D.J. Aneshansley, and L.V. Kochian. Three-dimensional root phenotyping with a novel imaging and software platform. *Plant Physiology*, 156(2):455–465, 2011.

[20] C. Couprie, X. Bresson, L. Najman, H. Talbot, and L. Grady. Surface reconstruction using power watershed. In *Mathematical Morphology and Its Applications to Image and Signal Processing*, pages 381–392. Springer, 2011.

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

[22] J. Digne. An analysis and implementation of a parallel ball pivoting algorithm. *Image Processing Online*, 4:149–168, 07 2014.

[23] A.W. Fitzgibbon. Robust registration of 2d and 3d point sets. In *Proceedings of the British Machine Vision Conference*, BMVC, pages 43.1–43.10, 2001.

[24] P. Getreuer. Rudin-Osher-Fatemi Total Variation Denoising using Split Bregman. *Image Processing On Line*, 2:74–95, 2012.

[25] M. Goesele, B. Curless, and S.M. Seitz. Multi-view stereo revisited. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2402–2409, 2006.

[26] T. Goldstein, X. Bresson, and S. Osher. Geometric applications of the split bregman method: Segmentation and surface reconstruction. *Journal of Scientific Computing*, 45(1-3):272–293, 2010.

[27] A.A. Goshtasby. *2D and 3D Image Registration: For Medical, Remote Sensing, and Industrial Applications*. Wiley-Interscience, 2005.

[28] T. Heimann, B. van Ginneken, and M. Styner. Comparison and evaluation of methods for liver segmentation from ct datasets. *IEEE Trans. on Medical Imaging*, 28(8):1251–1265, 2009.

[29] A. Hornung and L. Kobbelt. Robust reconstruction of watertight 3 d models from non-uniformly sampled point clouds without normal information. In *Symposium on Geometry Processing*, pages 41–50. Citeseer, 2006.

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

[31] M.G. Jameson, L.C. Holloway, P.J. Vial, S.K. Vinod, and P.E. Metcalfe. A review of methods of analysis in contouring studies for radiation oncology. *Journal of medical imaging and radiation oncology*, 54(5):401–410, 2010.

[32] P. Jassi and G. Hamarneh. VascuSynth: Vascular Tree Synthesis Software. *Insight Journal*, January-June:1–12, 2011.

[33] B. Jian and B.C. Vemuri. Robust point set registration using gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1633–1645, August 2011.

[34] Y. Jian, X. Bresson, T. Goldstein, and S. Osher. A fast variational method for surface reconstruction from sets of scattered points. *CAM Report*, 10(1), 2010.

[35] S. Jin, R.R. Lewis, and D. West. A comparison of algorithms for vertex normal computation. *The Visual Computer*, 21(1-2):71–82, 2005.

[36] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[37] L.J. Latecki, A. Lazarevic, and D. Pokrajac. Outlier detection with kernel density functions. In *Proceedings of Machine Learning and Data Mining in Pattern Recognition*, pages 61–75, July 2007.

[38] V. Lempitsky and Y. Boykov. Global optimization for shape fitting. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[39] M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings of Computer Graphics*, ACS, pages 31–42, 1996.

[40] J. Li, E. Li, Y. Chen, L. Xu, and Y. Zhang. Bundled depth-map merging for multi-view stereo. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:2769–2776, 2010.

[41] 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.

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

[43] D. O. Loftsgaarden and C. P. Quesenberry. A nonparametric estimate of a multi-variate density function. *Annals of Mathematical Statistics*, 36:1049–1051, 1965.

[44] H. Lu, H. Zhao, M. Jiang, S. Zhou, and T. Zhou. A surface reconstruction method for highly noisy point clouds. In *Proceedings of Variational, Geometric, and Level Set Methods in Computer Vision, LNCS*. Springer, 2005.

[45] S. Martin, M.A. Brophy, G. Rodrigues, D. Palma, A. Louie, B. Yaremko, B. Ahmad E. Yu, S.S. Beauchemin, J.L. Barron, and S. Gaede. A proposed framework for consensus-based lung tumour volume autosegmentation in four-dimensional computed tomography imaging. *Physics in Medicine and Biology*, 60(4):1497–1518, February 2015.

[46] S. Martin, C. Johnson, M.A. Brophy, D.A. Palma, J.L. Barron, S.S. Beauchemin, A.V. Louie, E. Yu, B. Yaremko, B. Ahmad, G.B. Rodrigues, and S. Gaede. Impact of target volume segmentation accuracy and variability on treatment planning for 4d-ct-based non-small cell lung cancer radiotherapy. *Acta Oncologica*, 54(3):322–332, March 2015.

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

[48] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5):577–685, 1989.

[49] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, December 2010.

[50] 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, pages 1009–1016, 2006.

[51] T.M. Nguyen, Q.M.J. Wu, and H. Zhang. Bounded generalized gaussian mixture model. *Pattern Recognition*, 47(9):3132–3142, 2014.

[52] R. Pan and V. Skala. Surface reconstruction with higher-order smoothness. *The Visual Computer*, 28(2):155–162, 2012.

[53] 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):63, 2012.

[54] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

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

[56] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.

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

[58] M. Rousson and N. Paragios. Shape priors for level set representations. In *In ECCV*, pages 78–92. Springer, 2002.

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

[60] 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'09, pages 1848–1853, Piscataway, NJ, USA, 2009. IEEE Press.

[61] 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.

[62] P. Savadjiev, F.P. Ferrie, and K. Siddiqi. Surface recovery from 3d point data using a combined parametric and geometric flow approach. In *Energy Minimization Methods in Computer Vision and Pattern Recognition, 4th International Workshop, EMMCVPR 2003, Lisbon, Portugal, July 7-9, 2003, Proceedings*, pages 325–340, 2003.

[63] O. Schall, A. Belyaev, and H.-P. Seidel. Robust filtering of noisy scattered point data. In *IEEE/Eurographics Symposium on Point-Based Graphics*,

pages 71–77, Stony Brook, New York, USA, 2005. Eurographics Association.

[64] 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.

[65] 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*, WBIR, pages 286–295, 2012.

[66] M. Sormann, C. Zach, J. Bauer, K. Karner, and H. Bishof. Watertight multi-view reconstruction based on volumetric graph-cuts. In *Image Analysis*, volume 4522 of *Lecture Notes in Computer Science*, pages 393–402. Springer Berlin Heidelberg, 2007.

[67] Philips Medical Systems. Pinnacle treatment planning system. http://www.healthcare.philips.com/main/products/ros/products/pinnacle3, 2015.

[68] G. R. Terrell and D. W. Scott. Variable kernel density estimation. *The Annals of Statistics*, 20(3):1236–1265, 1992.

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

[70] 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)*, pages 558–569, 2004.

[71] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Computer Graphics Proceedings*, ACS, pages 311–318, 1994.

[72] A. Vasilevskiy and K. Siddiqi. Flux maximizing geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1565–1578, 2001.

[73] L.A. Vese and T.F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *International journal of computer vision*, 50(3):271–293, 2002.

[74] G. Vogiatzis, P.H.S. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 391–398. IEEE, 2005.

[75] Y. Xi, Y. Duan, and H.K. Zhao. A nonparametric approach for noisy point data preprocessing. *International Journal of CAD/CAM*, 9(1):31–36, 2009.

[76] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust tv-l 1 range image integration. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[77] 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, November 2012.

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

[79] H.K. Zhao, S. Osher, B. Merriman, and M. Kang. Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method. *Computer Vision and Image Understanding*, 80:295–319, 2000.

Vita

NAME:	Mark Brophy
PLACE OF BIRTH:	St. John's, NL
YEAR OF BIRTH:	1982
POST-SECONDARY EDUCATION AND DEGREES:	Memorial University of Newfoundland St. John's, NL 2000-2005 B.Sc. (hons.)
	Memorial University of Newfoundland St. John's, NL 2005-2007 M.Eng.
RELATED WORK EXPERIENCE:	CTO MarketClouds / Cryptiv 2012-2014
	Lead Developer Global Ad Source 2014-Present
	Teaching Assistant University of Western Ontario 2008 - 2012