Quantitative Analysis of Plant Root System Architecture

James Johnson

Table of Contents

Abstract 8

Acknowledgments 9

1. Introduction 10

1.1 Contributions 14

1.2 List of Publications 14

2. Literature Review 15

2.1 Origin of Genotyping and Phenotyping 15

2.2 Phenotyping of Root Traits 17

2.2.1 Invasive Techniques 18

2.2.1.1 Root Washing 18

2.2.1.2 Rhizotrons 19

2.2.1.3 Hydroponics and Transparent Gels 19

2.2.2 Non-Invasive techniques 19

2.2.2.1 Manual Segmentation 20

2.2.2.2 Review of semi-automated and automated segmentation methods 21

2.3 Review of Image Skeletonisation and Thinning Methods 27

2.3.1 Morphological Thinning 28

2.3.2 Root System Skeletonisation 29

2.4 Root System Markup Language 30

2.5 Summary 32

3. Root Skeletonisation via Non-Parametric Curve Fitting 34

3.1 The root skeletonisation problem 34

3.1.1 X-ray CT imaging of soil pots containing plant root systems 36

3.1.1.1 Image Acquisition 36

3.1.1.2 Segmenting roots from soil 36

3.1.2 A non-parametric curve-fitting approach to root skeletonisation 43

3.1.2.1 Overview 43

3.1.2.2 Distance Map Computation 44

3.1.2.3 Root Tip Detection Algorithm 46

3.1.2.4 Evaluation of the effect of morphological operators on root tip detection 50

3.1.2.5 Control point selection and non-parametric linear curve-fitting 52

3.1.2.6 Defining fitted curve smoothing parameter 54

3.1.2.7 Using Cross-Validation for automatic smoothing parameter estimation 55

3.1.2.8 Root System Fractalisation 57
3.2 Evaluation and Results 60
3.3 Discussion 66
3.4 Summary 68
4. Plagiotropic Root Phenotyping Using Active Contours 69
4.1 Motivation 69
4.2 Plagiotropic Roots and Control Points 71
4.3 Active Contours 73
4.4 Spline Fitting and Trait Extraction 82
4.5 Results and Discussion 85
4.6 Conclusions 92
5. Plant Root Registration Methods 93
5.1 Motivation 93
5.2 Global Registration using Soil Features 94
5.2.1 Image registration 94
5.2.2 Registering Roots in Soil 95
5.2.3 A Minimum Spanning Tree approach 99
5.3 Global Registration Results 109
5.4 Local registration and 4D quantification of Root System Architectures 113
5.5 Conclusions 119
6. Towards Time-Series assisted segmentation of Root Systems 120
6.1 Watershed based supervoxel segmentation 120
6.2 Discussion 128
6.3 Conclusions 131
7. Conclusions and Future Work 132
7.1 Review of Aims and Objectives 132
7.1.1 Semi-automated skeletonisation tool 132
7.1.2 Morphologically accurate approximation of the skeletal ground truth 133
7.1.3 Root image analysis pipeline integration 134
7.2 Future improvements and applications 135
Bibliography 137

List of Figures

Figure 2.1 RooTrack 22
Figure 2.2 Root System Analyser 22
Figure 2.3 SmartRoot 23
Figure 2.4 WinRhizo 23
Figure 2.5 RootScape 24
Figure 2.6 RootNav 25
Figure 2.7 RootReader3D 26
Figure 2.8 PlantRoot 26
Figure 2.9 DynamicRoots 27
Figure 2.10 Maximum Discs. 27
Figure 2.11 Example of skeletonisation tree and application of pruning algorithms 28
Figure 2.12 Root System Markup Language 31
Figure 3.1 X-ray CT scanners in the Hounsfield facility, University of Nottingham 37
Figure 3.2 Automation in the Hounsfield Facility. 38
Figure 3.3 Manual Segmentation of a root system using VG Studio Max 39
Figure 3.4 Morphological dilation effect on misshaped and fused roots 39
Figure 3.5 Smooth segmented root surface resulting from RooTrak Segmentation 40
Figure 3.6 Illustrates the emergence of false-positive vs true-positive root tips 41
Figure 3.7 Incorrect segmentation by RooTrak level-set speed function 41
Figure 3.8 An example of false-positive root tips generated by RooTrak output 42
Figure 3.9 Overview of the proposed curve-fitting root skeletonisation method 44
Figure 3.10 Manhattan, Euclidean and Chessboard distance 45
Figure 3.11 Illustration of the calculation of chessboard distance map 46
Figure 3.12 Example of spatial distribution of false-positive root tips. 47
Figure 3.13 Manual classification of root tips 48
Figure 3.14 Root tips heuristics 49
Figure 3.15 Effect of morphological pre-processing on the number of false-positive root tips 51
Figure 3.16 Comparison of linear interpolation with smoothed curve-fitting 53
Figure 3.17 Illustration of Tree-like vs Fractal-like representation of a root system 58
Figure 3.19 Curve-fitted root skeletons of the scanned plants in the validation experiment 61
Figure 3.20 Correlation between the output of RooTh and WinRhizo 63
Figure 3.21 Demonstration of Fiji's built-in Oval Tool 64
Figure 3.22 2D Skeleton output of the WinRhizo 65

Figure 3.23 Effect of horizontal roots on curve-fitted root skeleton.....67
Figure 4.1 Effect of side-sweep control points on curve-fitting skeleton.....71
Figure 4.2 Side Sweep Control Points. 72
Figure 4.3 Side Sweep control points (Alternative View) 73
Figure 4.4 Integration using finite difference for calculating ACM.....74
Figure 4.5 2D Active Contour Model over an inverse square law vector force field 76
Figure 4.6 Initialisation of the Active Contour Model.....77
Figure 4.7 Simulated Annealing of Active Contour Model 79
Figure 4.8 Energy Minimisation Function of an Active Contour Model Interface. 81
Figure 4.9 Highly-entangled sections of curve-fitted skeleton. 82
Figure 4.10 Extrapolating connecting lines for ACM 84
Figure 4.11 10 days old wheat plant.....86
Figure 4.12 Comparison between curve-fitting and ACM skeleton.....87
Figure 4.13 Comparison between ACM and curve-fitted skeleton in a single cross section. 88
Figure 4.14 Nitrogen Uptake Efficiency experiment.....90
Figure 4.15 Biological Experiment 91
Figure 5.1 Watershed segmentation of the CT slice of a soil column 96
Figure 5.2 Split and merge algorithm used to reduce the number of over-segmented regions.....97
Figure 5.3 Large Soil Particles in a soil column 98
Figure 5.4 Minimum Spanning Tree of a feature set of soil particles' centre points 100
Figure 5.5 Random Sample Consensus 102
Figure 5.6 Correspondence pair points generated by MST 106
Figure 5.7 False positivevs genuine pairs of matching correspondence points 107
Figure 5.8 Cross-section of incorrectly registered 3D volume.....108
Figure 5.9 Weighted sum of trilinear interpolation.....109
Figure 5.10 Comparison of TurboReg and 3D soil particle registration.....110
Figure 5.11 Transformation process of the artificially generated image stack. 112
Figure 5.12 Results of validation registration of artificially transformed image stack. 113
Figure 5.13 Runge's phenomenon effect on the accuracy of image registration. 116
Figure 6.1 Global Histogram of an Image Stack.....121
Figure 6.2 Fixed Width Tube Histogram 122
Figure 6.3 Example of 4D assisted root segmentation 124
Figure 6.4 Local histogram charts of sliding windows in a single section of a primary root 125
Figure 6.5 Comparison of Fixed-Width and Weighted Histogram.....126

Figure 6.6 Results of 4D assisted Segmentation of the root system of a tomato plant 127
Figure 6.7 Comparison between Root Width Constrained and RooTrak Segmented outputs..... 128

List Tables and Equations

Equation 3.1 Minkowski Distance 45
Table 3.2 Summary of root traits measured with WinRhizo and RooTh 62
Table 3.3 Ground truth of manually skeletonised root system. 64
Equation 4.1 Energy Minimisation Function of an Active Contour Model 73
Equation 4.2 Euler-Lagrange Equation 74
Equation 4.3 Matrix form for calculating successive iterations of the ACM model 75
Equation 4.4 Equation of Force. 75
Equation 4.5 Equation of Tortuosity. 82
Equation 4.6 Cubic curve parametric canonical form 83
Equation 4.7 Matrix form for calculating the cubic spline interpolation of the snake points 83
Table 4.8 Ground truth of manually skeletonised root system. 88
Equation 5.1 Similarity Transformation Function 95
Equation 5.2 Minimisation term for ranking the correspondence pair points generated by MST 101
Equation 5.3 Number of iterations to brute-force registration parameters 103
Equation 5.4 Number of iterations to run RANSAC 103
Table 5.5 Ratios of Brute-force vs RANSAC performance for given probabilities 103
Equation 5.6 Affine Linear Transformation. 108
Equation 5.7 Translation matrix used for the artificial transformation. 111
Equation 5.8 Adjusted transformation matrix used to artificially transform 112
Equation 5.9 Transformation parameters generated by RootShed's registration algorithm 112
Table 5.10 Total root length estimate of the effect of Runge's phenomenon 115
Table 5.11 Dynamic traits evaluation of root elongation. 118

Abstract

The root system of a plant is responsible for supplying it with essential nutrients. The plant's ability to explore the surrounding soil is largely determined by its root system architecture (RSA), which varies with both genetic and environmental conditions. X-ray micro computed tomography (µCT) is a powerful tool allowing the non-invasive study of the root system architecture of plants grown in natural soil environments, providing both 3D descriptions of root architecture and the ability to make multiple measurements over a period of time. Once volumetric µCT data is acquired, the root system must first be segmented from the surrounding soil environment and then described. Automated and semi-automated software tools can be used to extract roots from µCT images, but current methods for the recovery of RSA traits from the resulting volumetric descriptions are somewhat limited.

This thesis presents a novel tool (RooTh) which, given a segmented µCT image, skeletonises the root system and quantifies global and local root traits with minimal user interaction. The computationally inexpensive method used takes advantage of curve-fitting and active contours to find the optimal skeleton and thus evaluate root traits objectively. A small-scale experiment was conducted to validate and compare root traits extracted using the method presented here alongside other 2D imaging tools. The results show a good degree of correlation between the two methods.

Acknowledgments

I would like to use this opportunity to express my deep gratitude to my supervisors Tony Pridmore, Malcolm Bennett and Sacha Mooney for their support and guidance throughout the course of my PhD. I would also like to thank Stefan Mairhofer for providing the data used in the evaluation work of my PhD. I would like to offer my thanks to Craig Sturrock and Brian Atkinson for their assistance with operating the CT facilities and their help with acquiring the plant root images which were invaluable part of my research. I would like to extend my thanks to my internal examiner Andrew French and my external examiner Hannah Dee for their useful critiques of my thesis. I would also like to thank Marcus Griffiths for his feedback which helped me design and refine the user interface of the tool developed in this thesis.

Finally, I wish to thank my friends Charlotte Amy Richardson, Alex Peter De Garr Wilkinson, Edward Matthew Haslam, Heather Elizabeth Rimmer, Tom Bishop and Jad Zeidan for their friendship and encouragement throughout my life as a PhD student.

Chapter 1.
Introduction

Throughout the recent history of humanity, most challenges faced by societies have been linked with their constant demand for limited resources. Availability of food and water has played a major role in the success of early human settlements throughout modern times [1] and in certain cases, shortages have determined the outcome of historical events such as the Irish potato famine of 1845, which is an evident case of our need to successfully farm food crops [2].

In modern times, the cultivation of crops couldn't be more important as our population is continuously increasing in size. It is estimated that the human population will reach 8 billion by 2020 [3] which would inevitably put a lot of pressure on the supply chains of basic commodities and could result in driving millions more people into poverty and limiting their access to essential goods unless necessary actions are taken to avert such scenarios. On top of the agenda are issues like climate change, renewable energy sources and access to clean water. Nonetheless, one very important issue is food security and the ability of farmers and agronomist to grow sufficient quantities of crops at affordable costs [4].

Ever since humans first took on farming, the productivity of crop fields or the total amount of useful harvest per unit of land has continued to increase. At the very beginning, early farmers would use very basic and mostly manual forms of plant cultivation resulting in relatively low productivity when compared to modern day quotas. Yet, at the time farmers had to produce enough food for a relatively small population and overall they could carry out sustainable form of farming [5].

Over thousands of years, farming practices remained mostly unchanged in the form of subsistence agriculture involving *slash and burn* methods used to clear land for the purpose of creating crop fields. As artificial selection of plant species was taking place, the result was an overall increase in the useful harvest which inevitably allowed the steady growth of the human population. Many of the fruits and vegetables we consume nowadays have distant

wild cousins which have much smaller and often less tasty and nutritious edible parts. It wasn't until the 1960s with the advent of the *Green Revolution*, largely masterminded by Norman Borlaug [6], that humanity saw a significant increase in the amount of harvest per hectare of land. According to [7], wheat production in Britain grew by a marginal 1.5 tonnes per hectare over a thousand years and it wasn't until late 1900s when in just a couple of decades it suddenly increased to over 7 tonnes. In the same period there was a sudden boom in the population of the UK which would most likely not have been possible without the rapid increase in food production.

The Green Revolution, as it is commonly known, was mainly a result of the rapid industrialisation of farming technology which meant that a lot of the labour intensive work could now be carried out by machines. A further contributor to the increase of the yield was the use of chemical fertilisers and pesticides. Another major breakthrough was the use of high-yielding varieties of wheat and rice instead of traditional varieties [8]. The high-yielding varieties are specially selected. Enhanced varieties of wheat and rice for instance, which produce much larger harvests by growing shorter stems and thus concentrating most of their energy in growing seeds instead of developing the plant body [9]. Subsequent increases in yield from the early 1990s until the present day have mostly been a result of advances in genetic engineering [10].

All these advances in agronomy and farming technologies are essential for the sustainability of our modern day society. According to [11], as the population continues to increase, we are faced with the same problem as half a century ago and it is essential that the productivity of crops continues to improve if we are to avoid mass starvation and hunger across the globe.

In a similar way to the means by which the *Green Revolution* was achieved, deeper understanding of the plant growth and development cycles could allow plant scientists to develop new varieties of commonly grown crops with even higher productivity and refine the conditions under which crops are grown, irrigated and fertilised to ensure maximum uptake of nutrients and highest rate of germination. For example, the recently discovered variety of wheat made by the National Institute of Agriculture and Botanics could provide 30% increase of harvest on average [12][13].

One particular aspect of plants is their root system. The "[…] roots are crucial for the delivery of water and nutrients to plants [and they have] impact on both the physiological and [the] biological status of the surrounding soil […]" [14]. Over the years, the majority of plant science research has been focused on studying the "upper half" of plants. The root system or the "hidden part" of plants has largely remained out of the view of scientist and has only attracted greater interest over recent years [15].

The primary function of the root system is to exploit the surrounding soil, absorbing water and nutrients that are essential for the growth, development and survival of the plant. Root system architecture (RSA) is shaped by both the genetic makeup of the plant species and the environment that the plant inhabits, and much remains to be learnt about how plants control the development of the root system. Understanding and developing new root system architectural traits related to increased water and nitrate uptake efficiency for instance is of particular interest to the plant science community. It is common knowledge that water and nutrients are not homogenously distributed across the soil profile in a typical field environment and thus the architectural and special organisation of the root system is essential in securing the most efficient access to and uptake of essential minerals [16].

Rooting depth, for example, is a key architectural trait for nitrate acquisition. Nitrate is highly soluble and therefore quickly leaches deep into the soil profile, out of the root system's influence [17]. Root angle manipulation could also improve crop performance by increasing phosphate uptake, which is an insoluble macronutrient and therefore accumulates in the top layer of the soil [18]. Root length is particularly important for water uptake as it enhances the capacity of the root system to effectively explore deep soil layers and locate available water [19].

Once RSA traits for water and nutrient uptake are identified, chromosome regions or genes that are associated with these traits can be identified using Quantitative Trait Loci (QTL). Determining the key genetic traits responsible for the major morphological properties of the root system of crops could subsequently be used by breeders to select varieties with RSA for improved drought tolerance and more efficient nutrient uptake [20].

The roots of plants are particularly hard to study as they are buried underground and soil is an opaque medium. Thus, direct observation without interfering with the plant would be

almost impossible so various different methods have been developed to allow the observation of roots as they grow. Understanding the morphology and structure of roots as they develop is an essential step in determining the genotype and phenotype of different varieties of common crop species of plants and it allows scientists to link characteristic root traits to genetic variation in the DNA sequence of plant species. This could suggest various different ways in which the plant productivity could be improved.

One particular method for imaging plant roots non-invasively which is the focus of the research reported here is micro-CT (Computed Tomography) 3D Imaging. Specialised tools like µCT X-Ray scanners are required to allow non-invasive observation of root systems in their natural state. The plants are grown in specialised soil pots for a number of days under various growth conditions related to the particular biological experiment being carried out. Once they are ready to be scanned, the soil pot is put between an x-ray gun and an x-ray detector. The x-ray gun irradiates the soil pot and a shadow is cast on the x-ray detector. As different materials inside the soil like air, water, organic matter and solid particles absorb x-ray radiation to a different extent the structure of the soil and root can be revealed. By scanning the same pot from multiple directions, one can use tomographic reconstruction software to recover the 3D volumetric data representing the internal structure of the soil pot where the soil, the air and water pores and crucially the roots of the plant are all visible.

Once the 3D volumetric data encoding the internal make-up of the soil columns is recovered, the root system architecture could be analysed and various traits could be measured. Due to the relatively large size of the resulting image stack (>10GiB) it is very difficult for a human to visually examine them in detail alone and thus impractical for them to be analysed solely using manual methods. Instead, a more efficient approach is to develop software tools which could automatically evaluate the properties of the root system by examining the raw CT data. The software analysis consists of two major steps – segmenting the root system out of the soil column and doing various global, local and dynamic trait measurements on the resulting segmented output. The second step involves skeletonising the root system and using the skeletonised curves to automatically estimate various root traits including length, width angles, volume, surface area etc.

The aims of this thesis are: 1. to develop a novel semi-automated plant root skeletonisation method allowing the quantification of root traits and 2. to produce a software tool which can use the method to reconstruct a complete description of a root system recovered from volumetric CT images where every root is individually quantified in its entirety (as opposed to a single global trait evaluation such as total root length).

1.1 Contributions

The following is a brief summary of the main contributions developed in this work:

- Use of curve-fitting approach to generate locally optimised preliminary skeleton description of a root
- Use of Active Contours to generate globally optimal descriptions of the root system skeleton
- Development of a software tool (RooTh which stands for Root Thinning) written in C++ which implements the curve-fitting and ACM methods to process volumetric CT scans and output a file describing the recovered root system architecture and traits of interest.
- Development of methods for automatic registration of CT volumes used in 4D temporal trait evaluation (i.e. time-series of roots growth) using watershed algorithm for feature selection and Minimum Spanning Trees (MST) for feature-correspondence detection. Implement the above method in RootShed which is a tool written in C++.
- Preliminary work on temporal image registration using the output of the watershed segmentation along with the skeleton output of older CT scans of a plant to aid the segmentation of earlier ones.

1.2 List of Publications

S. Mairhofer, J. Johnson, C. J. Sturrock, M. J. Bennett, S. J. Mooney, and T. P. Pridmore, 'Visual tracking for the recovery of multiple interacting plant root systems from X-ray µCT images', *Machine Vision and Applications*, vol. 27, no. 5, pp. 721–734, Jul. 2016.

S. Mairhofer, T. Pridmore, J Johnson *et al.*, 'X-Ray Computed Tomography of Crop Plant Root Systems Grown in Soil', *Current Protocols in Plant Biology*, vol. 2, no. 4, pp. 270–286.

J. Johnson, T. Pridmore *et al.*, 'RooTh: Plant Root System Skeletonisation', *Frontiers*, in preparation

Chapter 2.
Literature Review

2.1 Origin of Genotyping and Phenotyping

The motivation to describe and differentiate the morphological characteristics of many plant species predates our civilisation and started thousands of years ago when early farmers began cultivating common food crops. Those very early human settlers would have discovered that different members of the same species of plants bear differences in the quality and quantity of edible output they produce and thus they would have intuitively begun to select and breed only the ones with desired properties. In essence, they would have carried out the large-scale selective breeding of many food crops we now take for granted by visual inspection of the quantitative differences between different breeds [21].

A more formal definition of a quantitative description of the morphological traits of a biological organism is known as phenotyping, and the field of science which studies the phenotype of different organism is phenomics. Every biological organism can be uniquely characterised by a set of morphological descriptions or traits varying from its overall shape and size, to the shape, size, number, spatial distribution etc. of its various parts and organs. The exact set of such measurements defines each organism as does a fingerprint. The genome of a biological organism can be defined as the set of all genes that make up the cell nucleus and those located in the cells mitochondria and chloroplasts (in the case of eukaryotic cell organisms) and consists of all coding and non-coding DNA sequences. Whilst the genome is unique for every organism, it is more similar between members of the same species than it is for members of different species and the degree of similarity or dissimilarity is closely related to the hierarchical membership of the evolutionary tree of life i.e. the fundamental grouping of species of organisms based on their evolutionary origin. The genome is responsible for defining both the proteome and the phenome of an organism which is a consequence of the central dogma of biology [21].

The central dogma of molecular biology states that the flow of information in a living cell on a molecular level follows a particular pattern where information is copied from DNA sequences to RNA sequences to protein and peptide sequences which collectively result in

gene and phenomic trait expressions. Therefore, the genetic sequence of nucleotides making up the DNA polymers of the cell chromosomes directly affects the order in which amino-acids are chained in the proteins and peptides and thus affect their structure. Since the genes of every organism are unique to that organism so are the proteins being synthesised by its ribosome organelles. As a consequence, the organism's identity can be uniquely defined by all the proteins it produces in a similar way to its genotype and the field of science interested in studying the protein makeup of biological organisms is proteomics. Since the proteins are essentially the building blocks of life and are vital component of every living cell, they entirely determine the functional characteristics of a specific organism.

The majority of enzymes involved in metabolic chains as well as many of the supporting structures and organelles are made up of proteins. As a result of basic biochemical interactions between different proteins within individual cell bodies, more complex functionalities emerge like the ability of a plant organism to acquire energy from sunlight through photosynthesis or the ability to absorb water and move it upwards from its roots to its leaves where it transpires. Naturally, it can be assumed that the efficiency or capacity of an organism to carry out a specific task which has biological significance will be determined by the complexity of the biomolecular mechanisms underpinning that function. Over the course of millions of years of evolution, as a consequence of natural selection, the genetic makeup of species would have diversified and they would have become better adapted to their surrounding environment by carrying out their basic functions in the most efficient way possible.

The genotype and associated genetic traits of a plant are therefore directly responsible for the phenotypic expressions by determining the structure of underlying proteins which carry out the biological functions resulting in those expressions. The opposite isn't strictly true – different phenotypic traits may appear in genetically identical plant organism of the same species but which are grown or developed in different environmental conditions. In other words, the environment has just as big a role to play in the phenotype of plants as its underlying genotype [21].

Determining the phenotype of a plant organism could often prove a lot more challenging that determining the genotype. Phenotyping involves taking exact measurements of plant

metrics such as specific leaf area [21]. Traditionally this could be done using manual measurement techniques although more recently with the advances of digital imaging it is becoming more and more common for plant scientists to use computer software to automatically quantify and analyse digital images of various plant organs and thus produce a list of traits describing the phenotype of the sample. Advantages of digital image analysis over manual data collection are higher-throughput and objective data evaluation.

2.2 Phenotyping of Root Traits

A number of different methods and techniques have been developed over the years which allow for phenotyping of root traits [22]. Due to the complex nature of the soil composition and the fact that it is an opaque medium, it is often very hard to obtain a complete picture of the root system. Thus, different methods usually offer only a partial view of the roots or various trade-offs when measuring root properties. The majority of the techniques could broadly be classified in two major categories: direct and indirect ways of extracting root traits/measurements.

Direct ways of observing roots (those that allow the roots to be seen using visible and infrared light) are usually invasive, as they require the partial or complete separation of the root system from the soil and thus shorten or interfere with the natural life cycle of the plant. Whilst such methods could offer very accurate measurement of certain root traits like root length, they often do not permit prolonged observation of root system development.

Indirect ways of observing a root system usually involve the use of additional tools which in one way or another allow for one to “see” through the soil. Such tools could be x-ray or MRI µCT scanners, the use of radioactive isotopes etc. Whilst such methods are non-invasive, they cannot be used to directly measure the properties of the roots, but only generate a digital imprint of the roots and soil. Thus, a secondary post-processing of the image data is necessary to allow for the complete extraction of roots traits of interest. Due to their non-invasive nature, methods like µCT imaging could also be used to observe the plant over time as it grows and thus provide valuable information for picking-up certain root traits.

The ideal observation method would be one that allows observation of the roots of plants in their natural environment (ideally in-field) over prolonged periods of time without

interfering with their natural lifecycle (i.e. does not destroy the plant and does not affect its growth). If a control sample was to be grown at the same time, the observed and non-observed plant samples would ideally have the same morphological structure at the end of their life. As such a method is unlikely to exist in practise it is necessary to make some trade-offs and their nature is determined by the context of the experiments being carried out and the type of root traits that need to be measured.

2.2.1 Invasive Techniques

The majority of invasive methods usually deploy techniques which interfere with the integrity of the plant and/or alter the environment in which the roots grow. Different methods will interfere with the plant to a different degree with some resulting in the immediate death of the plant after measurements are taken, whilst others like X-Ray µCT have negligible/not detectable effects [23].

2.2.1.1 Root Washing

Root washing could be considered to be the most basic method for quantifying root traits. It involves growing plants in soil or any other compound resembling soil for a period of time until the measurement takes place. Usually this involves careful washing off of the soil particles from the roots to enable their visualisation. Various ways for removing the soil include using water or shaking the plant (works well only for medium-dry to dry soil). Prior to the experiment the plants can be grown in field or in lab (in a controlled environment). If the plants are grown in a field environment they need to be dug up first. This could be done by either taking a soil core [24] or by pulling out the plant for its stem using force [25]. Both methods are susceptible to leaving parts of the root system in the field. When grown in lab conditions, it is generally significantly easier to separate the plant from the soil by carefully removing the pot and washing off the soil.

The root washing technique can be a very useful method for successfully picking up traits in roots. It is in general a very versatile method as it can be used in many different scenarios in both plants grown in lab and in field conditions. It is also considered to be a cost efficient technique. On the down side, since the soil environment is completely destroyed in the process the plant has to be disposed of upon taking the required measurements making the method unsuitable for continued observation of the root development. It is also only

suitable for measuring basic traits like root length, root width or the total number of roots/lateral roots and cannot be used to measure traits involving the way roots are "suspended" within the soil as in measuring root angles, root directions, convex hull etc. This is because most plants have evolved to grow in soil and thus their roots are very flexible as opposed to the top part of the plant (stem and leaves) which have evolved in air and can support themselves.

2.2.1.2 Rhizotrons

Rhizotrons are special sleeves laid down in the soil prior to sowing the seeds of the plants which are then used to monitor the root system as it grows. Rhizotrons are particularly suited for field studies and they introduce much less interference with the development of the plant, crucially allowing for it to be monitored throughout its life. Usually specialised camera equipment/endoscopes [22] are used to obtain digital images which can then be process with specialised software like Rootfly [26] or WinRhizo [27]:

The main disadvantage of the rhizotrons is that they only offer partial views of the root system (only roots close to the glass surface of the sleeves are exposed). Another limitation is that the measurements can only be done in two dimensions.

2.2.1.3 Hydroponics and Transparent Gels

Hydroponics and Transparent Gels are similar to rhizotrons, in that they are in-lab methods for observing root growth. They can be used to monitor plants as their root system develops over time. Whilst they do allow for a complete view of the root system, unlike rhizotrons, it could be argued that since the roots do not grow in their natural environment (soil) they may develop unique traits not present in nature.

2.2.2 Non-Invasive techniques

Non-invasive methods usually involve growing plants in normal soil conditions and they rely on special machines capable of visualising the inside of opaque objects to image the roots within the soil. Such machines include X-ray μCT scanners which deploy high-energy photons which are able to penetrate through soil and locate the positions and dimensions of roots [14].

X-rays are capable of penetrating through physical objects and more importantly they lose some of their energy as they do so. The denser the material is, the more energy they lose. If an object is put between an x-ray gun and an x-ray detector, the total loss of energy known as attenuation can be quantified. By taking multiple images of all possible directions in 360° and using appropriate 3D reconstruction algorithms, it is possible to recover a detailed 3D picture of the exact attenuation coefficients at every location in the object. Since different components of the soil and the roots will have different density, they will attenuate the x-ray beam to a different extent. Thus, the attenuation coefficients can be used to distinguish between the roots and the surrounding soil.

Similar techniques could be used with other forms of energy like MRI scanners [28] or by saturating the roots with a radioactive isotope and then taking a series of radiograms from all viewing angles similar to the way x-ray CT scanners work.

The data generated by a CT scanner consists of a three dimensional grid of attenuation coefficients also known as voxels where each voxel contains value proportional to the density of the matter. In order to be able to visualise various real-world objects which might be present in the scanned 3D scene, further processing of the data is necessary.

Since different real-world objects will be made of different materials, they will attenuate the x-ray beam to a different extent and generally the voxels encoding the attenuation coefficients of a particular object will have a unique value within a narrow range. Using this basic representation of the CT data i.e. a set of voxels, if the range of attenuation values corresponding to a particular object is known, it would be possible to classify each voxel as either being part of the object or being part of the background. An alternative representation known as image segmentation does the opposite and if given an object within the 3D scene it returns all voxel coordinates which are part of that object. Thus, in order for one to be able "see" the location of the roots within the µCT scanned pot of a plant, one must first segment the roots from the surrounding soil.

2.2.2.1 Manual Segmentation

There are many different ways to segment an object from a scene and over the past few decades a number of algorithms have evolved which allow automated image segmentation. The most basic form of image segmentation could be considered to be manual

segmentation, where a user with the aid of interactive Graphic User Interface (GUI) tools can manually "pick" those voxels which he/she believes belong to the object they want to segment. They would usually be presented with a view of the scene (in case of a 3D scene they will often be presented with every single CT slice sequentially) and they will need to recognise where the object is based on the attenuation values of the voxels. One GUI tool allowing for manual segmentation of 3D objects is VGStudio Max [29] and it could be used to segment roots from 3D scans. The tool assists the user by deploying region growing techniques so the user only needs to choose a single point within each CT slice where he/she believe there is a root and the algorithm will automatically match all adjacent voxels having similar attenuation coefficient and will classify them as being part of the root system (the object being segmented).

2.2.2.2 Review of semi-automated and automated segmentation methods

It is also possible to extract the root system using fully automated image segmentation techniques and algorithms. A simple method for extracting a homogenous object whose voxels' values all lie in a very narrow range and are clearly distinguishable from the surrounding environment is thresholding [30]. The method identifies all voxel values within a predefined range and if the values fall within that range, the voxels are classified as being part of that object. Unfortunately, in CT data, voxel values describing roots are not uniform across the root system and also change as the root path progresses downwards away from the stem [31]. In order to overcome this issue a more sophisticated approach is necessary.

The following is a short review of existing root segmentation and skeletonisation software tools:

- RooTrak is designed to automatically segment out the roots of plants in CT images by deploying image tracking techniques. It traces the path of the roots from the stem of the plant to all the root tips using level-sets to compare adjacent CT slices and match the sections of the roots with one another. Once the initial seed is selected (the top of the stem of the plant) RooTrak can trace all the roots downwards. As tracking algorithms are able to adapt to variations in local image conditions, the algorithm successfully recovers the majority of the root system [31].

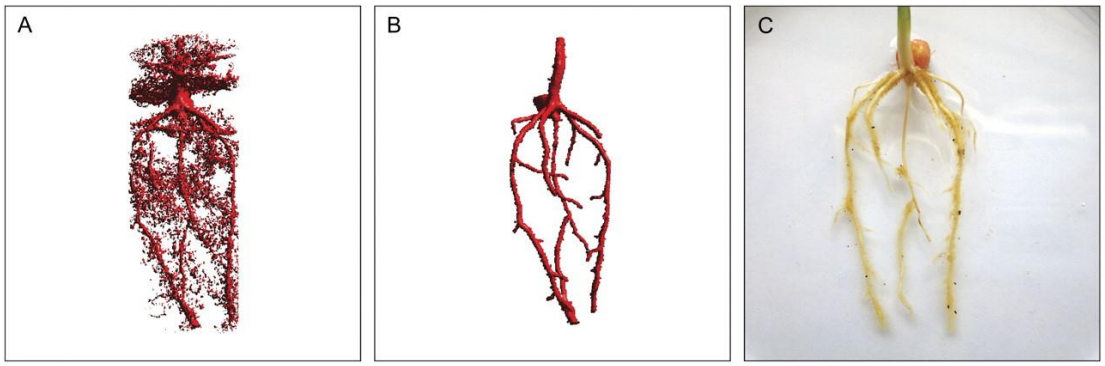


Figure 2.1 RooTrak Segmentation – A: simple threshold-based segmentation – very poor accuracy; B: RooTrak level-set based segmentation – root system has almost entirely being segmented successfully; C: picture of the root-washed plant used for the RooTrak segmented output (Credits: Photo[1] by Stefan Mairhofer[2] licenced under CCC[3])

- Root System Analyser is a segmentation and skeletonisation tool designed to extract root system traits from 2D neutron radiography scans of plant roots [32]. Neutron radiography, unlike CT imaging techniques makes it relatively easy to distinguish root material from the surrounding substrate in which the roots are grown. It represents the input image as a graph and deploys tracking techniques to recover the path of the roots through that graph, which is used for the subsequent segmentation and skeletonisation of the root system. After skeletonisation is completed, Root System Analyser is able to automatically evaluate global root system traits such as the topological characteristics.

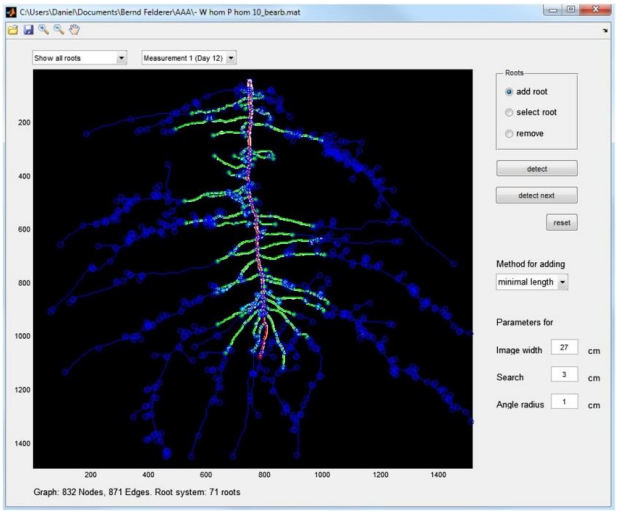


Figure 2.2 Root System Analyser (Credits: Photo[4] licenced under CCC[3])

[1] http://www.plantphysiol.org/content/plantphysiol/158/2/561/F4.large.jpg
[2] https://www.cpib.ac.uk/people/stefan-mairhofer/
[3] http://www.copyright.com/
[4] http://www.plantphysiol.org/content/plantphysiol/164/1/24/F8.large.jpg

- SmartRoot is a semi-automated tool designed to evaluate the growth and architectural characteristics found in 2D images of root systems. The root system is represented as a "parsimonious set of connected segments"[33] where all morphological information of the root is represented in a vector format which, according to the authors of the tool shares some characteristics with Geographic Information Systems (GIS) [33]. This structure is then used for root trait evaluation.

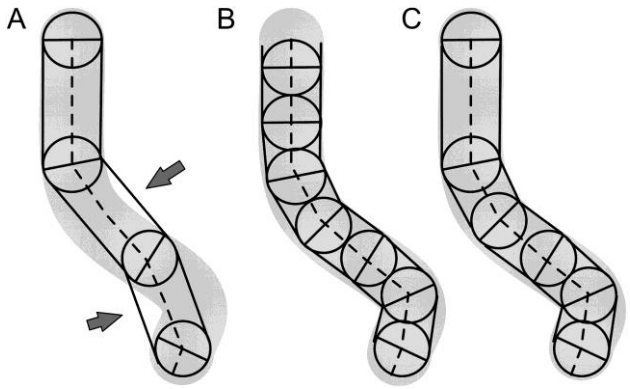


Figure 2.3 SmartRoot Skeletonisation evaluation (Credits: Photo[5] licensed under CCC[6])

- WinRhizo is a popular software tool "specifically designed for root measurement in different forms: […] morphology (length, area, volume), topology, architecture and colour analysis" [27].

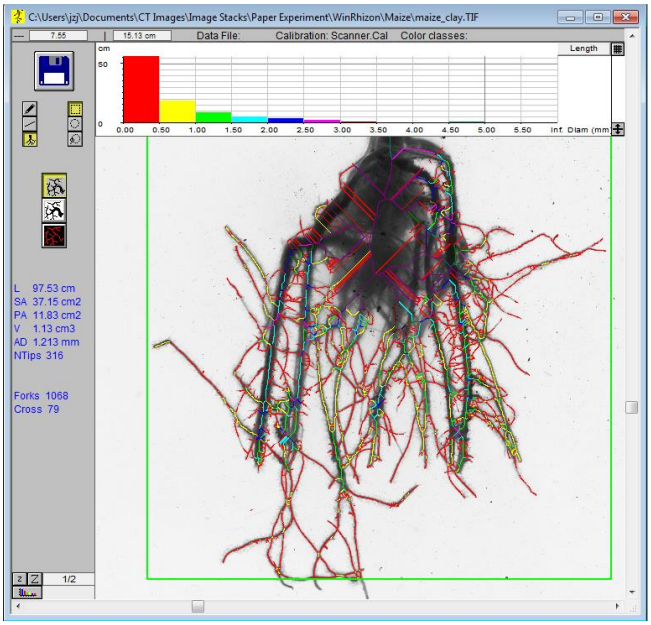


Figure 2.4 WinRhizo skeletonisation of a maize plant

[5] http://www.plantphysiol.org/content/plantphysiol/157/1/29/F2.large.jpg
[6] http://www.copyright.com/

It primarily works with 2D digital images of plants grown in rhizotrons, obtained using commercial high-quality digital scanners. Its main disadvantage is its inability to work with 3D root data representation: all 3D measurements made (e.g. root volume) are based on estimates.

- RootScape is another interesting root skeletonisation software tool which allows for phenotyping of Arabidopsis plant's root system architecture using a "20-point landmark model" [34]. It analyses the architecture of the root system and it then identifies a number of different key points, which are used to build a model of the root system:

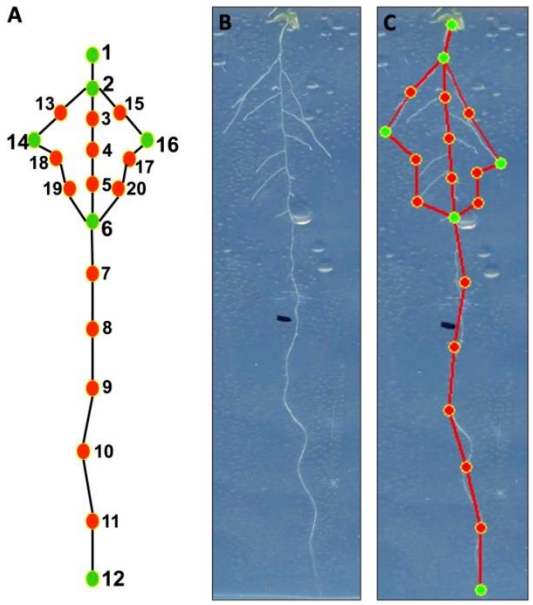


Figure 2.5 RootScape (Credits: Photo[7] licenced under CCC[8])

This model can then be used to classify plants, pick up traits and establish the purpose of different genes. Although the model does not encapsulate the complete skeleton of the root system architecture, it can still be sufficiently informative to detect genetic traits in Arabidopsis [34]. A main disadvantage of this tool could be considered that it assumes Arabidopsis-specific root architecture.

- RootNav is a semi-automated root segmentation and skeletonisation tool developed by Michael Pound at The University of Nottingham. It is designed to work with 2D images of plants captured with a conventional digital camera. It used the Expectation-

[7] http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3585581/figure/fig1/
[8] http://www.copyright.com/

Maximisation classification algorithm [35] to segment the roots from the background and applies Dijkstra's shortest path algorithm to skeletonise the roots and compute measurements like root length etc. [35]:

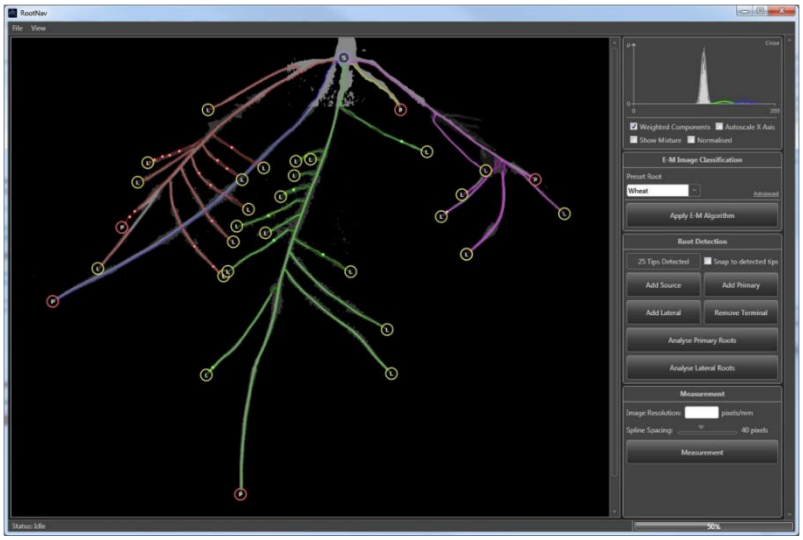


Figure 2.6 RootNav (Credits: Photo[9] licenced under CCC[10])

RootNav provides a novel approach for skeletonising roots and plant imagery in general, distancing itself from the classic Blum's Medial Axis Transform skeletonisation [36]. This novelty allows it to obtain more accurate root measurements avoiding the need of extensive pruning algorithms and heuristics to be put in place and ultimately enhance the detection of root traits.

The RootNav tool is similar to RooTh, the tool developed here, in that it used a contemporary approach to skeletonising objects. Both tools adopt a top-down approach and attempt to find an optimal path/curve connecting the root tips with the stem of the plant. A major lesson to be learnt from the RootNav tool is that it is in general very difficult to accurately determine the location of all root tips, which is essential for the operation of the skeletonisation algorithms deployed by both tools. The RootNav tool uses a workaround where it asks the user to correct any mistakes which mostly occur as a result of misclassified root tips. Due to the nature of the 3D data used by the RooTh tool, it is essential that a reliable root tip detection algorithm is found, as it would be

[9] http://www.plant-image-analysis.org/content/01-software/67-rootnav/rootnav-1.png
[10] http://www.copyright.com/

inherently difficult for the user to manipulate 3D data intuitively and interactively, if he/she was to solely rely on a mouse and keyboards inputs.

- RootReader3D is "a novel imaging and software platform […] developed for the high-throughput phenotyping of three-dimensional root traits during seedling development" [37]. It is designed to work with plants grown in transparent gel, allowing for the roots to be seen and imaged using a standard camera.

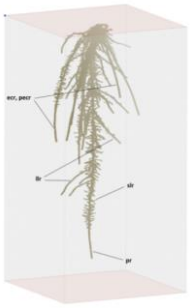


Figure 2.7 RootReader3D (Credits: Photo[11] licenced under CCC[12])

- PlantRoot [38] is a 2D root skeletonisation tool developed to work with automatically acquired images using a customised rhizobox. The software tool uses built-in MatLab subroutines to perform medial axis transformation and morphological thinning on the input dataset for the purpose of recovering the raw skeleton. The tool then determines the location of root tips using endpoints determined by evaluating the values of neighbouring pixels of points along the skeleton and searching for one surrounded by black pixels. The resulting skeleton descriptions are used for the evaluation of both spatial and temporal root traits.

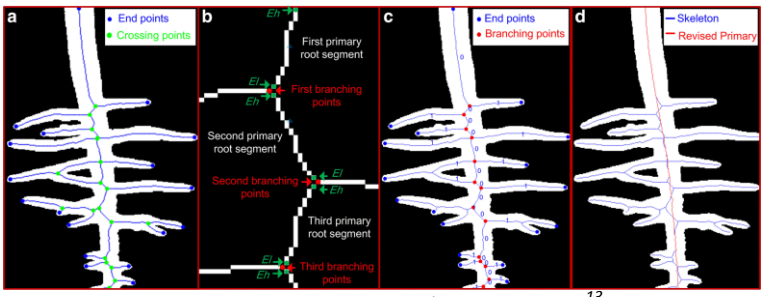


Figure 2.8 PlantRoot skeletonisation method (Credits: Photo[13] licenced under CC[14])

[11] http://www.plant-image-analysis.org/content/01-software/70-rootreader3d/rootreader3d-3.png
[12] http://www.copyright.com/
[13] https://media.springernature.com/full/springer-static/image/art%3A10.1186%2Fs13007-018-0316-5/MediaObjects/13007_2018_316_Fig3_HTML.png
[14] http://creativecommons.org/licenses/by/4.0/

- DynamicRoots [39] is a 4D root skeletonisation tool which allows for automated growth analysis of roots by reconstructing time-series data. It uses geometric graph representations of the root system which allows the reconstruction of individual branches by tracking their emergence through time.

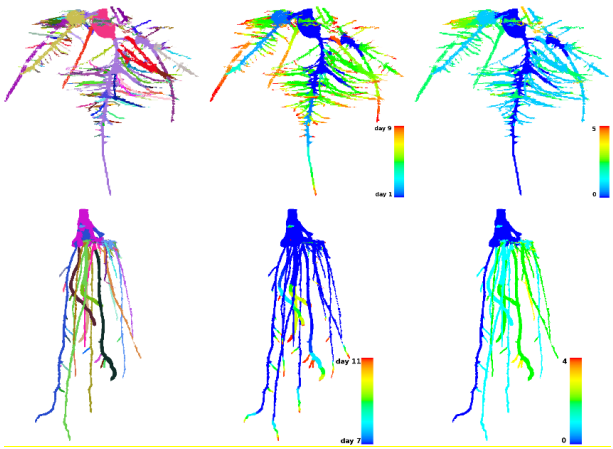


Figure 2.9 DynamicRoots Branch reconstruction (Credits: Photo[15] licence under CC[16])

2.3 Review of Image Skeletonisation and Thinning Methods

The original concept of skeletonisation was introduced by Harry Blum in 1967 along with an algorithm widely known as the grassfire transform [36]. This defines the skeleton of an object as the centres of a set of all Maximum Discs as illustrated in Figure 2.10. A disc is a circle around a point within the body of a binary object which stretches to the closest boundary. If a disc around a point touches the boundary of the object at more than one location, then that disc is by definition a maximum disc [40].

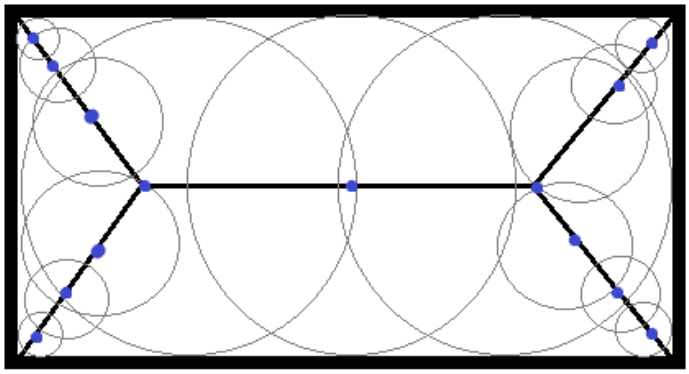


Figure 2.10 Maximum Discs and resulting skeleton.

[15] https://journals.plos.org/plosone/article/figure/image?size=large&id=10.1371/journal.pone.0127657.g002
[16] http://creativecommons.org/licenses/by/4.0/

2.3.1 *Morphological Thinning*

There are a number of different ways to efficiently compute the skeleton of an object following the above definition. The simplest way is to use a morphological erosion operator and iteratively remove layer by layer the body of the object until only a single thread of voxels is left. This method is also known as thinning. Alternatively, one could compute the skeleton of a binary image object by calculating the distance map of that object and then looking for ridges in the distance map. The ridges will correspond to the centre of all Maximum Discs contained in the object [41].

Whilst the thinning and skeletonisation methods described above could be very useful in many circumstances, the algorithms are prone to generate large quantities of false positive branches of the skeleton as a result of over-skeletonising the object. Thus, the derived skeleton could no longer be considered to accurately represent the morphological structure of the object. Therefore, additional post-processing is often necessary in order to remove the excess parts of the root skeleton. According to [42] "Pruning methods are an essential complement to thinning and skeletonisation algorithms because these procedures tend to leave parasitic components that need to be 'cleaned up' by post-processing". Often pruning algorithms and methods involve the use of heuristics to decide which parts of the root system are genuine and which are result of false-positive branches. If all branches are traced up along the sub-skeletons they are attached to, then a skeletonisation tree can be devised and appropriate heuristics can be applied to cut off unnecessary sections of the skeleton as illustrated in Figure 2.11 below.

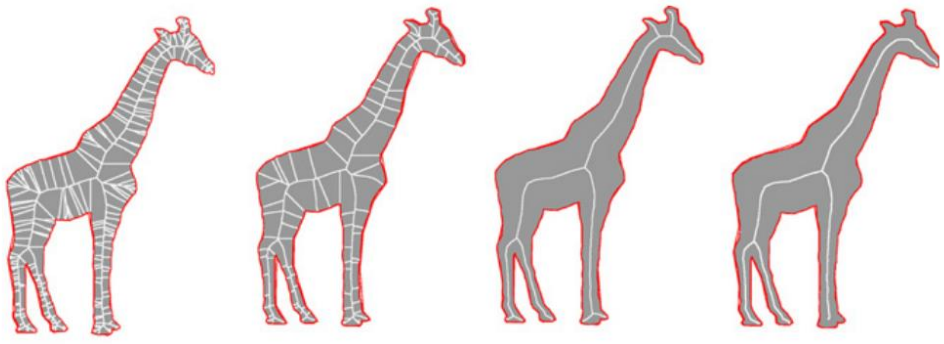


Figure 2.11 Example of skeletonisation tree and application of pruning algorithms. (Credits: Photo[17] licenced under CCC[18])

[17] http://www.sciencedirect.com/science/article/pii/S0097849312000684
[18] http://www.copyright.com/

Blum's definition of a skeleton as a Media Axis Transform is a very generic one, which is why the use of pruning post-processing is unavoidable. Devising a set of heuristics for each specific case is thus a complicated task and it cannot always be guaranteed that an optimal set of heuristics can be found. When the problem of skeletonising roots is considered, a more specific approach could be used to take advantage of the fact that the root system generally consists of long thin tubes connected with one another and forming a tree-like structure. The problem of skeletonising a root system could be reduced to the problem of finding an optimal path connecting every root tip with the stem of the plant. When the skeletonisation problem is defined in this way, there is a lot more flexibility to fine tune the algorithm to fit the root system structure and to a very great extent the use of heuristics can be avoided.

2.3.2 *Root System Skeletonisation*

The problem of finding an optimal path between the root tips and the stem of a plant i.e. the root system skeleton consists of two major steps – identifying the location of all root tips and finding the actual paths connecting them to the top of the root system. Since the density distribution of the roots overlaps with the density distribution of the surrounding soil [31], it is inherently difficult to segment the root system from its surrounding environment. As well as relying on manual segmentation such as that produced using commercially available tools like VG Studio Max, many authors have sought to develop an automated or semi-automated root segmentation tool (for a review of existing image segmentation tools see [43]).

The recovery of root systems from µCT image data requires each voxel to be classified as depicting either root material or the surrounding soil. RooTrak is a root segmentation tool that uses a modified Level-Set function based on the Jensen-Shannon divergence to model the intensity values and shape of root cross-sections, updating those models as roots are tracked down the image stack [31]. There are two major versions of the RooTrak software which were used during different stages of my PhD research – single-pass and multi-pass version. The single-pass version scans the 3D image volume slice by slice starting from the initial seed point selected by the user at the top of the image stack where the plant stem is located all the way to the bottom and extracting and segmenting the roots on the way. This

version has limited capacity at detecting upward and horizontal growing roots. The multi-scan version traverses the image stack up and down multiple times and at each iteration it re-seeds the level-set segmentation method with the root endings of the currently extracted system, thus allowing it to detect upward and horizontal growing roots as well. The Large Apparatus Root Segmentation or LARS tool is a follow-up tool derived from version 2 of the RooTrak software which is designed to segment root system architectures out of very large volumetric data with low spatial resolution where often the roots are too thin and thus disconnected between consecutive image slices and special image tracking techniques need to deployed in order to segment them correctly.

The resulting segmentation is represented as a 3D binary data structure, often stored as a stack of 2D images, in which root material is marked white and everything else marked black. This representation is immediately suitable for measuring global traits like total root system volume, total surface area, convex hull etc. In order to evaluate local traits, such as root length, root angles, and branching factor, however, it is necessary to identify the skeleton of the root system.

2.4 Root System Markup Language

The output of the root skeletonisation algorithm generated by RooTh is a detailed description of the root system which can be used to evaluate various different traits such as root length, root angles, root width and root volume, however, the skeleton itself could also be used as input to other plant imaging tools which offer further post-processing and analysis of the root system, notably tools such as SimRoot [44] which allows for the complete simulation of the growth cycle of a plant.

The structure of the root skeleton as generated by RooTh is represented internally inside the computer volatile memory (RAM) using basic language constructs offered by the C++ STL (Standard template library) such as vectors, lists, arrays and pointers. In order to export and allow the seamless exchange of the structure of the root system structure to any another relevant program or application, it becomes necessary to create a unified file format where the root system architecture can be represented in universal, machine independent format. Such a file format is RSML (Root System Markup Language) [45]. RSML is a “[…] unified language that enables root system architecture information storage based on [Multi-Scale

Tree Graph] formalism and [the] XML standards" [45]. RSML separates the root system structure in three main categories as depicted in Figure 2.12: root topology, root geometry and root functions:

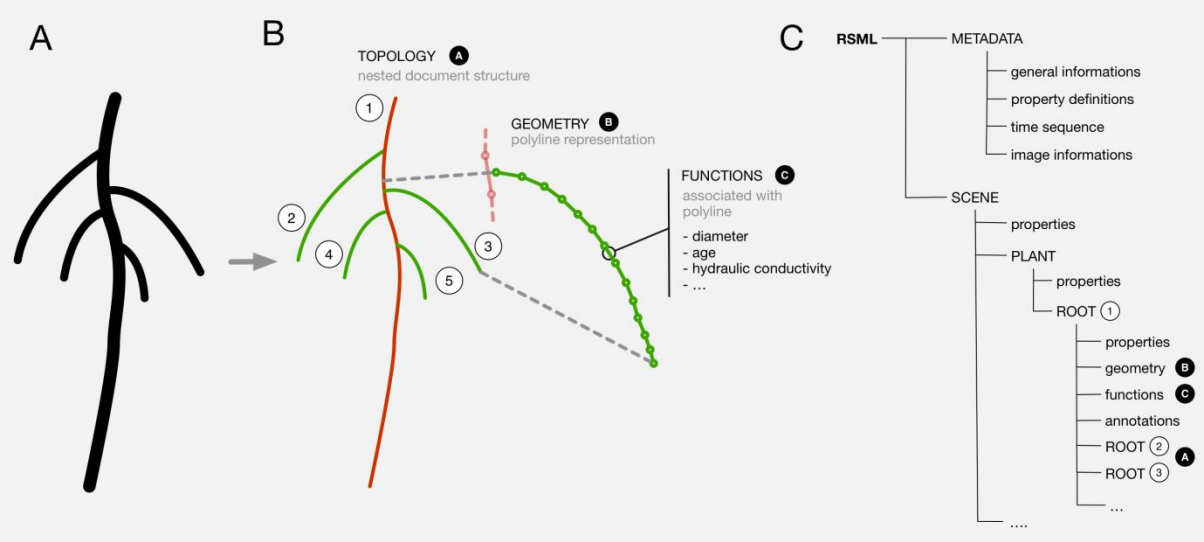


Figure 2.12 Root System Markup Language file format. The Root System Architecture (RSA) is encoded using three main categories – topology, geometry and function. An example of topology is the nesting of individual roots and classification of primary, seminal and lateral roots. Root geometry is concerned with the spatial positioning of the root within the boundaries of the scene and the root function are properties which are intrinsic to the rot such as root length, with, volume etc. (Credits: Picture[19] by Guillaume Lobet[20] licensed under CCC[21])

The topology of a root system represents the internal nesting of individual roots and the way they are connected with one another forming a complicated tree-like structure. Roots are classified into primary, seminal or lateral root based on their point of origin i.e. roots which originate from the seed of the plants are labelled seminal and those that originate from other seminal roots later on in the development phase of the root system are labelled lateral.

The geometry of the root system represents the spatial orientation of individual roots within the boundaries of the scene. The RSML format allows support for both 2D and 3D scenes and it represents the shape and form of the root using polylines [45]. RSML defines a polyline as a sequence of coordinate points and in the case of a 3D scene such as that processed by RooTh, the polyline is a sequence of 3D coordinate points. For the method described in Chapter 3 the output is natively a sequence of points and it can easily be converted to the

[19] https://github.com/RootSystemML/RootSystemML.github.io/blob/master/images/format.png
[20] https://github.com/guillaumelobet
[21] http://www.copyright.com/

RSML encoding. The output of the method presented in Chapter 4, however, is a sequence of piecewise parametric splines and as such, a converter function needs to be supplied within RooTh which will convert the parameters of each spline into a sequence of 3D points based on a predetermined resolution (number of 3D points between any two snake points).

Every root in the root system is defined by many different intrinsic properties such as its overall length, the width at every point along the polyline etc. All these properties, many of which are automatically evaluated by RooTh as part of the root trait extraction process, can be encoded directly within the RSML file. Certain properties such as root diameter are relevant to every single point in the polyline describing a given root and they vary along the length of the root. Such properties are defined as functions in the RSML format and they can be added individually to every polyline XML tag. Other properties such as root length or root volume however are relevant to the entire root and they are added as a separate XML tag at the beginning of the opening root tag prior to defining the polyline.

All of the above description of a root system is defined within a <SCENE> XML tag in the RSML file format. The format also allows the inclusion of metadata such as the time or place where an image scan was taken, the specific species of plant imaged or the software version of the tool used to skeletonised the image.

2.5. Summary

Phenotyping of root traits forms an important part in understanding the biological function of common food crops with commercial value as well as proving a link in understanding the relationship between gene expression and gene sequence. It is an important tool used by plant science researchers in their work of improving the useful harvest thus ensuring future food security. Many existing phenotyping methods fall short of meeting the needs of experimental research when applied to root system development. This is to do with a number of factors including the complexity of the soil environment and the fragility of roots. Novel imaging techniques involving the use of µCT X-Ray scanners is used to compensate for some of shortcomings of more traditional manual methods of root phenomics. Unlike classical methods such as root washing, CT based phenotyping requires heavy amount of computational post-processing before any meaningful traits could be evaluated. The data generated by the CT detector first needs to be reconstructed in a 3D volumetric stack and

subsequently, the distribution of the root material within the soil column needs to be determined through process known as root segmentation discussed in the literature review above. The resulting raw binary data needs further analysis in the form of root skeletonisation before useful traits such as root length or root volume could be evaluated. The work presented in this thesis addresses those needs by developing a tool for semi-automated skeletonisation of segmented µCT images of roots.

Chapter 3.
Root Skeletonisation via Non-Parametric Curve Fitting

3.1 The root skeletonisation problem

The recovery of quantitative data on plant traits – plant phenotyping – is now widely accepted to be a major bottleneck restricting improvements in crop production, and so efforts towards global food security. Following a focus on shoot traits in the first Green Revolution, roots have received increased interest of late, and X-ray computed tomography (CT) has shown significant promise due to its ability to non-invasively image roots growing in soil. Methods have been reported [31], [46], [47] for the segmentation of CT images to identify root material, but further techniques and tools are needed to recover structural descriptions of root architectures, and associated traits, from segmented images.

The raw data acquired from µCT is a three-dimensional grid containing approximated attenuation coefficients, each related to the density of the corresponding matter and the penetration of the X-ray beam. Following segmentation, this becomes a binary image, often stored as a stack of 2D images, in which root material is marked white and everything else marked black. Root systems are therefore represented as sets of voxels in a 3D array, with individual root branches appearing as elongated clusters of these voxels, some very narrow, some several voxels wide. This representation is immediately suitable for measuring global traits like total root system volume, total surface area, convex hull etc. In order to evaluate local traits, such as root length, root angles, and branching factor, however, it is necessary to identify the skeleton of the root system.

A skeleton of a root system can be defined as a single-threaded voxel representation of individual roots: a collection of one element wide strings of voxel locations, connected to reflect the structure of the root system. The skeleton should accurately encapsulate all major morphological features of the root whilst avoiding any non-essential shape variations;

those that do not directly alter any root traits of interest. In order to meet these criteria the skeleton must be:

- *Homotopic* – preserve all major morphological properties of the roots
- *Connected* – no root segments can be disconnected from the rest of the root system
- *Equidistant/Centred* – the distance between the skeleton of a root and the root wall should be maximised
- *Thin/single threaded* – "[the skeleton must be] one-dimensional, that is at most one voxel thick in all directions, except at joints where the curve-skeleton might become thicker to ensure connectivity between the different branches" [48]
- *Smooth* – "[this] is not only an aesthetic property, but is actually useful in some applications" [48] – if, for example, the thread of pixels representing the skeleton is uneven, the length of the root could easily be overestimated

A root skeleton displaying the properties described above can be used to derive a root system architectural description that in turn can be analysed to provide trait estimates. Various different formats can be used to encode a digital representation of plant root architecture with one particularly popular format commonly used by the plant phenotyping community being the Root System Markup Language, RSML [45].

The rest of this chapter is presented in 4 subsections. Section 3.2 introduces the root segmentation methods and the data used in this research. Section 3.3 presents the novel methods developed to skeletonise segmented CT data and produce RSML descriptions and section 3.4 goes on to evaluate results obtained from real-world CT images of roots. Section 3.5 discusses the strengths and weaknesses of the proposed approach and introduces the problems which are addressed in the following chapter.

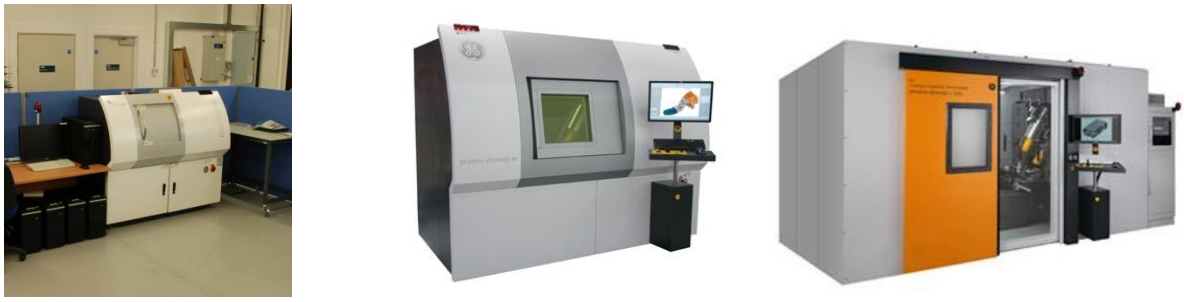
3.1.1 X-ray CT imaging of soil pots containing plant root systems

3.1.1.1 Image acquisition

All experimental data presented and analysed in this thesis was acquired at the Hounsfield Facility for Rhizosphere Research, located at The University of Nottingham's Sutton Bonington Campus. The Hounsfield Facility is a purpose-designed building comprising an automated glasshouse connected to an industrial unit containing x-ray CT scanners of various specifications and purposes. This unique setting allows a variety of plant phenotyping experiments to be devised and conducted and for large numbers of soil pots to be scanned without human intervention, raising the possibility of high-throughput 3D phenotyping of plant roots grown in soil.

The Hounsfield Facility is equipped with 3 x-ray µCT scanners, ranging from a small standalone scanner to a custom-built room-sized one. The smallest scanner, the GE Nanotom S (Figure 3.1.a) allows CT images with spatial resolutions of between 60 µm and 1 µm to be taken of samples no larger than 150x70x70 mm in size. This is equivalent to a small flower-size soil pot. This scanner is well suited for providing high-resolution images of very young plants (1 to 5 days post-germination). The majority of the images segmented with the single-pass version of RooTrak [31] were provided by this scanner. Very young root systems are composed predominantly of a primary and only a few seminal roots which are all gravitropic by nature. Such roots exhibit mostly downwards spatial orientation, making them suitable for segmentation using a single-pass tracking method (and for skeletonisation using a single-pass curve-fitting approach).

The Hounsfield Facility also features a larger, medium-sized scanner, the Phoenix v|tome|x M, also provided by GE (Figure 3.1.b). The v|tome|x M is capable of imaging samples of up to 400x290x290mm with a spatial resolution between 5 and 150 µm. This allows somewhat older plants of up to two weeks past germination to be imaged. Plants can of course grow for a significantly longer period than two weeks, but within 10 to 14 days (depending on any soil treatments stimulating or suppressing root growth) the root system would usually grow to occupy the full volume of the soil column. Imaging beyond this point is of little biological significance as the root system will take the shape of the soil column, rendering phenotyping meaningless.



a. *b.* *c.*

Figure 3.1 X-ray CT scanners in the Hounsfield facility, University of Nottingham
a. GE Nanotom S; b. Phoenix v|tome|x M; c. custom built Phoenix v|tome|x L

The large scanner is a custom-built, room-sized CT machine capable of hosting meter long soil columns of up to 200mm diameter, allowing fully grown wheat plants to be scanned months after their germination. These columns weigh some 80-100kg, and are difficult for a single person to manipulate unaided. To reduce the risk of damage to the scanner, and facilitate high-throughput imaging, the Hounsfield glasshouse is fully automated.

The Facility can house 160 1m soil columns, held in fixed metal racks within the glasshouse and moved to the scanner by an OCME Automated Guided Vehicle (AGV, Figure 3.2.a). The AGV guidance system incorporates an accurate map of the column racks, each of which has a unique identifier. The user specifies which columns are to be scanned, and when, via a custom built software interface. When a given column is due to be imaged, the AGV automatically navigates to its location using the map and precise estimates of its location provided by a laser positioning system. A scanning laser is fitted to the top of the AGV, and detects reflective targets mounted on the glasshouse walls, allowing the AGVs position to be recovered by triangulation. A purpose-designed gripper on the front of the AGV holds the column as it is moved to the large scanner door, where it is deposited into a fixed holder. A FANUC robot arm (Figure 3.2.b) then places the sample in the scanner and imaging begins. When scanning is complete, 3D images are deposited on a local, dedicated disc array.



a. *b.*

Figure 3.2 Automation in the Hounsfield Facility.
a. An OCME AGV carries columns within the glasshouse
b. a FANUC arm loads and unloads the v|tome|x L scanner.

3.1.1.2 Segmenting roots from soil

The segmentation tool supplied with the GE scanners and used prior to the development of RooTrak is VG Studio Max. This provides an interactive region-growing segmentation tool, requiring input from the user throughout the image segmentation process. The user must sequentially input seed points identifying the presence of root material and set a tolerance window of attenuation voxel values. The region-growing algorithm then automatically classifies all voxels lying within a small neighbourhood of these seeds as either root or non-root material based on the tolerance window threshold. The algorithm does not attempt to produce a fully connected set of voxels, only to label individual data values. As a result, the output comprises large numbers of blobs and clusters of voxels spaced around the roots, forming a disconnected and highly fragmented representation of the root system as can be seen in Figure 3.3. The clusters of voxels and blob artefacts which appear on the root surface are a source of false-positive root tips, described in Section 3.3.3. They are usually very short in length and can be pruned using length-based thresholding, however, and a set of dedicated heuristics are proposed to automatically detect and eliminate the corresponding false-positive root tips (Section 3.3.4).

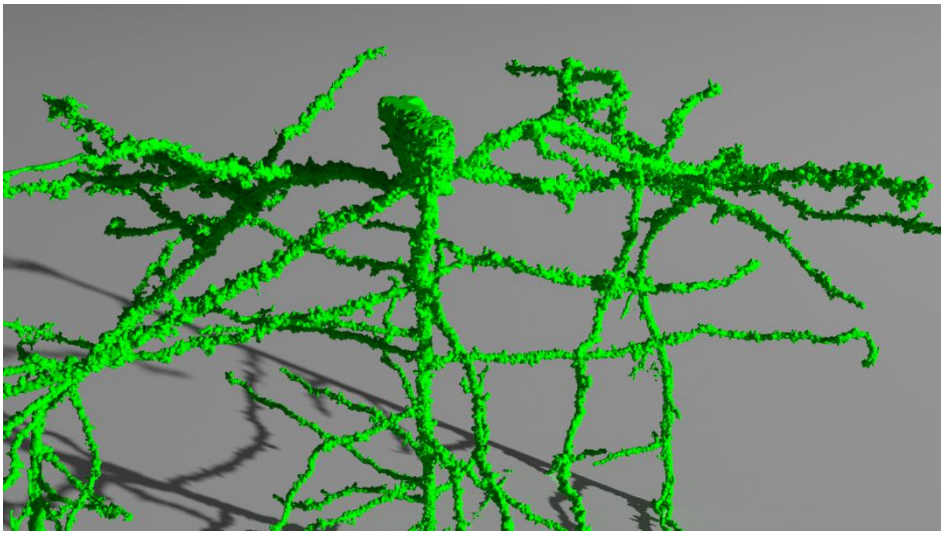


Figure 3.3 Manual Segmentation of a root system using VG Studio Max. Illustrated is the large number of incorrectly segmented cluster of voxels near the surface of the root (note: due to the marching cubes rendering technique many of the blobs appear fused to the root system)

The output of the manual segmentation can be improved using morphological post-processing involving the sequential application of the morphological opening operator (dilation followed by erosion) [42]. This goes some way towards connecting the disconnected root system fragments and fusing some of the blobs clustered around the roots with the root system, producing more complete root segmentation. Depending on the quality of the root segmentation, different levels of morphological opening might be required. If morphological processing is too aggressive it may result in disfigured roots and, especially in older plants where roots come into close proximity with each other, may fuse neighbouring roots into a single segment as illustrated in Figure 3.4.

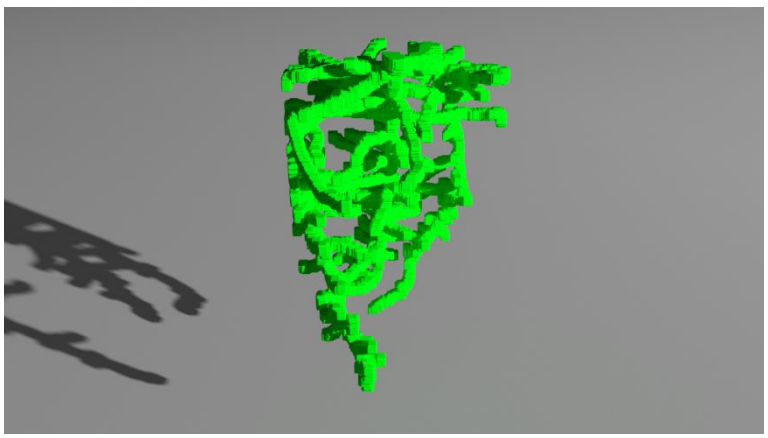


Figure 3.4 Illustrates that excessive application of morphological dilation may result in misshaped and fused roots of the segmented root system output

The data provided by RooTrak, on the other hand, is fully connected. This is guaranteed by the level-set tracking algorithm, as only fully connected sections of the root system that are visible in the image stacks can be traced. The surface of the root wall (Figure 3.5) is usually much smoother than that obtained via manual segmentation; this is largely due to the level-set speed function, which penalises high curvature in the root-soil boundary.

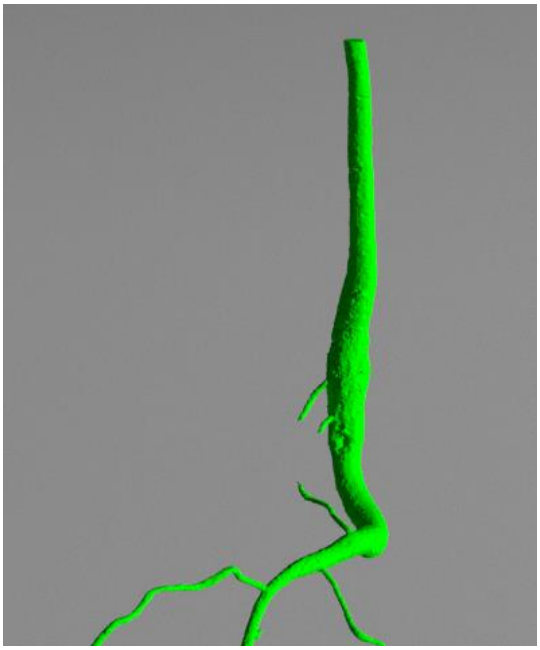


Figure 3.5 Smooth segmented root surface resulting from RooTrak Segmentation

This benefits the skeletonisation process by making it easier to define an equidistant and smooth skeleton path. However in some cases the level-set function has been found to bleed-out into pore spaces adjacent to the root, resulting in small parasitic bulbs as illustrated in Figures 3.6 and 3.7. This poses some challenges when recovering the structure of the underlying root system from the segmented output. A set of heuristics is, however, proposed in Section 3.3.3 which deals with those challenges.

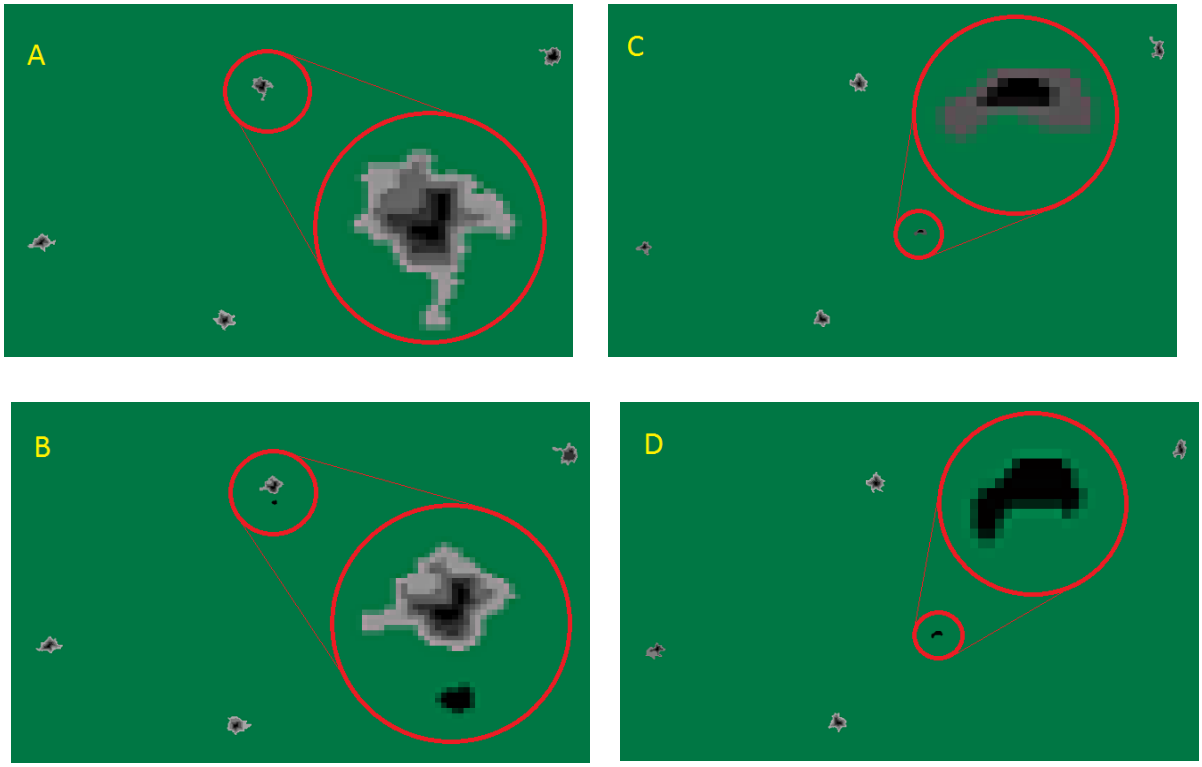


Figure 3.6 Illustrates the emergence of false-positive vs true-positive root tips
a. encircled – root cross section of single root; b. false-positive root tip
c. penultimate cross-section of a genuine root; d. true-positive root tip

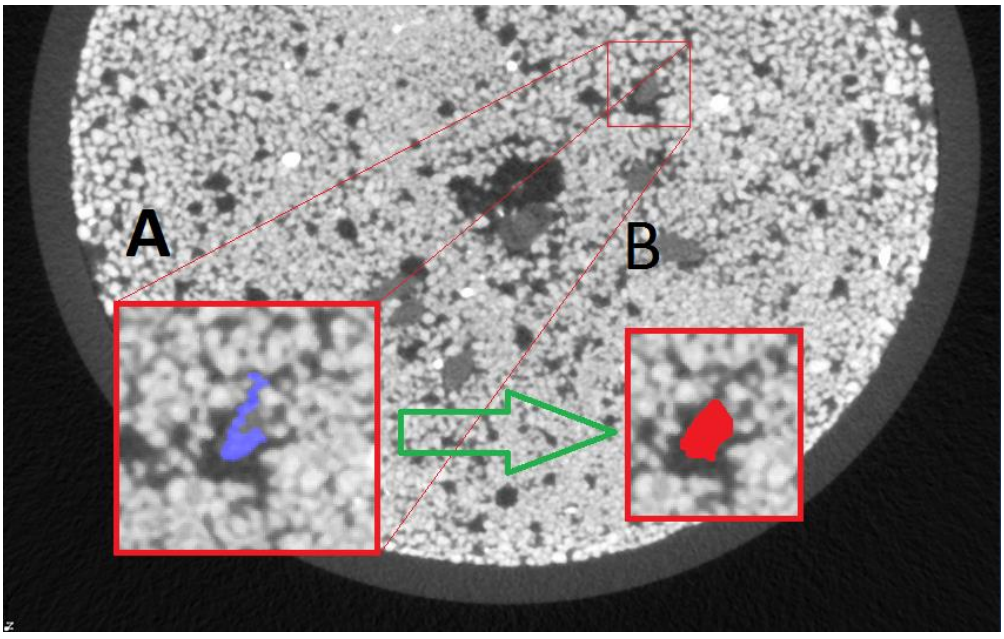


Figure 3.7 Root bleeding out/incorrect segmentation by RooTrak level-set speed function
a. Segmented cross-section as outputted by RooTrak
b. High-precision manual segmentation of what appears to be the genuine shape of the root-cross section done in 2D for visualisation purpose

Further to this, in some cases, the level-set function used in RooTrak may generate very small regions for individual cross-sections which appear as disconnected roots to the root tip detection algorithm in a similar way to fully disconnected sections in manually segmented roots. This results in the *orphaned* false positive roots jumping across to the nearest neighbouring root when skeletonised (the curve-fitting algorithm used to recover the skeleton of root tips and its drawbacks are described in detail in the following Section) forming fairly long branches and thus making it difficult to prune them in a subsequent post-processing step. This may happen if large number of "seed points" are manually entered by the user whilst recovering a particularly difficult section of a root resulting in large concentration of false positive root tips along the root as illustrated in Figure 3.8:

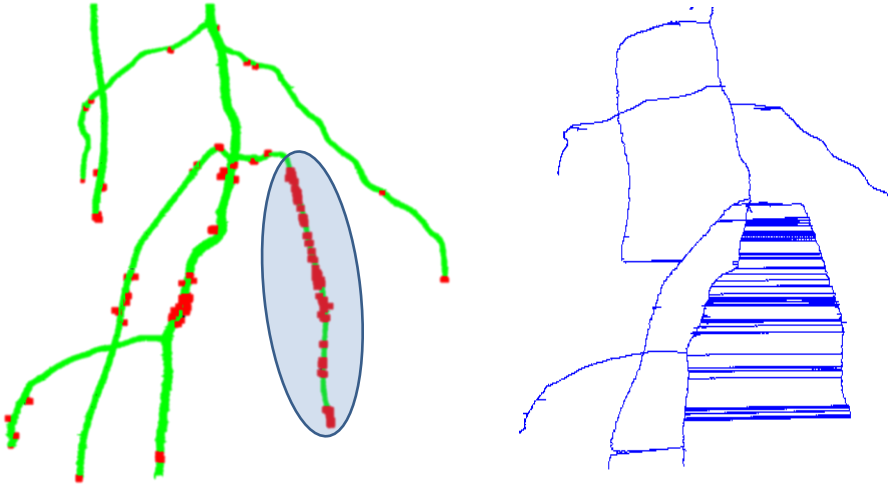


Figure 3.8 An example of large concentration of false-positive root tips (all tips marked with red in the left image) generated by RooTrak output along a section of a root (circled out) where the user has had to re-initialise the level-set method multiple times by adding seed points every few CT images throughout the tracking progress. The right image illustrates the large number of false positive root tips where the fitted curve of the skeleton jumps over to the nearest neighbouring root.

Similar to the way disconnected root sections are dealt with using morphological operators in the case of manually segmented data, morphological opening can be deployed to as a post-processing step applied directly to the RooTrak output in order to mitigate the effect of the tiny cross-sections and reduce or eliminate the number of false-positive root tips emerging as a result of these artefacts. The sensitivity of the false positive root tip detection algorithm with respect to the applied morphological operators is further explored in Section 3.3.4.

3.1.2 A non-parametric curve-fitting approach to root skeletonisation

3.1.2.1 Overview

Figure 3.9 outlines the proposed root skeletonisation method. The first step in the process is to locate root tips in an input, segmented CT image. A simple detection algorithm is applied, followed by a set of heuristics to remove unwanted false-positives. Once the root tips are known, the centres of root cross-sections are located in each horizontal slice through the 3D data volume and used as control points when fitting curves between each root tip and the origin of the root system (i.e. the stem of the plant). The centre points are identified using a 3D distance map computed over the set of root voxels. Piecewise linear smoothing curves are then used to fit a path through each root branch. The optimal smoothing parameter across different combinations of control points is identified by cross-validation. Then, a fractal representation of the root system of nested roots in which all lateral roots originate from the primary and/or seminal roots is generated using depth-first traversal of the tree structure of connecting cross-sections, starting at the base of the plant and recursively iterating until all root tips are reached. A manual annotation tool allows the user to correct any incorrect branches, mostly resulting from cases of crossing or touching roots. The output of the root architecture is exported in an RSML formatted file [45] and multiple set of traits are computed.

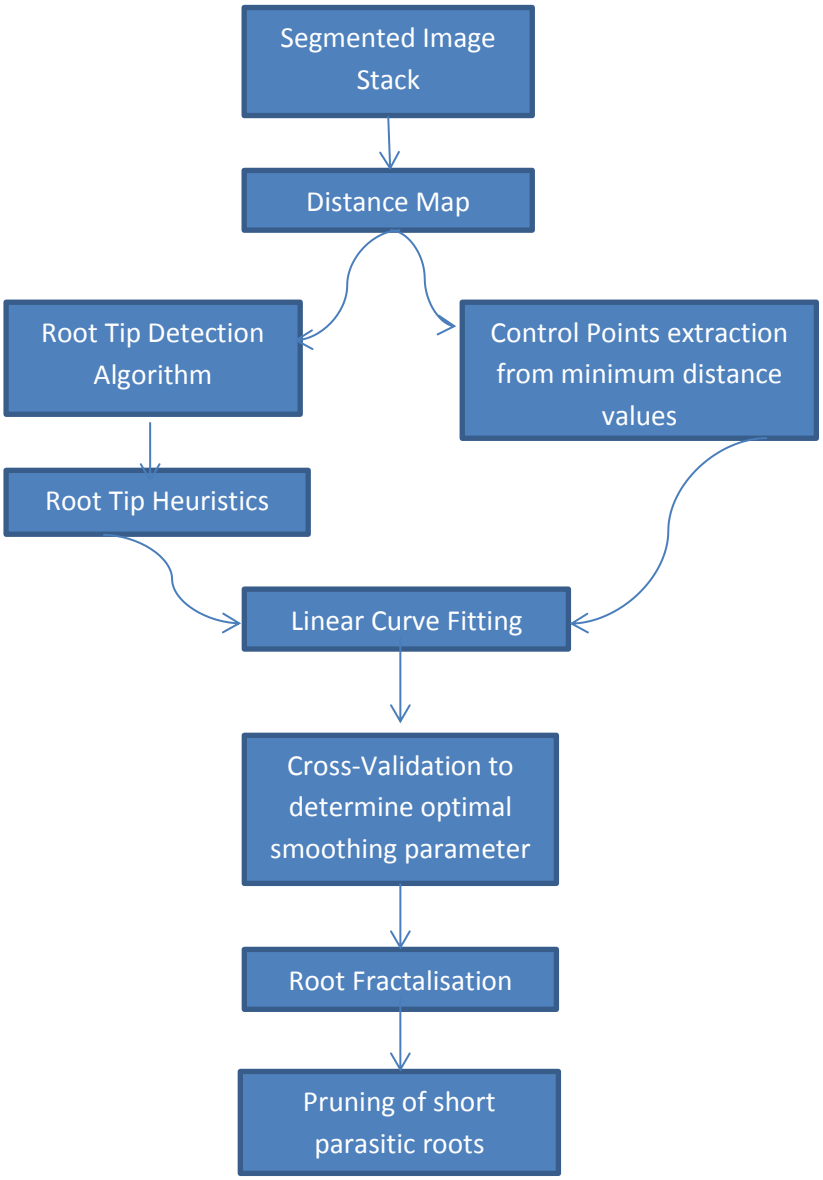


Figure 3.9 Overview of the proposed curve-fitting root skeletonisation method

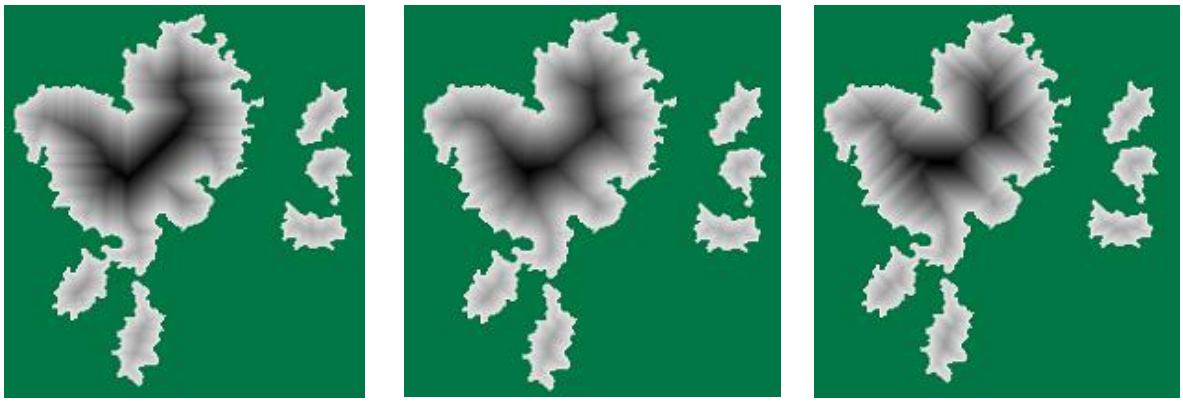
3.1.2.2 Distance Map Computation

The distance map of a binary image object records distance values for every pixel/voxel in the image which is part of the body of the object. The distance value of a voxel is defined as a numeric value which quantifies the distance to the nearest point on the boundary of the object to that pixel/voxel. There are different ways to calculate the distance between two points and they can be generalised by the Minkowski distance:

$$\left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{\frac{1}{p}}$$

Equation 3.1 Minkowski Distance: x and y are the coordinates of the set of control points

When the value of the exponent *p* is 1 then the measured distance between two points will simply be the sum of the difference between their x and y coordinates; the Manhattan distance. If *p* is 2 the formula will calculate Euclidean distance. When *p* approaches infinity the above formula will sum the maximum of the values of *x* and *y*. The results of applying the distance map to sample of 2D cross-section of a root are illustrated in Figure 3.10 below.



$\sqrt[1]{a^1 + b^1}$ $\sqrt[2]{a^2 + b}$ $\lim_{n\to\infty} \sqrt[n]{a^n + b^n}$

Figure 3.10 Calculating Distance maps of a root cross-section using (left to right) Manhattan, Euclidean and Chessboard distance

The distance resulting from taking the maximum value of either *x* or *y* is also known as chessboard distance as it represents the number of times a King has to be moved to travel from one point to another. When used to compute the distance map of a binary image object, chessboard distance can be implemented very efficiently in *O(n)* time by deploying a technique related to image thinning – gradually eroding the object layer by layer and at each iteration assigning the distance values to all voxels belonging to the layer being eroded. The deeper inside the core of the object, the greater the values assigned to the layers to reflect the greater distance of that layer from the original boundaries of the object.

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

Original Binary Image Object
(0 – background, 1 – foreground)

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	2	2	2	2	1	0
0	1	2	3	3	2	1	0
0	1	2	2	2	2	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

Distance Map
(numeric values representing distance from edge in pixel units)

Figure 3.11 Illustration of the calculation of chessboard distance map

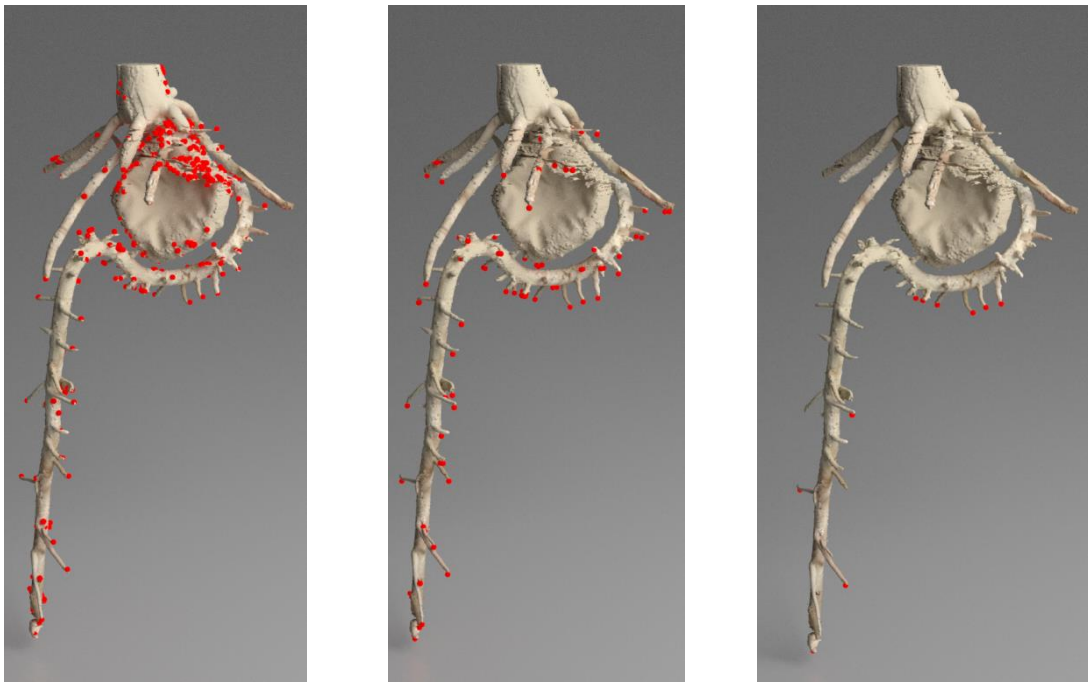
The above algorithm is very efficient to implement over 3D datasets such as the segmented 3D volume of a root system, resulting in only slightly reduced accuracy of the distance map edges. This is why it is the preferred method of choice for deriving the distance values in root skeletonisation tool developed in this research - RooTh (see Figure 3.11 for visual comparison of the different distance metrics).

3.1.2.3 Root Tip Detection Algorithm

A simple way to identify root tips is to compute a 3D distance map of the root system. Since the distance map assigns a distance value to every voxel that is considered part of the root system, the walls of the roots will be assigned values of 1. Thus, if a horizontal cross-section of a root is examined, voxels marked 1 will denote the edge of the cross-section and those greater than 1 will lie within the body of the cross-section. In cross-sections of root tips all voxels will be marked with 1, as they all form part of the wall of the root. Root tip detection therefore reduces to locating cross sections comprising all 1s.

This simple algorithm is guaranteed to detect all root tips. To prove its validity, imagine there is a root tip whose cross-section distance map contains values other than 1. From the definition of the distance map this implies that those values must be adjacent to other voxels that either form part of the wall or themselves form part of the body of the root. Therefore, there must be another cross-section region immediately below or above the cross-section region under consideration, implying that it cannot be a root tip.

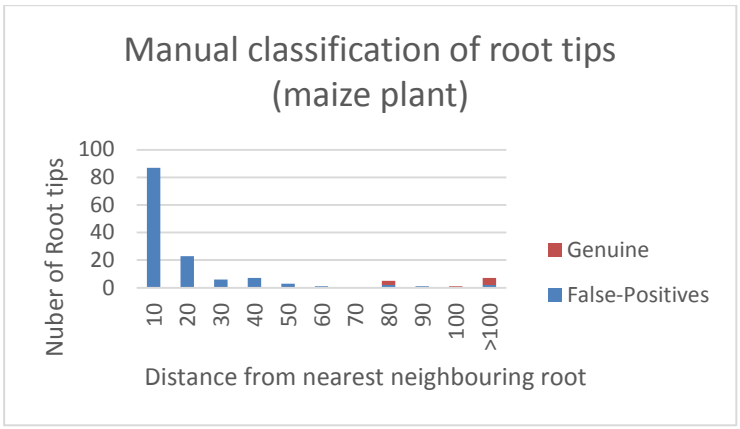
Whilst the root tip detection algorithm is guaranteed to detect all genuine root tips, it is unfortunately prone to detect false-positive, *parasite-like* root tips, introduced by noise in the segmented image. The majority of the cases are areas in the root system where pore space cavities are erroneously considered part of the root system and thus form small parasite-looking branches. Figure 3.12a illustrates all the root tips which have been detected using the above described procedure. As can clearly be seen, many root tips are detected, the majority of which are false positives.

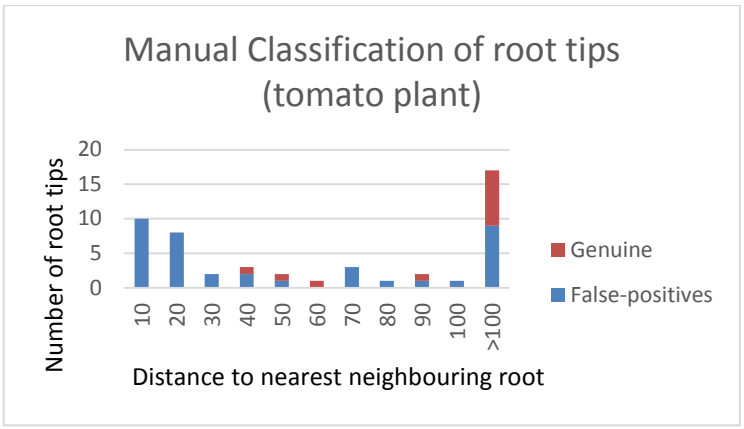


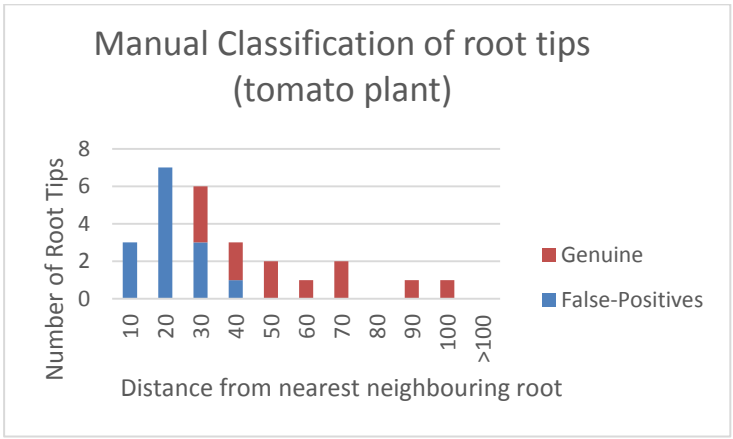
A) B) C)

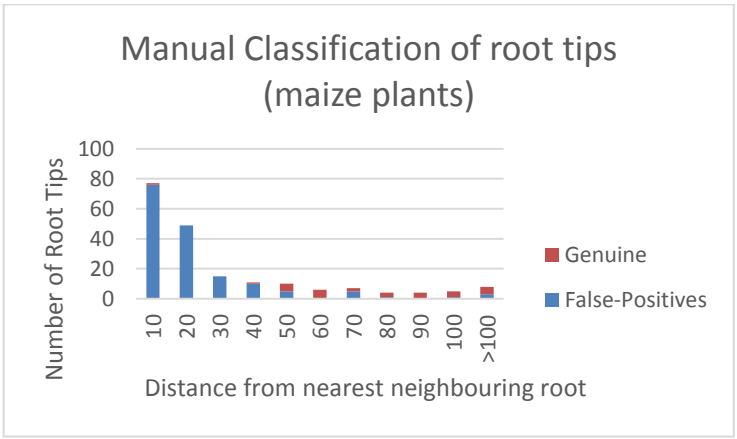
Figure 3.12 Example of spatial distribution of false-positive root tips.
A) All Root Tips Detected by the Root Tips detection algorithm – large number of false-positives especially around the kernel of the pan.
B) Genuine root tips left after the application of false-positive heuristics.
C) Pruning threshold set too high resulting in large number of false-negative genuine root tips omitted.

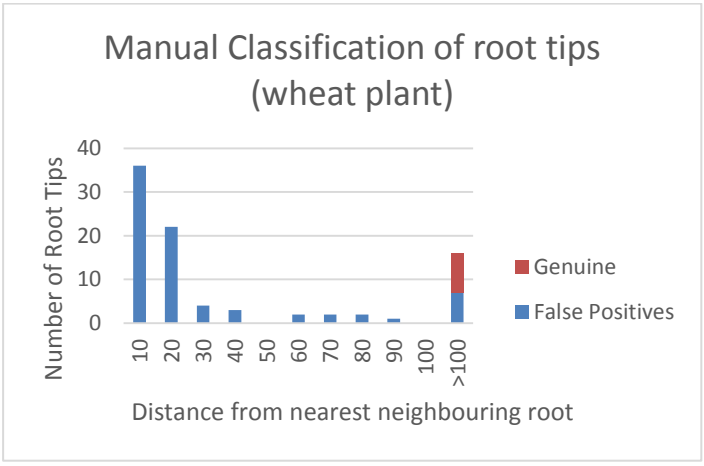
False-positive root tips are pruned by application of a set of heuristics. Due to their nature, false-positive root tips are easily distinguishable from the genuine ones. Simple heuristics are to limit the distance between the centroid of a root tip and the centroid of the nearest neighbouring root cross-section, and the distance between the root tip and the nearest point on the neighbouring root's wall. Figure 3.13 illustrates the relation between that distance and how likely it is that a root tip is false-positive or genuine and Figure 3.14 illustrates the different types of heuristics used.











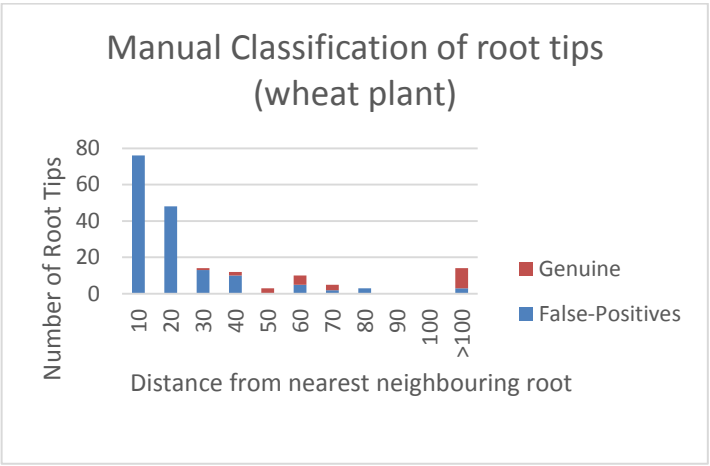


Figure 3.13 Comparison of manual classification of root tips with automatic false-positive detection heuristics. The blue bars represent the number of false-positive root tips detected whereas the red bars are root tips which have manually been classified as genuine by inspecting the 3D visualisation of the root system. Using a simple heuristic by measuring the distance between the centroid of the root tip and the nearest neighbouring root cross section it becomes clear from the graphs above that the two classes of root can be separated with relatively high degree of accuracy by using a single threshold. The distance to the nearest neighbouring root is chosen as parasitic root tips usually occur very close to the surface on the root wall of genuine roots and as such are easily pruned

A simple threshold put in place at the right level will be able to classify the root tips into genuine and non-genuine category with sufficiently large accuracy. The level of thresholding is set to a small number of voxel units (under 20), which is a standard practice for many pruning heuristics. If a pruning threshold needs to be set to any higher value it is unlikely to be efficient at separating the two classes of objects in question. The RooTh tool further allows the user to manually set the value of the pruning threshold, through a GUI interface, if deemed necessary for a particular scan.

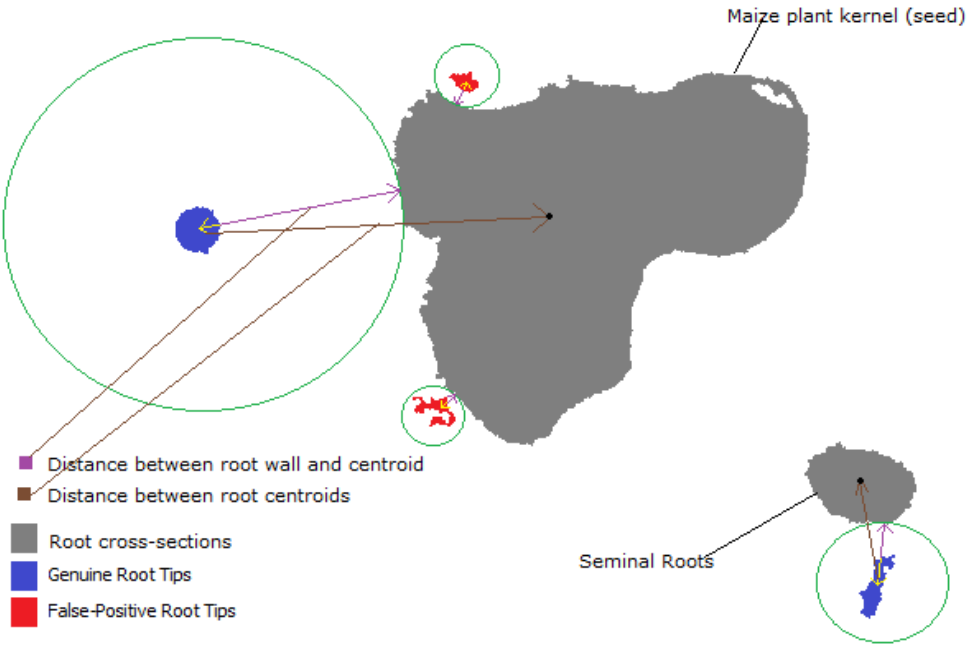


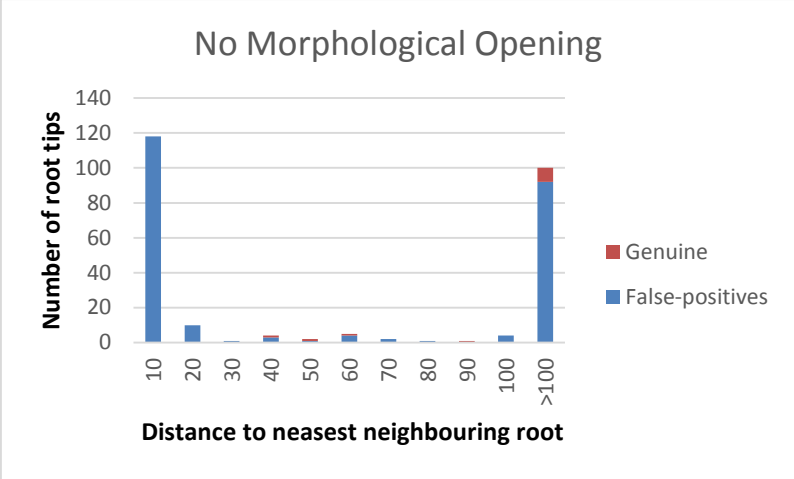
Figure 3.14 Root tips heuristics

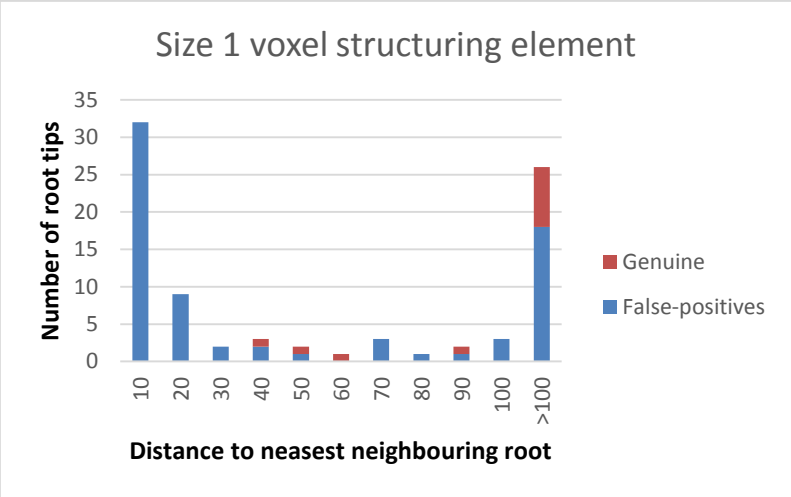
As can be seen from Figure 3.13, there are a relatively low number of genuine root tips and the vast majority of them are located far away from neighbouring roots. An exception to this is when roots grow near one another resulting in one of the roots' tip terminating in very close proximity to the other root's main body. In this case the root tip may wrongly be classified as false-positive even though it is a genuine tip. A secondary heuristic is used based on the length of the root and assumes that genuine roots are significantly longer than parasitic roots, thus allowing the above root to be classified as true-positive. Further to this, it should be noted that since the value of the pre-set pruning threshold only needs to be a very small number of voxel units for it to be effective at classifying the root tips, the two roots need to come very close to one another in order for the root tip of one of the roots to

be wrongly classified as false-positive. If that were to happen, the segmentation algorithm or the manual segmentation approach used is likely to fuse the two roots together. If that doesn't happen either, the morphological smoothing pre-processing step is likely to do that (the case of crossing or touching roots and the means to deal with such cases are discussed in Chapter 4). For that reason, false-negative root tips resulting from the application of the root tip detection are rarely seen in real-world data.

3.1.2.4 Evaluation of the effect and sensitivity to initialisation of morphological operators have on the false-positive root tip detection algorithm

Prior to the output of RooTrak or manual root segmentation being fed into RooTh, a pre-processing morphological opening step is automatically applied in order to ensure the root system is fully connected and free from artefacts in order to minimise the amount of false-positive root tips generated. A small (1 or 2 voxels wide) cube is used as a structuring element which is set as a default setting in RooTh, though the user has the option to manually adjust it if they have encountered a particularly bad batch of root data to skeletonise. This section evaluates the sensitivity of the false-tip detection algorithm to the amount of morphological opening applied to the segmented root system. To do that, different levels of morphological opening are applied to the same plant and the resulting root tips are manually classified and plotted in a diagram similar to that of Figure 3.13. The results are shown in Figure 3.15:





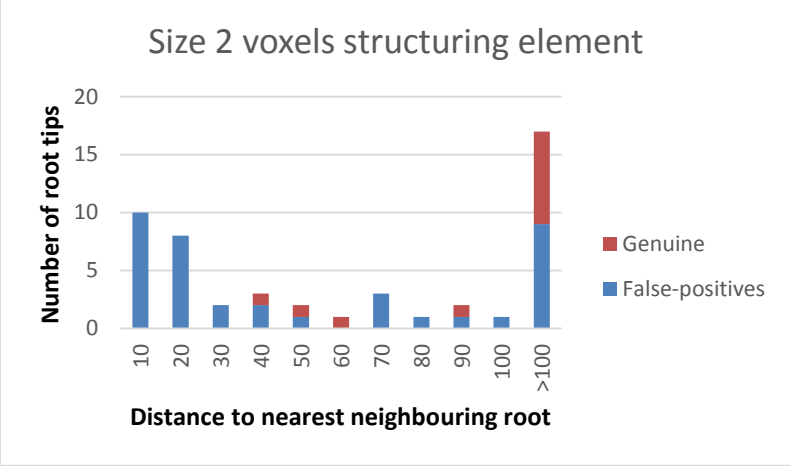


Figure 3.15 Different levels of morphological pre-processing and their effect on the number of false-positive root tips with relation to the number of genuine ones. The blue bars represent the amount of false-positive root tips and the red bars are the number of root tips which have been marked as genuine using manual classification. Using larger structuring elements to perform morphological opening results in significant reduction of the overall number of false-positives.

As can be seen from figure 3.15, if no morphological opening is used over the data, large numbers of false-positive tips are formed. They largely appear in two categories – those which are very close to neighbouring roots and those which appear relatively far away from neighbouring roots. As previously shown in Figure 3.8, the second category is largely a result of disconnected roots. Whilst the first category (the large blue bar on the left) can easily be eliminated using the false-positive root heuristics by thresholding the distance from the nearest neighbouring root, this approach will not work for the later one. However, it becomes apparent from the second and third diagrams in Figure 3.15 that larger structuring elements significantly reduce the overall number of false positive root tips as well as the number of false positive root tips emerging from disconnected or poorly connected roots. Therefore, morphological opening pre-processing is needed, combined with the false-positive heuristic thresholding described above and further pruning of very short roots in order to effectively eliminated all false-positive roots and only leave genuine ones out before further skeletonisation of the root system is applied.

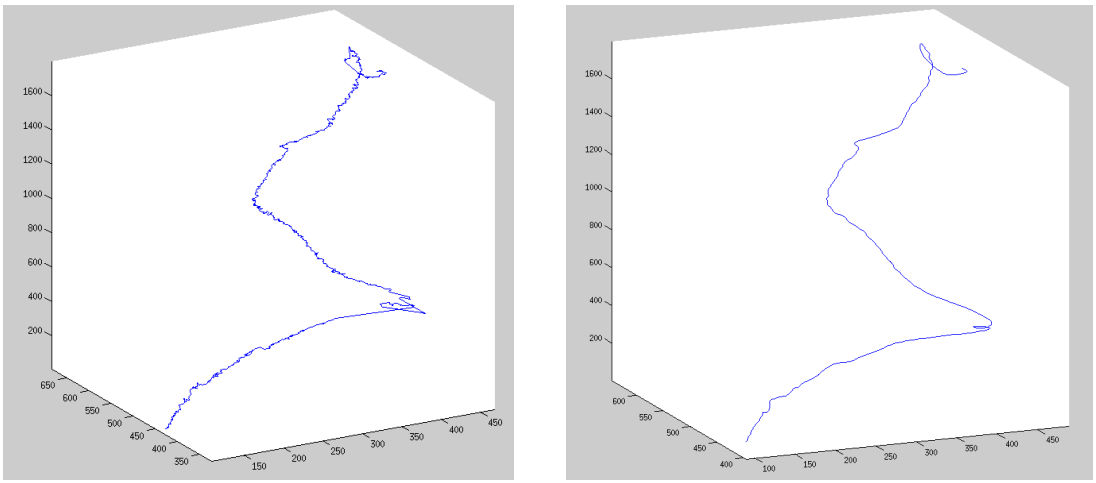
3.1.2.5 Control point selection and non-parametric linear curve-fitting

Once the locations of all genuine root tips have been found, it is necessary to recover the paths from those tips to the top of the root system in order to establish the root skeleton. The skeleton and thus the paths will need to meet all the skeletonisation criteria discussed in Section 3.1 in order for it to be suitable for determining root traits. One way of reconstructing the path would be to think of the 3D image stack as a graph, in which every voxel is a node and is either 17 or 28-way connected with its neighbours. The path will then consist of the shortest path through voxels connecting the root tips with the stem. If the distance map of the root system is taken into consideration, then Dijkstra's shortest path algorithm can be deployed. A similar technique is used by the RootNav tool [35] which recovers the path of roots in 2D image of plants taken over a static background using standard camera equipment.

An alternative approach is to use non-parametric regression to fit a smoothing curve passing as near as possible to all the root centroids and at the same time maintaining the smoothness and equidistance properties required to meet the skeletonisation criteria. This is the method adopted here. In order to fit such a curve, first the centroids of every root

cross section need to be located. The centroids, like root tips, are determined using the 3D distance map values of the root cross-sections. The highest values in each cross-section of a root correspond to the voxels furthest away from the root wall and can be assumed to denote root centroids. By matching the nearest centroids between root sections in two adjacent CT slices, the root path is traced up from the root tip all the way to the plant stem. The final task in determining the root system skeleton is then to find the best fitting curve passing through those centroids.

The aim of the skeletonisation method described in this chapter is to find the root path most equidistant from the walls of the roots. Such paths will therefore have to pass exactly through every centroid in the roots. In reality, since the segmented root data is generated from real world CT scans of plants containing small levels of uniform noise, root walls will display a lot of tiny local variations. Rather than having a smooth round shape, cross-sections are likely to be uneven as illustrated in Figure 3.16. Consequently, the cross-sections' centroids will be slightly offset from one another when consecutive CT slices are compared. Thus, if skeletonisation is defined as finding a path which strictly follows the centroids, it will result in a zigzag, highly-variable curve which will fail to accurately preserve the morphological structure of the underlying roots and would not meet the smoothness requirement. If such a curve is then used to measure e.g. the length of a root it will contain significant errors i.e. that curve can no longer be considered to be homotopic.



A) B)

Figure 3.16 Comparison of linear interpolation with smoothed curve-fitting
A) A single seminal root fitted with linear interpolation by connecting all control points sequentially
B) The same seminal root fitted with a smoothing curve resulting in improved skeleton representation

In order to find a better solution of the problem, an optimal curve which goes as near as possible to all centroids and at the same time and is as smooth as possible needs to be identified. There are two broad ways to represent a curve. One method is using a mathematical function *y = f(x)* which given a set of parameters a, b, c, etc. for every coordinate x will return a value of y corresponding to a continuous curve; in this case the curve fitting algorithm will correspond to a parametric regression. Parametric regression involves (given a set of points) determining the best values for the function parameters such that the function curve will be the best possible fit to the data points.

Parametric regression requires prior knowledge of the function which one desires to fit over a set of points. When trying to fit a curve through the centroid points of root cross sections it will be very hard to identify a generally applicable model of a root branch and describe it with a single mathematical function. In cases where it is impossible or very hard to have a prior knowledge of the function describing the data points, an alternative curve fitting method is to use non-parametric local regression.

Non-parametric regression involves fitting a smoothing curve solely based on the data points using localised piecewise function which is applied sequentially to every group of around three to five adjacent points. A key element of a non-parametric regression is that it allows the inclusion of a smoothing parameter. The smoothing parameter is usually a floating point coefficient ranging from 0.0 to 1.0 and it determines how well the fitting curve will fit (how strictly it will follow) the data points, in this case the control points extracted from the distance map of the segmented root system.

3.1.2.6 Defining fitted-curve smoothing parameter

Various local functions can be used, including smoothing bi-cubic splines, 2^{nd} or 3^{rd} degree polynomials [49] etc. A simple local regression function is the nearest-neighbour technique. The method is used over CT image stacks of roots consisting of sequentially iterating through the individual CT slices starting from the slice containing a root tip and consecutively scanning slices upwards towards the slice containing the stem. At each step the location of the curve in that section is set to match the closest root centroid point. The smoothing parameter determines how close to the centroid the curve will move. Low values

for the smoothing parameter (near 0.0) imply very little smoothing and thus the curve jumps all the way to the nearest centroid at each slice, effectively resulting in a zigzag curve. High values on the other hand imply high level of smoothing; the curve will attempt to maintain a straight line and only slightly be offsetting from its trajectory.

3.1.2.7 Using Cross-Validation for automatic smoothing parameter derivation per root system

The smoothing parameter allows the local regression algorithm to adjust the smoothness of the resulting curve. A key aspect of the non-parametric regression method is to choose the right value for the smoothing coefficient to ensure that the local regression will produce a desirable skeleton. An optimal best-fitting curve is one that would accurately pick up the general root model of the root system whilst avoiding the local noise in the 3D coordinates of the root centroids caused by the rapid variations of the wall boundary between adjacent cross-sections. The smoother the curve, the less noise it will contain but it would also deviate further away from the path of the root, and vice versa. The less smoothing is applied, the better the curve will follow the root path and at the same time preserving more of the noise. A trade-off between those two factors will produce the optimal curve.

In order for the non-parametric regression to find “perfect middle” for the smoothing parameter, it needs two types of data points sets – training and testing sets. The training set is used to produce a curve with any given smoothing parameter and the testing set is used to quantify how well the curve has managed to accurately represent the root model whilst avoiding local noise variations. The mean square error of the distance between the curve and the test centroid points is used to quantify how well the curve fits the data. If the curve under-fits the data it will not have accurately picked up global trends of the path undertaken by the root and thus it will result in high values for the mean square error. Similarly, when the curve over-fits the data then it will follow the path of the control points too well, thus missing out the testing points by large error again resulting in large mean square error. The global minimum of the mean square error will normally lie close to the smoothing parameter which produces an optimal curve i.e. one that contains all major features of the root model and which avoids local fluctuations.

When fitting a curve though the centroids of root cross-sections, there is only one set of data points and that is the set of centroid coordinates. In order to generate a testing and a

training set of points, the set of centroid coordinates needs to be split between the two. Both the training and the testing set need to contain points evenly sampled from the entire set of centroid coordinates to ensure that they still accurately represent the original set. The best way to do this is to use a random number generator to classify each data point as either belonging to training or testing set.

The above method of generating the training and the training set implies that every time a smoothing parameter is evaluated, the result will be slightly different as the content of the training and testing set would have been randomly chosen. Better results could be achieved if each set is used both as training and as testing one and then the results are averaged out. This method is known as cross-validation and in its simplest form it splits the set of points into two subsets and then each set is used to fit a curve and the other is used to estimate the mean-square error. Then the two sets are swapped, the mean square error is calculated again and the results of both runs are averaged. A K-fold cross-validation splits the set of centroid points into K subsets; each of K-1 subsets are used as training set and the remaining 1 set is used as a testing set. The process is repeated K times and the results of each run are averaged. When the value of K is equal to the cardinality of the set of points, the cross-validation method is known as Leave-One-Out cross-validation (LOO-CV). LOO-CV does not require the use of pseudo-random number generator to split the set and thus will ensure the optimal smoothing parameter is found every time it is run. The downside of it is that it requires N runs, where N is the size of the data set (the number of root centroids for every root in the root system) resulting in higher computational cost than the K-fold CV. In practise a trade-off between the two is used to balance computational complexity with accuracy [50].

In summary, the root skeletonisation algorithm used in RooTh consists of a linear smooth curve-fitting used to recover the skeleton of the root system from the set of control points extracted from a single bottom-to-top sweep of the 3D image stack. The control point extraction process also recovers a tree-like structure of the root system which is subsequently used to determine the fractal representation of the root architecture as described in the following chapter. The optimal value of the smoothing coefficient is automatically determined using k-fold Cross Validation by initially splitting the set of control points into 50 classes of randomly selected points (the user can choose a different value for

K from the RooTh GUI interface if deemed necessary for a particular scan). The Cross Validation operation is run *K* times with each replicate choosing one of the classes as a testing set and all remaining classes as the training set. A brute-force set of all smoothing coefficients within the 0.0 to 1.0 range with a small step of 0.01 is calculated for each replicate based solely on the training set points for that replicate and the testing set is used to evaluate the best smoothing coefficient. An average of all replicates is then determined to derive the smoothing coefficient of the Cross Validation method. This curve-fitting algorithm is run for one last time with all control points and the value of the smoothing coefficient derived above and the resulting is the skeleton of the root system.

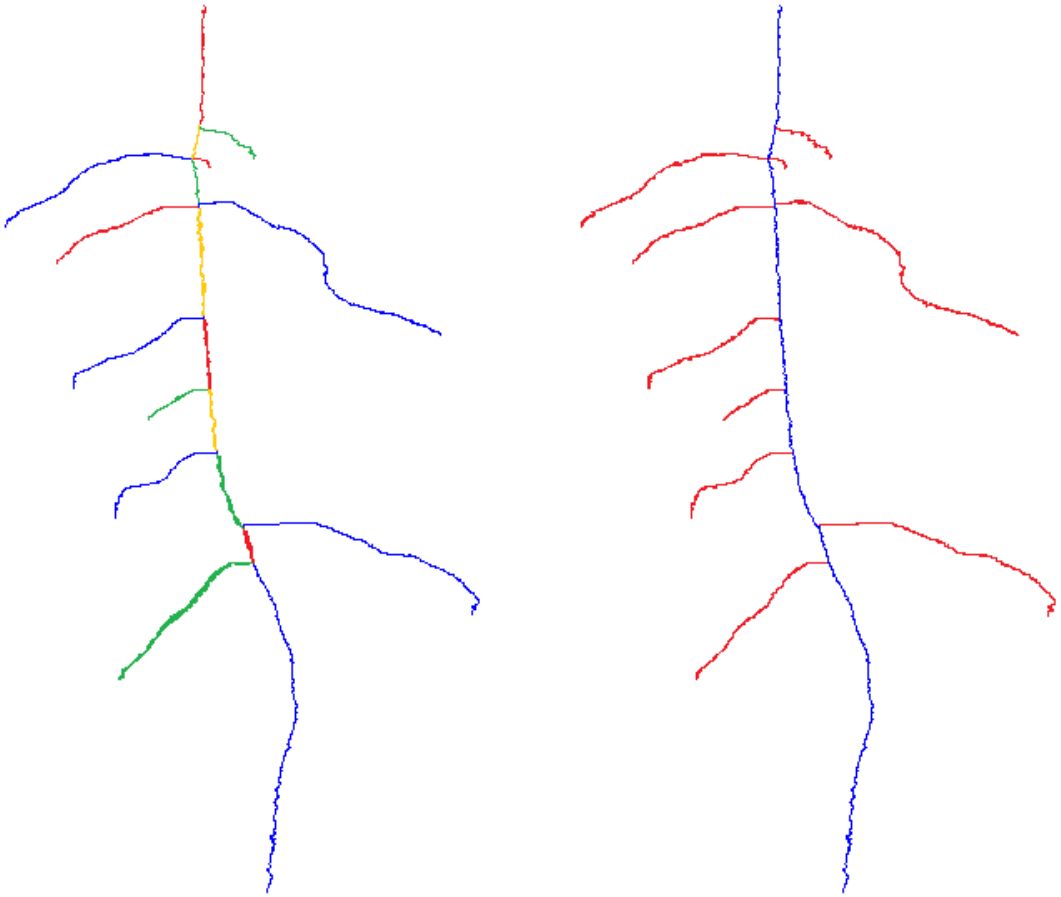
3.1.2.8 Root System Fractalisation

In order to evaluate certain root traits, like root length, in some cases it may be desirable to convert the skeleton representation to a fractal form. The tree-like representation produced by curve fitting encodes individual roots only to the point where each section diverges into lateral roots. At the point at which a lateral root emerges from the primary root, a new node is formed. Every root growing out of the node is given an equal weight and is treated as a separate branch respectively.

The above representation does not always accurately reflect the nature of some common root systems, particularly those [51] where primary roots grow from the seed of the plant and terminate with root tips and lateral roots emerge from the primary ones. The fractal description of an RSA is naturally better suited for measuring architectural traits like root length or average root width per root for instance. It isn't always possible, however, to recover the full root path from a root tip to the point of origin for every root and correctly determine the parent link at each branch along the way of the root. In some cases a "wholly arbitrary decision must […] be made as to which of the two roots emanating from a junction is of higher order" [51]. RooTh uses a simple recursive method for fractalising – representing complete base to tip roots the root system – in which it assumes there is one primary root which is parent to all other lateral roots and each lateral root can by itself be a parent of more sub-lateral roots.

The fractal representation as depicted in Figure 3.17b is computed using a recursive depth-first traversal algorithm where at each iteration, the longest path between a point of origin

and a root tip is marked as a parent root and all remaining roots emerging from the path are recursively fractalised in a similar manner. The above approach doesn't attempt to recover a biologically accurate representation of the root system of any one particular plant species. Its overall aim is to generate a more natural root system description. Some simple root systems of plant species like tomatoes, for instance, which have one main taproot would certainly be a better fit for the above description model than other species like wheat which usually have multiple initial seminal roots where every node of the root skeleton tree is visited first and its length evaluated before returning the fractal representation of the skeleton.



A) *B)*

Figure 3.17 Illustration of Tree-like root skeleton vs Fractal-like representation of the root system
A) Tree structure – every individual section connecting branching points is a separate node in the root skeleton
B) Fractal structure – every node in the root skeleton is a complete root originating from the emergence point of the root and terminating at its root tip. All lateral roots are derived from their corresponding parent root in a nested hierarchy

In older, more complicated, root systems like those of wheat plants crossing or touching roots of result in additional false “root branches” being formed in the root system. This will commonly result in incorrectly fractalised roots, in which the paths of two crossing roots are swapped with one another due to the depth-first traversal algorithm seeking the longest-path. In order to correct such errors, the user is presented with an opportunity to manually annotate individual root branches.

The manual correction is high-level and it involves highlighting entire root segments and determining whether they should be part of the same root or not (as opposed to annotating individual cross-sections). The user is presented with a 3D rendered view of the root system skeleton in the RooTh GUI 3D Root System Viewer/Editor as depicted in Figure 3.18. The user has the option to consecutively iterate through the individual roots by highlighting the currently active root. This is done until the root that the user wishes to correct or manually annotate is selected. Initially all roots are coloured grey and the currently selected root is coloured in red. The user can use the mouse wheel to scroll through adjacent root sections of the active root (section of root skeleton connecting two branching points or a branching point and a root tip) by highlighting the currently selected root section in yellow, the ones which the user has scrolled-past in green and the remaining ones in red. At the branch pointed by the highlighted root section (the branch point at the end of the yellow section) the GUI highlights all alternative root paths in blue. The user is free to make a selection to any of them by highlighting them in yellow and thus making them the currently active branch. Once the user is finished with their selection they can confirm their choice and the entire root system fractalisation algorithm is re-run taking into account the user selection in order to arrive at a new root system architecture incorporating the manual annotation.

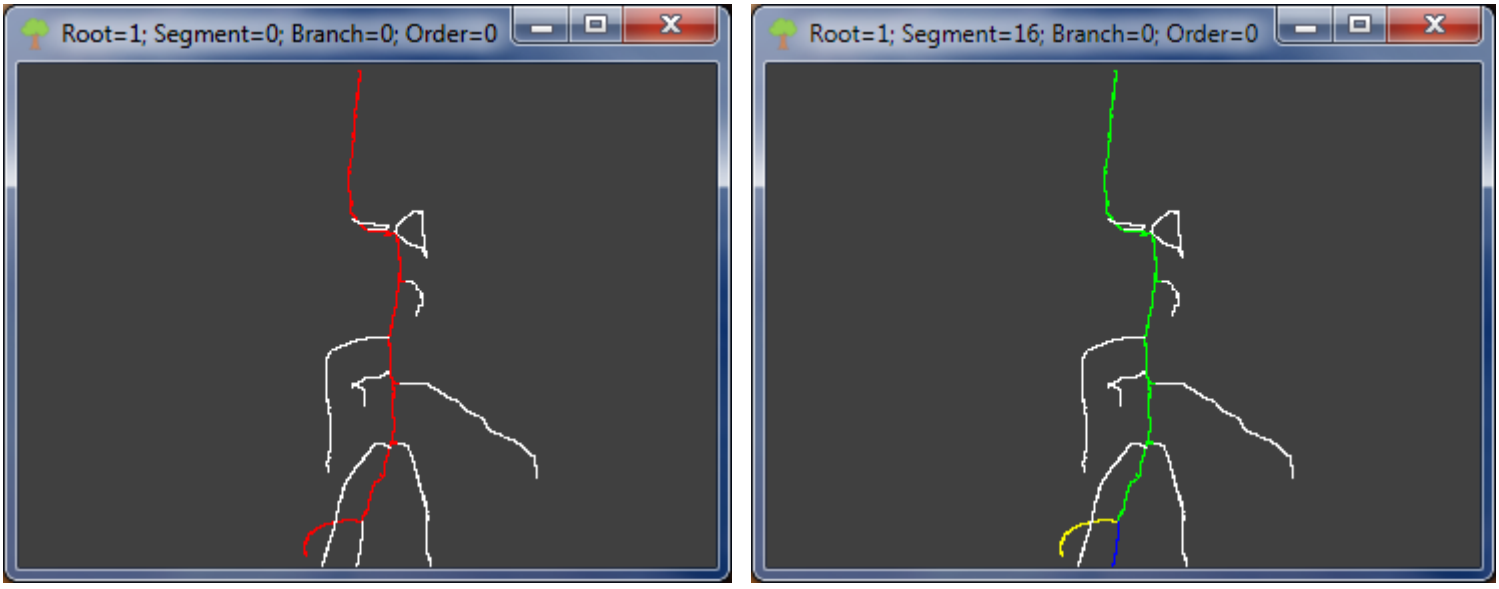


Figure 3.18 Example of manually annotation for correction of root path (RooTh 3D Viewer)

3.2 Evaluation and Results

In order to evaluate the efficiency and accuracy of the proposed skeletonisation method, a validation experiment was conducted consisting of a set of plants grown in soil columns and imaged using two independent methods – the more traditional 2D image scanning following root washing and analysis using the WinRhizo platform [27] (refer to Chapter 2 for more details of the method) and the novel approach of non-invasive CT reconstruction followed by RooTrak segmentation and RooTh skeletonisation. The goal of the experiment is to compare the two techniques and to evaluate how well they agree on the extracted root trait. It is expected that if there is good correlation between the independent evaluation of global and local root traits that could be indicative that both methods are producing results close to the ground truth and that in turn could be used to validate the accuracy of RooTh output. Furthermore, since the results are being compared against WinRhizo, which is a well-established 2D commercial root quantification tool and has been for over a decade, it could be expected that the output it produces would have been somewhat independently calibrated against the ground truth by its creators. We suggest that if there is close agreement between the values of traits accessible to both methods, it is reasonable to assume that any further, 3D traits recovered from RooTh output will also be sufficiently accurate.

For the purpose of the experiment three species of plants were used as follows – Jubilee F1 maize (*Zea mays L. convar. saccharata var. Rugosa*), Cordiale winter wheat (*Triticumaestivum L.*) and Aisla Craig tomato (*Solanumlycopersicum L.*). Two replicates of each plant were grown in two sets of soil types – clay-loam and loamy-sand. The soil was prepared and sieved with a 2mm sieve before been used to fill 30mm soil columns. The seeds were germinated for two days on wet paper trays before being sown in the columns and grown for 9 days in a controlled environment (a growth room). Subsequently the soil columns were scanned in the mid-size Nanotom v|tome|x scanner at 25µm resolution in a single day and the plants were then root-washed on the same day and placed in cold water bags to suppress further growth. The root-washed root systems were then scanned on a conventional flat-bed 2D computer scanner and analysed with the WinRhizo software tool within the next two weeks. The aim of the experimental set was to provide a diverse set of plant species and scanning conditions (i.e. different soil types) in order to better represent

the real-word deployment of RooTh in future biological experiments. This particular experiment carries no biological significance and all seeds were of wild type, equivalent to the control samples in a normal plant science experiment. The resulting image skeletons of the 6 replicates are shown in Figure 3.19.

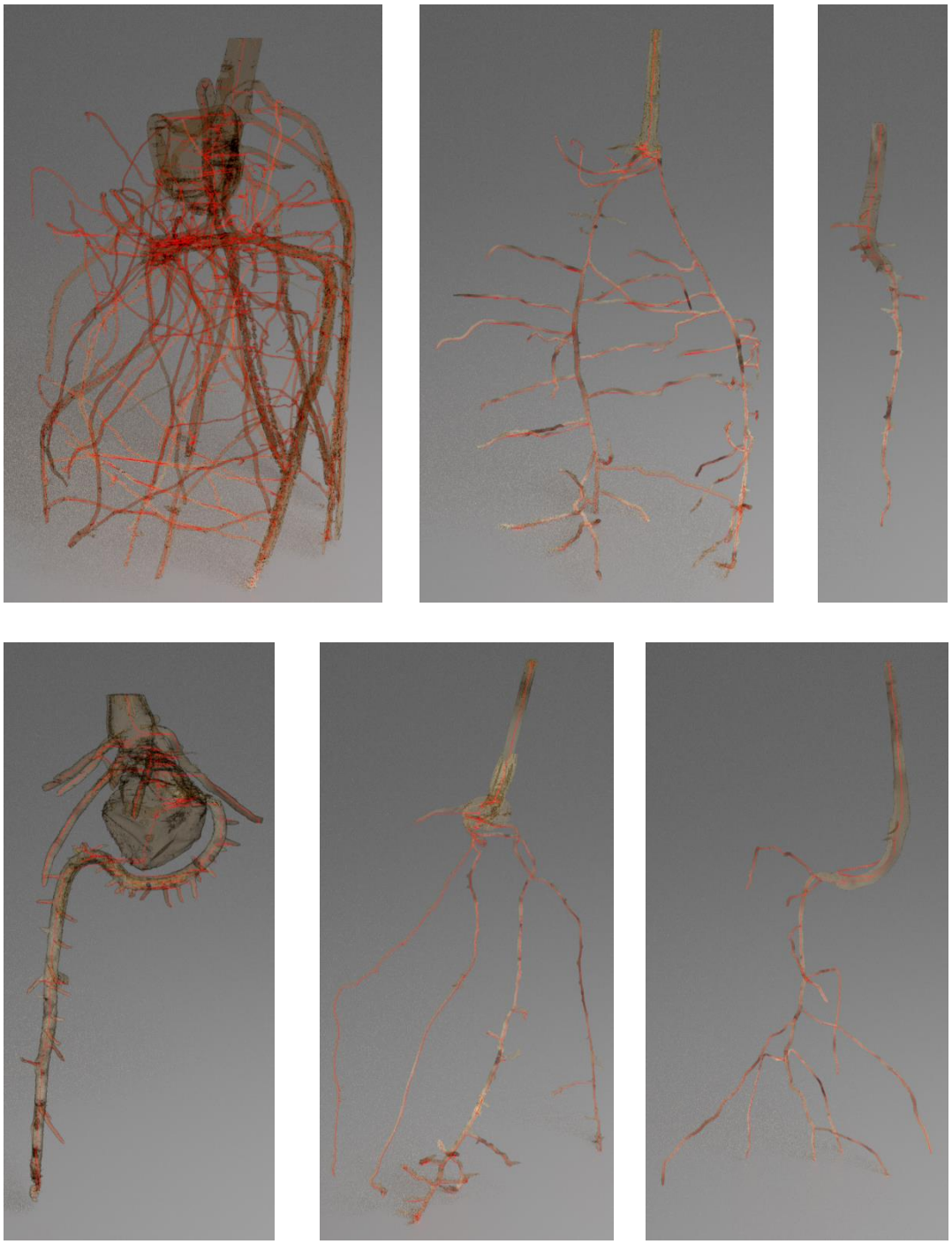


Figure 3.19 Curve-fitted root skeletons of the scanned plants in the validation experiment (Left column – maize plants, middle column – wheat plants, right column – tomato plants; Top row – clay-loam soil, Bottom row – sandy-loam soil)

A summary of the results of the experiment is shown in Table 3.2. There is overall a good correlation between the two methods suggesting that both provide a good estimate of the ground truth. The scale of the experiment was small and does not allow us to conclude with statistically significant certainty that the two methods are correlated due to the very small number of replicates, nevertheless it appears there was a very good match between the measurements for both wheat and tomato plants.

Plant	CT+RooTrak+RooTh		Scanner+WinRHIZO	
	Total Root Length	Average Root Width	Total Root Length	Average Root Width
Tomato Rep 1	240mm	0.398mm	291mm	0.366mm
Tomato Rep 2	67mm	0.689mm	34mm	0.807mm
Wheat Rep 1	599mm	0.446mm	795mm	0.574mm
Wheat Rep 2	377mm	0.511mm	493mm	0.566mm
Maize Rep 1	420mm	1.624mm	182mm	1.574mm
Maize Rep 2	2680mm	0.613mm	964mm	1.235mm

Table 3.2 Summary of root traits measured with WinRhizo and RooTh

The results for the maize plants however suggest that RooTh is overestimating the total length of the root system when compared with WinRhizo [27]. This should not come as a surprise if a closer look is taken of the maize root skeleton as depicted in Figure 3.19 where it can be seen the very high number of zig-zag oscillations appearing in the skeleton of the primary root, inside the maize plant kernel. Therefore, based on this observation it can be assumed that the maize series are outliers with respect to the validation experiments and due to the unique morphological characteristics of a maize plant (i.e. presence of large kernels in the segmented root data) RooTh underperforms for that plant species. Omitting the maize plant data improves the correlation between trait measurements of the two independent methods as it can be seen in Figure 3.20, where the values of the length and width trait outputs are plotted against one another. The significance of the correlation for all six plants is p=0.193711 and is therefore not significant. The significance of the correlation for the 4 plants omitting maize is p=0.0006 and is therefore a statistically significant correlation.

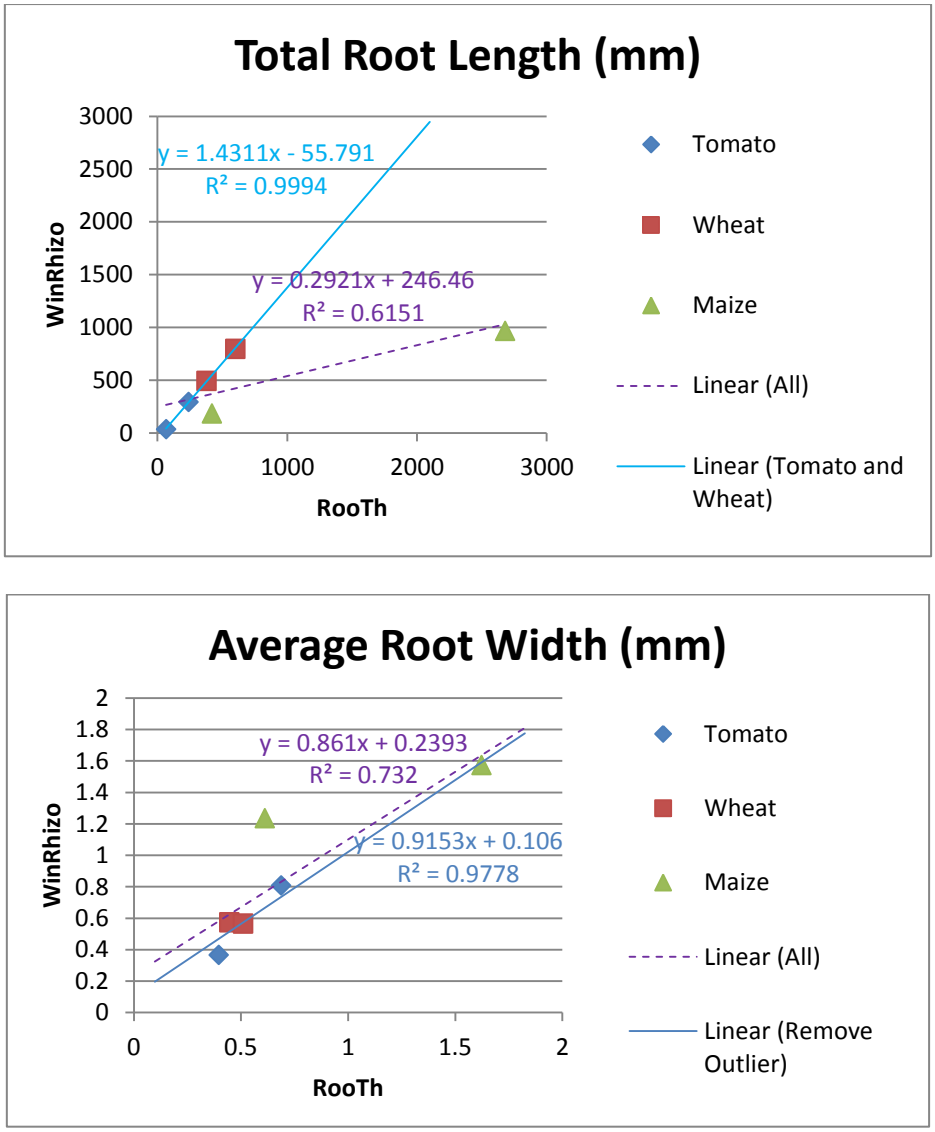


Figure 3.20 Correlation between the output of RooTh (horizontal axes) and WinRhizo (vertical axes)

Further evaluation of the accuracy of the curve-fitted skeleton was carried out by comparing the 3D coordinates of selected points on the skeleton to a sample set of ground truth points generated using manual skeletonisation. The manual skeletonisation of points was done using Fiji's orthogonal viewer tool. The orthogonal viewer allows a 3D volumetric stack to be projected from all 3 main axes; that is top down (XY projection), front-back (XZ projection) and left-right (YZ projection). Suitable cross sections of the root system were chosen using a method described in Chapter 4 and then the ground truth for that specific cross-section was estimated using Fiji's build in oval measurement tool, as illustrated in Figure 3.21:

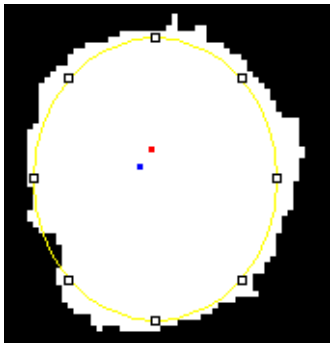


Figure 3.21 Fiji's built-in Oval Tool used to estimate the ground truth skeleton for selected root cross-sections

The average distance between the curve-fitted skeleton and the ground truth points for a sample of 15 root-cross sections per plant is shown in Table 3.3 below:

Plant	Tomato	Tomato	Wheat	Wheat	Maize
Difference (voxel units)	3.92	3.12	4.17	2.92	12.54

Table 3.3 Estimate of the difference between the ground truth of manually selected skeleton points and the corresponding curve-fitted points for a sample of 15 root cross sections per plant for 5 validation experiment plants.

The average distance of ~4 voxel units (~12 voxel units for the maize plant which is expected given that the maize plant has a relatively large kernel compared to the rest of the root system) suggests that the curve-fitted skeleton is a good approximation to the ground truth skeleton of the sample validation experiment plants. Further improvements to the accuracy of the skeletonisation algorithm are presented in Chapter 4.

It should be noted that due to the significantly higher complexity of the root architecture of the maize plants, both the RooTh and WinRhizo methods are likely to underperform, taking them further away from the ground truth and introducing greater variation into measured traits. In particular, a more sophisticated root system would partially prevent the proper 2D imaging, and therefore skeletonisation, of individual roots by WinRhizo. As can be seen in Figure 3.22, which shows the 2D skeleton as computed by WinRhizo for the top-left maize plant in Figure 3.19, whilst the tool is quite successful at picking up the majority of lateral roots, it is limited at recognising occluded roots, especially those lying near or behind the kernel, at the centre of the root system where the density of roots is relatively high therefore likely preventing it from accurately estimating the total root length in its own

terms. WinRhizo also appears to struggle when skeletonising the kernel of the maize plant, producing a high number of diagonal false-positive roots contained within the kernel which are likely to skew the root length estimation.

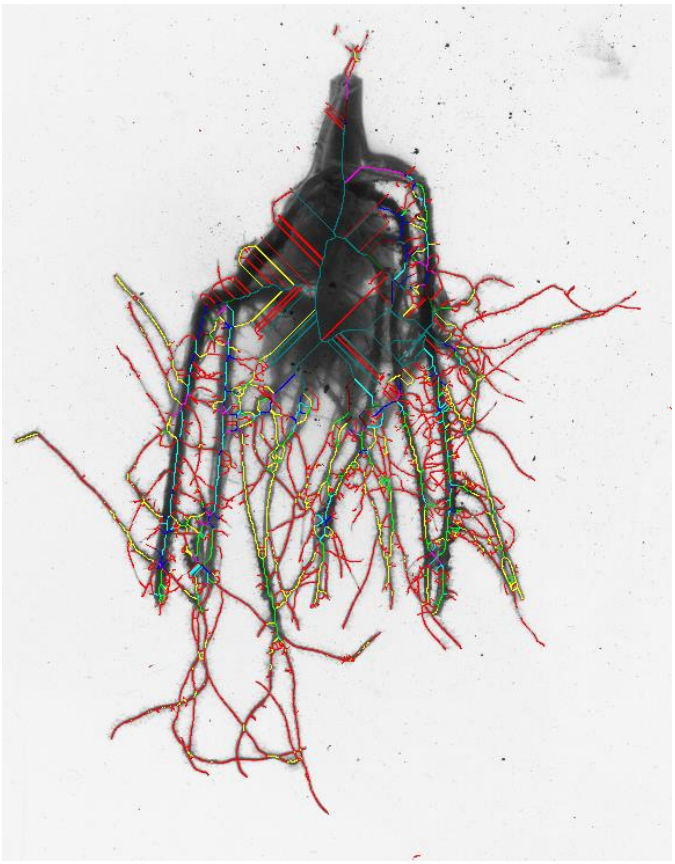


Figure 3.22 2D Skeleton output of the WinRhizo software tool for a maize plant showing inaccuracy in the skeletonisation of the root system

In conclusion, the experimental data suggests that the CT imaging approach combined with RooTrak and RooTh allows for the evaluation of root traits measurements comparable to those that would be calculated by the WinRhizo software if they were to root-wash the plant and use a 2D flatbed scanner at first place. The added benefit of using the CT approach, however, is that the CT method is non-destructive and therefore one may find it particularly useful to study 4D time-series or evaluate dynamic root traits. This is something that is not possible with the root-washing method as it is an invasive technique by nature.

3.3 Discussion

The linear curve-fitting skeletonisation method is suitable for processing image stacks of soil columns containing mostly downward or upward growing roots. In soil columns scanned with the small Nanotom µCT scanner, due to the restrictions imposed on the sample size, only very small columns measuring no more than 3.5cm can realistically be scanned. As a result, only very young plants, just a few days after germination are suitable to be imaged using this equipment. If scans are left to grow for a longer period after their germination point, the root system will overgrow and almost entirely occupy the volume of the soil pot, in effect taking the shape of the pot itself. Any scans will therefore be of little value for any meaningful phenomic analysis. At the very early stages of plant root development the only roots present in the image stack will be the primary and in some cases one more seminal root, all of which are gravitropic roots - that is their root tips can "sense" gravity and grow in a direction away from the surface of the soil essentially resulting in a root system mostly compromised of downward growing roots. The original RooTrak software was designed to work with image volumes of mostly downward growing roots for similar reasons.

The linear curve-fitting algorithm however lacks the capacity to deal with horizontally growing roots or horizontal sections of a root architecture. All control points are extracted from the minimum values of the distance map of the 2D root cross-sections as the image stack is scanned slice by slice from top to bottom. When the algorithm is scanning through CT image slices containing vertical sections of roots, the root cross section will appear as well rounded or elliptical enclosed areas and the centroid points of the distance map will be a good match for the control points guiding the smoothing curve which will subsequently represent the root system skeleton when it is generated in subsequent steps. Furthermore, due to the high number of cross-sections which will appear along the length of the roots as it is slices sequentially in each individual 2D image, the generated set of centroid points will contain a rich set of control points meaning that the smoothing curve won't be "starved" of anchoring points when the curve-fitting step is initialised and run, resulting in well-defined skeleton for those vertical sections of the root.

Horizontal sections of the root however present a unique challenge for the curve-fitting method. Horizontal sections of roots will be starved of control points forming a root

skeleton consisting mostly of straight lines connecting the vertical sections of the root which inevitably will result in a less descriptive skeleton and could also result in underestimating root length and other root traits derived directly from the root skeleton. The issue is particularly problematic for the plagiotropic secondary and lateral roots that are present in older plants and develop from the primary root 5 to 10 days after germination as can be seen in Figure 3.23 below.

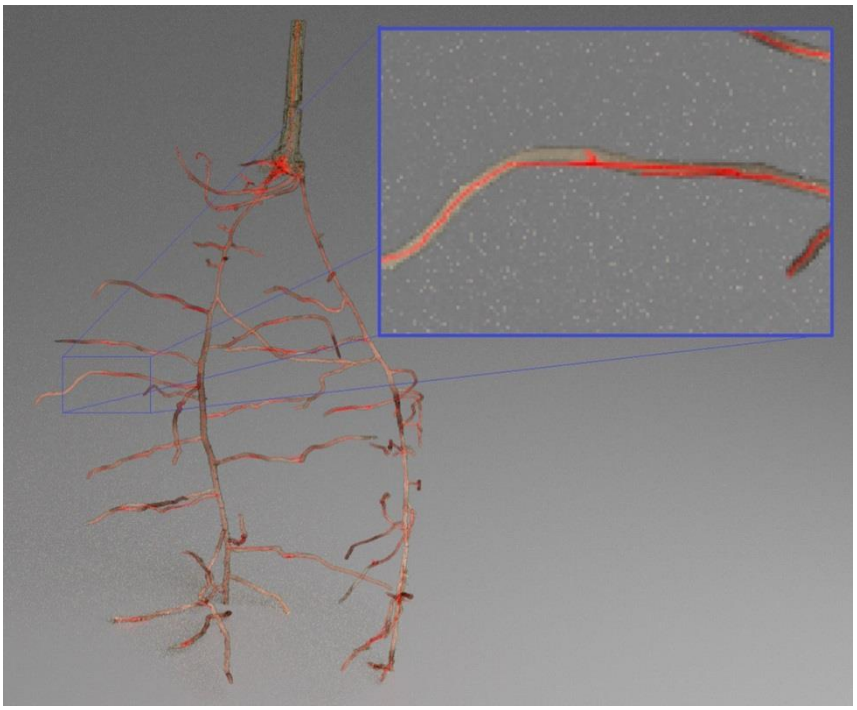


Figure 3.23 Horizontal (plagiotropic root) resulting in inaccurate, less descriptive curve-fitted root skeleton

In many cases such plagiotropic roots are of particular interest and importance and as such a better method is required in order to ensure they can be properly skeletonised and quantified. Further, newer versions of RooTrak deploy a multi-pass level-set tracking algorithm allowing for the efficient and accurate segmentation of such roots, removing any further barriers in earlier stages of the imaging pipeline that would otherwise prevent those roots from being skeletonised.

In order to address those issues, an improved skeletisation method based on multi-directional control point extraction and active contours was developed and is the topic of the following chapter. This allows the quantification of all segmented roots present in a CT image stack irrespective of the direction of growth.

3.4 Summary

This chapter presents a novel technique for the automated skeletonisation of segmented output of plant root systems based on linear smoothed curve fitting algorithm with cross-validation used for smoothing parameter optimisation. The segmented output data is used to calculate the distance map of the root system and the minimum values are used to create a set of control points from which a smoothed curve is derived representing the underlying root model. Automatic fractalisation is applied to the resulting skeletonisation tree based on longest depth traversal algorithm to segregate individual roots from end to end and manual annotation may be used to correct any inconsistencies. The method is only fit to skeletonise upward and downward growing roots and a more advanced approach is necessary to deal with the issue of horizontal growing roots based on multi-sweep control points' extraction and active contours which is presented in the following chapter. The software tool which implements the methods proposed in this chapter, RooTh, is released under the GPL licence and available on GitHub: https://github.com/rootth/rooth

Chapter 4.
Plagiotropic Root Phenotyping Using Active Contours

4.1 Motivation

Chapter 3 developed a method for the skeletonisation of 3D volumetric data of segmented root systems based on fitting non-parametric linear smoothing curves over a set of control points extracted from the distance map values of root cross-sections in horizontal CT image slices. The method uses k-fold cross-validation over the set of control points to automatically determine the optimal smoothing parameter with the aim of ensuring the resulting skeleton is smooth and equidistant from the root wall, improving trait estimation in subsequent steps. This method is well suited for RooTrak segmented data obtained from small sized soil columns (2.5cm to 5cm in diameter) where the root system is in its very early stages of development (very young plants) and consists predominantly of seminal roots (roots originating from the plant seed) which exhibit gravitropism and thus have vertical direction of growth in relation to the soil column. The vertical distribution of the growth directions of roots is the key to success of the curve-fitting method, as its non-parametric nature restricts its search space to a single dimension (the initial bottom-to-top sweep of the soil column).

Challenges arise when larger soil columns of plants need to be scanned. Large soil columns (10cm to 15 cm in diameter) allow for the imaging of more mature and thus significantly more complicated root systems where plagiotropic, horizontally growing lateral roots are much more common. Horizontally oriented roots are largely omitted by the control point generation method used in the curve-fitting skeletonisation algorithm, resulting in a significantly less descriptive skeleton representation of those roots. Since the imaging of the soil column is done in a vertical position, the cross-section of a segmented root growing upwards or downwards produces sections perpendicular to the direction of their growth. This results in oval root sections. The same cannot be said for horizontal roots, which appear as elongated slices along the length of the root in individual CT images. Entire sections of

horizontal roots also form only a handful of distinct cross-sections in the 3D image stack. That becomes a problem when the minimum values of the distance map in those cross-sections are then used to derive the set of control points guiding the fitted curves. The above severely deprives the root of control points distributed along its length. This chapter offers a different approach to the problem of skeletonising horizontal roots by the means of performing multiple sweeps in different directions within the soil column in order to extract a richer set of control points independent of the direction of growth of individual roots. The resulting set of control points has its own challenges, as it is no longer compatible with the non-parametric smoothing curves discussed in Chapter 3. Parametric curves guided by an Active Contour Model (ACM) are used instead to handle the multi-sweep set of control points and generate a smoothed, equidistant skeleton of the segmented root system.

Further challenges arise when the source of the root segmentation output is manual segmentation, like those generated using commercial tools such as VG Studio Max. Manual segmentation methods offer less accurate segmentations, resulting in a ridged root surface that affects the spatial distribution of control points and thus makes the problem of skeletonisation much more challenging. The higher dimensionality of the proposed ACM method, combined with the larger number of control points present in the multi-sweep data set, further help to deal with these issues.

The Active Contour technique also offers unique advantages when it becomes necessary to deal with broken or missing root segments. It does that by filling-in the gaps in the segmented data where it is expected that next to no control points will be present to guide the ACM interface, resulting in a fully-connected skeleton of the otherwise disconnected segmented data. It can also be extended to the analysis and processing of 4D time series, as discussed in Chapter 5.

The following description of the proposed plagiotropic root skeletonisation method based on Active Contours is split into 4 sections. Section 4.2 presents the technique for extracting a richer set of control points through multi-directional sweeping of the 3D volume stack; this is essential to the workings of the method. Section 4.3 presents and defines the nature of Active Contours, and the various force fields used in conjunction with the ACM interface, and introduces simulated annealing as a way of dealing with some common issues related to

snakes which are discussed in greater detail later in this chapter. Section 4.3 defines 3D splines and their application to the extraction and evaluation of root traits from the ACM interface. Section 4.5 consists of a summary of results and evaluation of the method presented in this chapter.

4.2 Plagiotropic Roots and Control Points

The curve-fitting method relies on control points extracted from the distance map of root cross-sections from a single top-to-bottom sweep. In horizontal sections of roots the number of control points extracted using this approach is significantly reduced as a ratio of control points per unit of root length. If a side view of the root system is used so that the same process is repeated from one of the sides of the stack to the other (i.e. front-to-back or left-to-right), the number of control points in those root sections will be significantly higher. However all vertical root sections will now lack a sufficient concentration of control points.

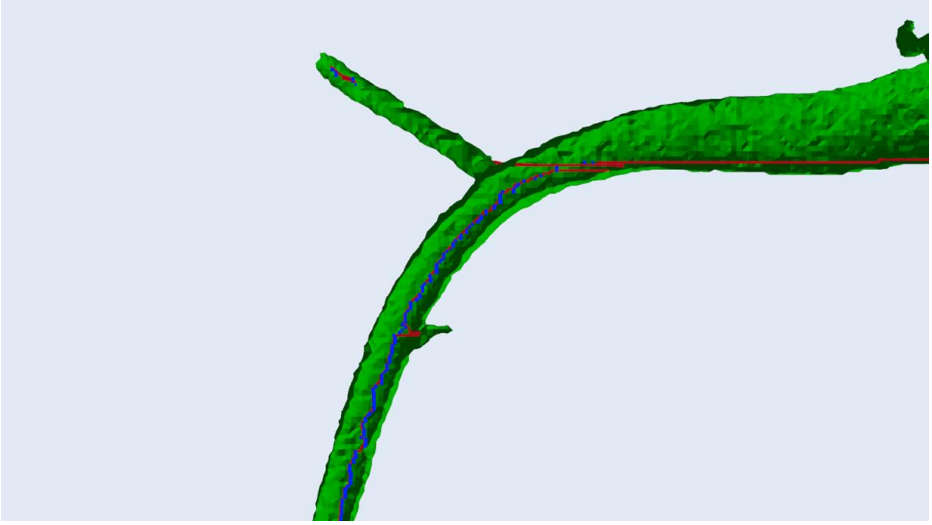


Figure 4.1 Lack of top-to-bottom scan control points in a horizontal section of a root resulting in inaccurate curve-fitting

Due to the nature of the root system and the presence of both gravitropic and plagiotropic roots growing in all directions it is often impossible to find a single angle to sweep the image stack such that all sections of the root system will be equally represented. Instead, the image stack can be swept from all three major axis (top-bottom, front-back and left-right)

and the resulting set of control points combined. The initial fractalised, non-parametric curve-fitted skeleton can then be used to group and separate the combined control points from all the sweeps and allocate them to their corresponding roots. This process is illustrated in Figure 4.2.

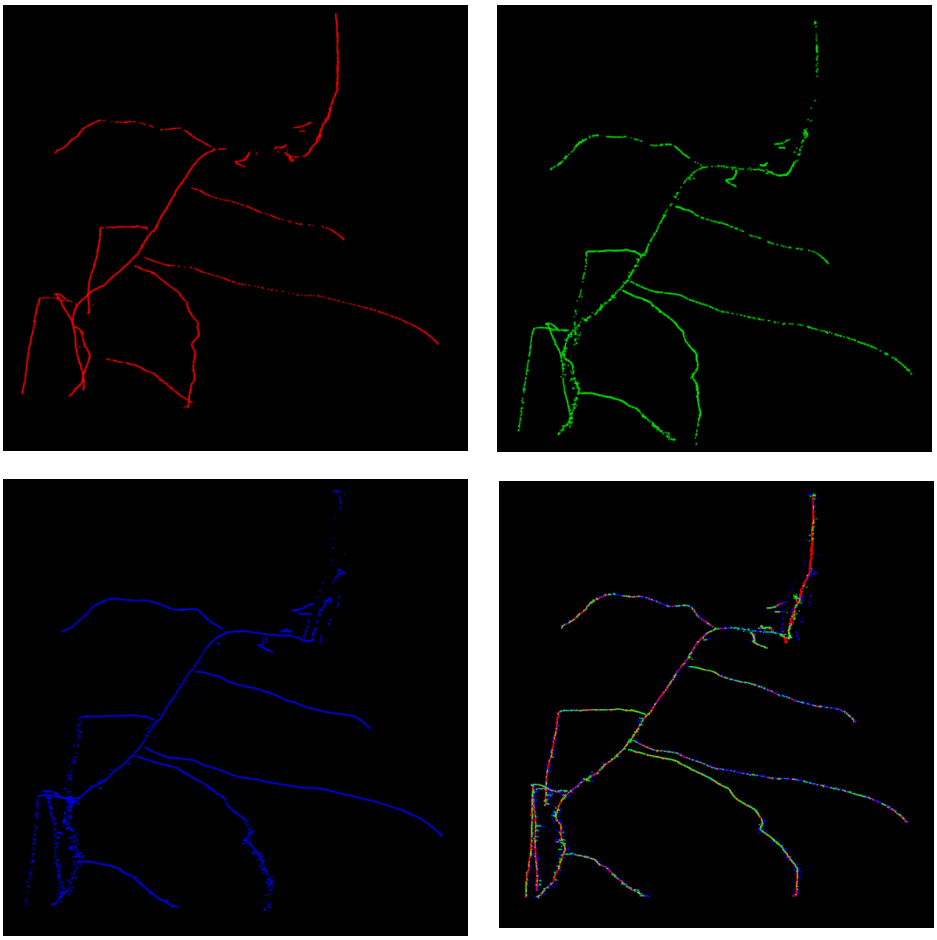


Figure 4.2 Side Sweep Control Points. Red Points – top-to-bottom sweep; Green Points – front-to-back sweep; Blue Points – left-to-right sweep; All three sets of points overlapped in bottom right image to reconstruct a much more complete representation of the root system with no gaps in control points in horizontal sections of roots.

This set offers a much richer and more complete representation of the root system, however, it is no longer possible to use non-parametric curve fitting over it. This is because the order of the control points along a given root branch is no longer known, as is the case with a single top-to-bottom sweep. A parametric spline curve is used instead, employing an Active Contour Model to align the spline with the control points.

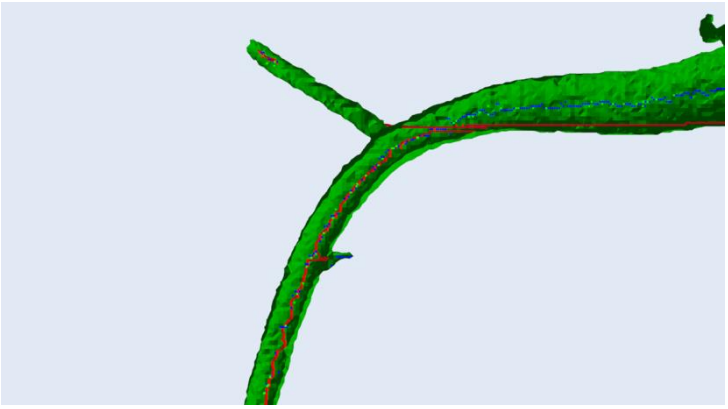


Figure 4.3 Side control points overlaid on top of a curve-fitted skeleton of a horizontal section of plagiotropic root illustrating that their use provides a richer and more accurate description of the underlying root model

4.3 Active Contours

An Active Contour Model is "an energy-minimising spline guided by external constrained forces and influenced by image forces that pull it towards features such as [the centroid points of a segmented root system output derived from the distance map of the 3D volume]" [52]. The energy minimisation function can be defined as a sum of three forces:

$$E = \int_0^1 E_{internal}\, ds + \int_0^1 E_{external}\, ds + \int_0^1 E_{constraints}\, ds$$

Equation 4.1 Energy Minimisation Function of the Active Contour Model

where the internal force operates between neighbouring snake points and determines the smoothness and stiffness of the ACM interface and thus the resulting skeleton. The external force is the force derived from the force field computed from the control points and the forces of attraction and repulsion between adjacent snake points, and ensures the propagation of the snake interface and its even reach and distribution to all turns and bends of the root as it evolves through the different iterations.

Active Contours use a set of points called snake points to control the shape and location of a cubic spline [53]. The snake points are subject to attraction forces towards the control points and repulsion forces from one another. Active Contour Models ensure that the first and second derivatives of the piecewise cubic splines are equal at the snake points, thus smoothing out the spline. The spline's interface is initialised as a straight line connecting

each root base with its tip. The model is then allowed to evolve over a number of iterations until it fits the set of control points. The user is then allowed to modify the model's parameters (scale of attraction and repulsion forces and stretch and stiffness coefficients) in order to produce an optimal fit. Alternatively, the optimal fit can be calibrated by re-initialising and running the model within a search space of combinations of parameters. The resulting cubic splines can be used for root system quantification and representation.

The ACM interface used by RooTh is an open-ended interface instead of the classic closed ending interface, meaning that the two end points of the interface are anchored or in fixed locations, rather than being linked together to form a closed loop. This could be considered a "constraint" force in the above equation, although in practice it translates to omitting to update the locations of the two end points of the ACM interface.

The internal forces of smoothness and stiffness are derived from the first and second derivatives of the polynomial piecewise cubic curves passing through the snake points. Their values can be calculated using the Euler-Lagrange Equation 4.2:

$$\frac{\partial x}{\partial t} = a\frac{\partial^2 x}{\partial s^2} - b\frac{\partial^4 x}{\partial s^4} = 0$$

Equation 4.2 Euler-Lagrange Equation for calculating the values of smoothness and stiffness of the ACM model

However, due to the difficulty of computing continuous integrals on a finite precision machine like a CPU, integration using finite differences based on the Pascal triangle can be used instead.

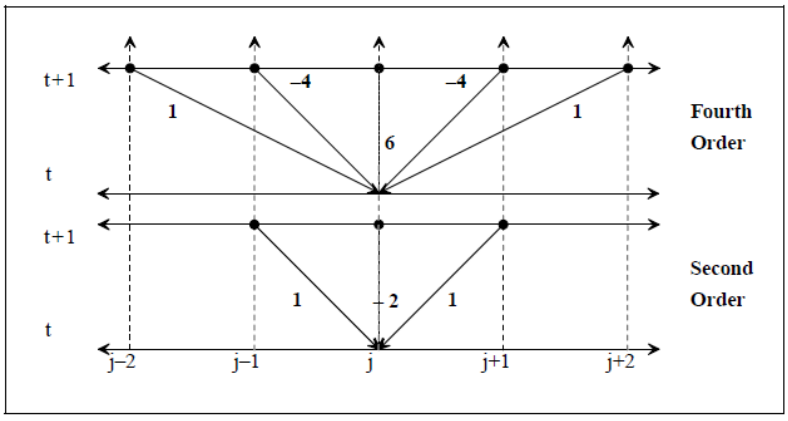


Figure 4.4 Integration using finite difference for the smoothness and stiffness parameter of the ACM Interface

The above process derives the smoothness and stiffness parameters which need to be calculated for every snake point on to the ACM interface. Due to the use of finite integrals, it is not possible to derive their values directly from a closed form equation, instead, their values are derived by multiple applications of the smoothness and stiffness parameters over the set of control points, with each application or iteration of the process conduction only very small changes over a small step (Δt). This allows the interface to evolve over time and fit the path defined by the control points, producing optimal fitting spline representing the skeleton of the underlying root. At each iteration the new coordinates for each snake point are calculated using a simple penta-diagonal matrix defined as:

$$\begin{bmatrix} \mathbf{1} & & & & & & \\ \boldsymbol{p} & \boldsymbol{r} & \boldsymbol{p} & & & & \\ \boldsymbol{b} & \boldsymbol{q} & \boldsymbol{r} & \boldsymbol{q} & \boldsymbol{b} & & \\ & & \ddots & & & & \\ & \boldsymbol{b} & \boldsymbol{q} & \boldsymbol{r} & \boldsymbol{q} & \boldsymbol{b} & \\ & & & \boldsymbol{q} & \boldsymbol{r} & \boldsymbol{q} & \\ & & & & & \mathbf{1} & \end{bmatrix} \begin{bmatrix} \boldsymbol{u_0^t} \\ \boldsymbol{u_1^t} \\ \boldsymbol{u_2^t} \\ \vdots \\ \boldsymbol{u_{n-2}^t} \\ \boldsymbol{u_{n-1}^t} \\ \boldsymbol{u_n^t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u_0^{t+1}} \\ \boldsymbol{u_1^{t+1}} \\ \boldsymbol{u_2^{t+1}} \\ \vdots \\ \boldsymbol{u_{n-2}^{t+1}} \\ \boldsymbol{u_{n-1}^{t+1}} \\ \boldsymbol{u_n^{t+1}} \end{bmatrix}$$

Equation 4.3 Matrix form for calculating successive iterations of the ACM model

where the values of a, b p, r and q are derived from the smoothness and stiffness parameters and the values of u^t and u^{t+1} are the old and new coordinates of the corresponding snake points. The above matrix can be computed in $O(n^2)$ time with respect to the number of snake points. This would require significant computational resources, given that many hundreds of iterations might be needed before the snake interface reaches equilibrium. The Symmetric Pentadiagonal matrix algorithm solver can be used instead to compute the above matrix in $O(n)$ [54].

The external force in the above equation is derived from a force field, generated from the spatial distribution of control points extracted from the distance map of multi-directional sweeps in 3D volume stacks using the inverse square law correlation.

$$F = \frac{1}{r^2}$$

Equation 4.4 Force is inversely proportionated to the square of the distance of the source. F is the calculated Force and r is the distance between a control point and a point within the image stack.

The algorithm calculating the force field of the image stack uses an inductive loop in which every voxel location within the stack is visited once and, for each voxel, the distance to all other control points as output by the multi-sweep control point generation step is calculated and the weighted sum of the inverse square law is assigned as the value of the force for that point. Note that it is not necessary to create a look-up table of the entire volume stack, as that would require a significant amount of processing time and a lot of available memory. It is only sufficient to calculate the force field at every coordinate ever occupied by a snake point as the ACM interface evolves. The force field can thus be computed in-situ for every coordinate when it is needed. It should also be noted that for very small values of *r*, in order to avoid division by 0 or the production of very large values of the force at that point resulting in uncontrollable oscillation of the ACM interface, the value of *r* is rounded up to 1 voxel unit.

Figure 4.5 shows an example of an ACM model running over a 2D force field. The control points of the force field are chosen at random to describe somewhat a wiggly curve. Notice that the control points encode no information regarding the ordering of the points within the imaginary curve. This demonstrates that the Active Contour is able to recover the underlying model even in cases where there is no meta-information related to the ordering of control points as would be required by non-parametric or parametric curve-fitting or interpolation.

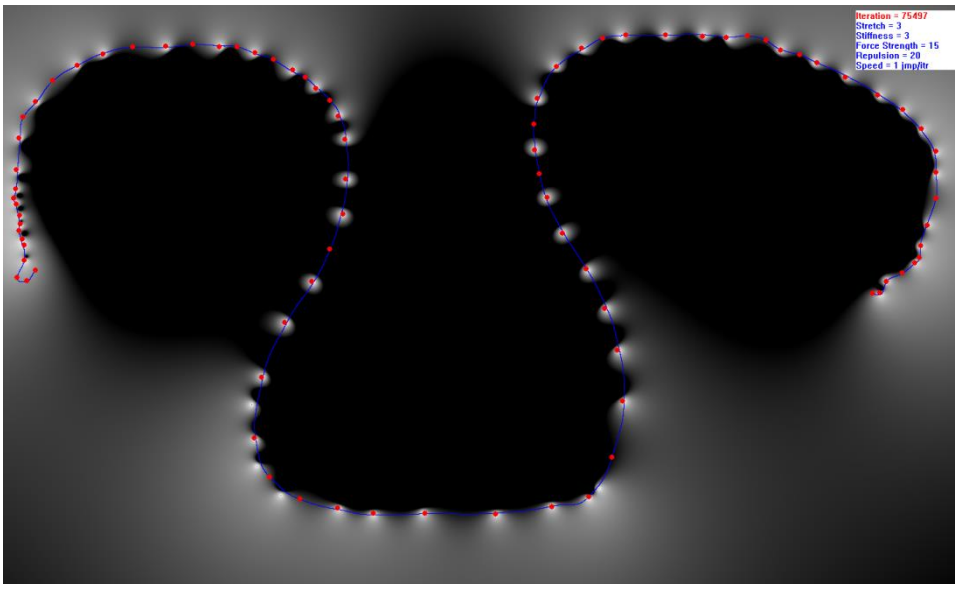


Figure 4.5 2D Active Contour Model over an inverse square law vector force field

The open-ended Active Contour Model interface used in RooTh is initialised using two anchoring points per root, one at the root tip and the other at a corresponding root branching point (as opposed to a single anchoring point initialisation at root tips alone as in the curve-fitting approach). No root branch detection algorithm is developed, however. Unlike the curve-fitting method, which starts with the bare root segmented output, the ACM method is initialised with the curve-fitting fractalised output (which is standalone) allowing it to locate branching points and root tip points for every root. The ACM interface is initialised as a straight line and is then allowed to evolve over a series of iterations in order to match the control points as illustrated in Figure 4.6:

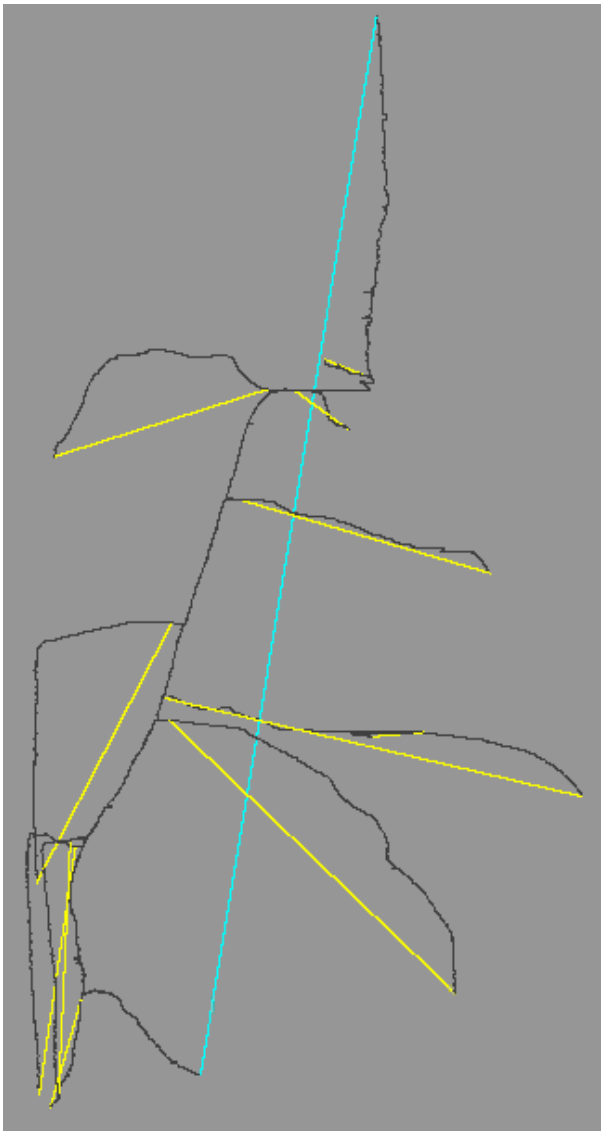


Figure 4.6 Initialisation of the Active Contour open-ended interface as straight lines connecting root branches with their corresponding root tips (black – curve fitted skeleton; cyan and yellow – primary and lateral roots initialised ACM interface)

The ACM interface is initialised as a straight line in order to minimise the occurrence of entangled root sections. In cases where there is a high concentration of snake points, commonly around high tortuosity sections (those with a large degree of bending) the ACM interface can become entangled. As a result, the reported skeleton will end up representing the same root section multiple times. In other words the uniqueness property of the root skeleton will be violated: every individual section of a root must be represented by a single skeleton thread. Due to the same section of root been represented multiple times, any subsequent trait evaluation will result in incorrect or overestimated measurements (for example, measuring the length of a root using a non-unique, entangled skeleton where the same segment of root is represented multiple time will result in overestimating the total root length). For this reason, ACM skeleton entanglement must be avoided all cost in order to produce high quality and accurate root traits measurements.

In order to deal with the problem of ACM skeleton root entanglement a form of simulated annealing is used where the values of the smoothness and stiffness are artificially and repeatedly incremented and decremented in sequential fashion letting the ACM to run for a few hundred iterations in between and thus allowing it to straighten up and smooth itself out (self-repair). The process of simulated annealing is illustrated in Figure 4.7:

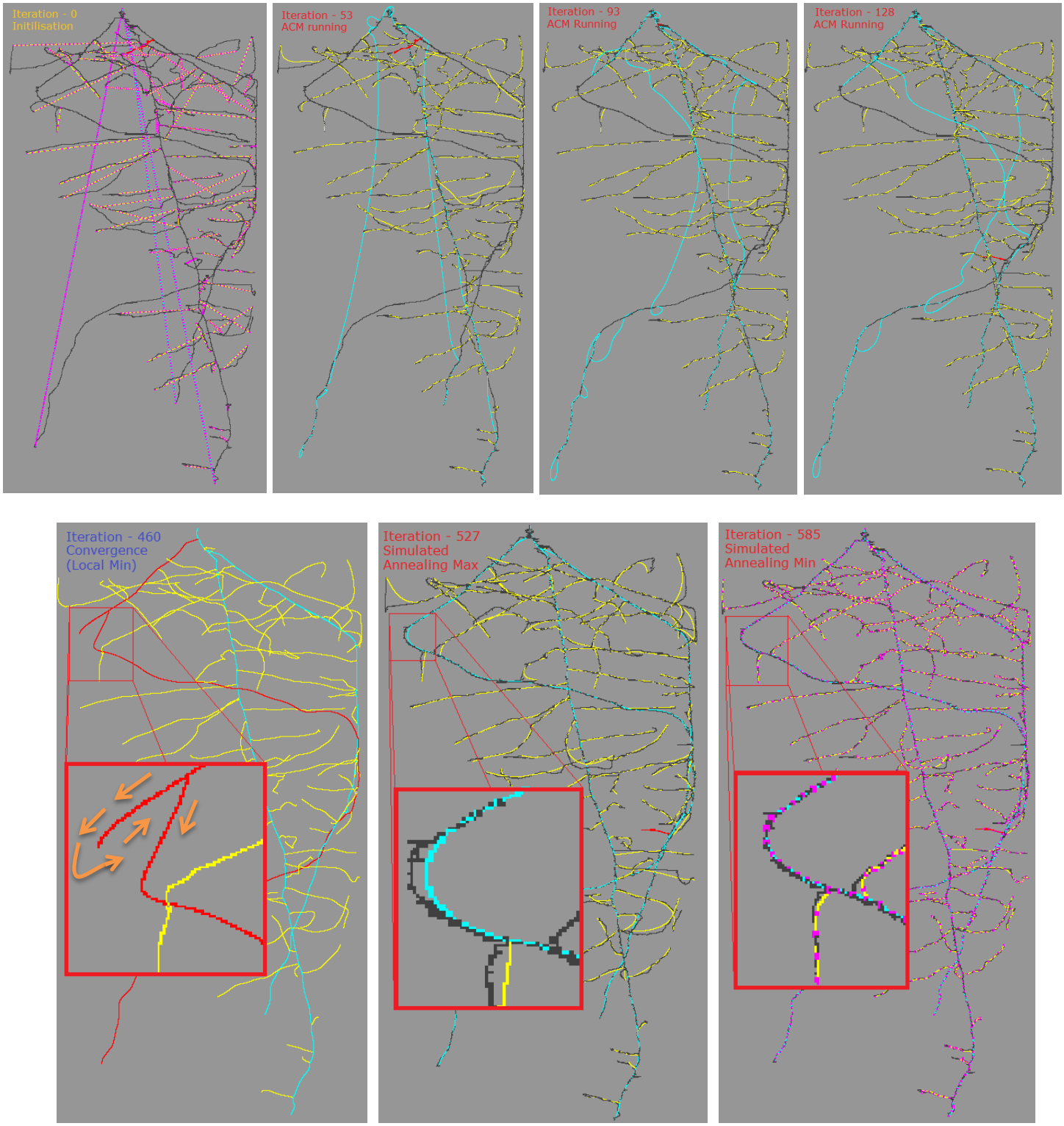


Figure 4.7 Simulated Annealing of Active Contour Model – Iterations 0 to128: running of ACM interface. At iteration 460 the snakes converged to a settled stated and only slightly oscillate around the control points of the root. Entanglement occurs in one of the root bends as illustrated in the red squares (the path of the entangled ACM skeleton section is visualised using the orange arrows resulting in a section of the root being described multiple times i.e. the uniqueness property of the root skeleton is violated). Iteration 527 is the first step of simulated annealing, increasing the stretch and stiffness of the ACM interface resulting in stretching the snakes and untangling the incorrectly skeletonised root section, however now many of the snakes cut corners around bends resulting in an incorrect skeletonisation in its own terms. The last stages of the simulated annealing at iteration 585 brings the values of stretch and stiffness back to their original allowing the ACM to fall back into the roots, this time however without the entangled sections resulting in correctly skeletonised root system architecture.

Simulated Annealing is more commonly deployed in the field of computational complexity where it is used to devise probabilistic polynomial algorithms with the aim of finding near-optimal solutions for problems which otherwise lie within NP and are thus not directly solvable on a deterministic machine. The ACM cannot be solved analytically using a closed form solution [53], however, it can be solved through optimisation using gradient descent. As such, it bears similarities with some other optimisation problems in that multiple concurrent solutions exist, some of which are better than others. The simulated annealing technique can be used to find a near-optimal solution (one lacking any entangled sections). According to [55] simulated annealing is particularly well suited for problems where:

1. *Solution quality can be expressed as a value/number*
2. *Initial solution is straight forward*
3. *There exist small adjustment steps between alternative solutions*
4. *Every two solutions can easily be derived from one another using combination of those small adjustments*

The above list suits the ACM method quite well where 1. The quality of the ACM skeleton can easily be quantified based on how well it fits the control points, how smooth it is and whether any entangled sections are present; 2. The initial ACM interface is a straight line computed by the linear equation $y=ax+b$; 3. The application of the ACM penta-diagonal matrix converts between alternative solutions for the ACM skeleton and 4. a succession of iterations leads to local minimum for the energy minimisation function converging the ACM.

Initialising the ACM interface as a straight line allows it to evolve its shape to match that of the underlying root it is intended to represent by starting from the lowest possible tortuosity point in the ACM energy minimisation function [53] (a straight line for which both the first and second derivatives are 0) and evolving towards a higher order curved spline (for which the first and second derivatives will both be larger than 0). A straight line ACM will correspond to a very high energy point in the energy minimisation function as snake points will be far away from the control points they are attracted to (Figure 4.8). Having a high energy initialisation configuration represents a good initialisation strategy as it allows the ACM interface to evolve towards the desired low-energy non-entangled unique skeleton configuration without being stuck in an unusually large local maximum during the simulated annealing epochs.

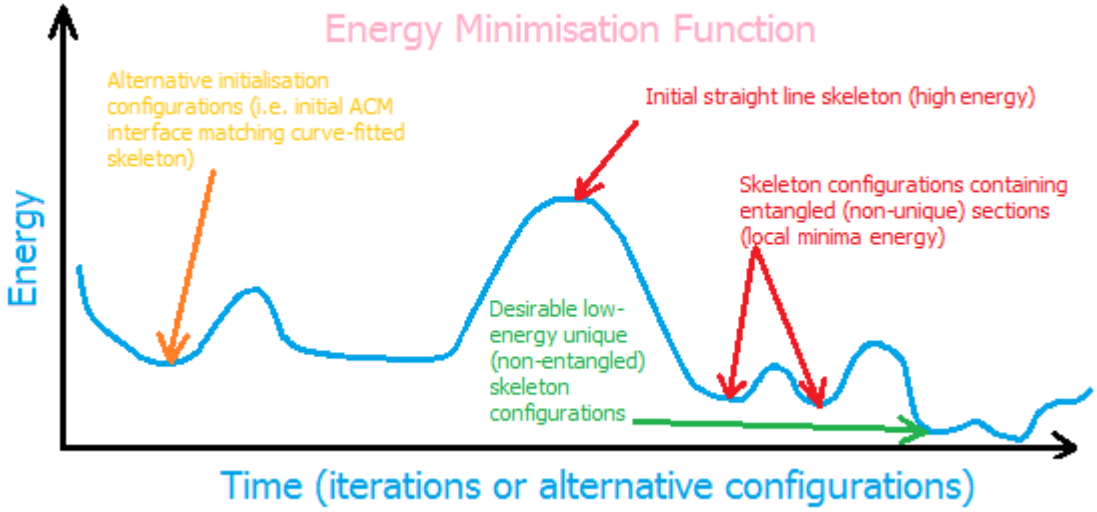


Figure 4.8 Illustration of Energy Minimisation Function of an Active Contour Model Interface. The y-axis represents the cumulative energy of all snake points of the ACM. The x-axis represents all possible configurations of the ACM interface (3D coordinates of snake points in relation to 3D coordinates of control points). Entangled skeleton sections (pointed out by an orange arrow in the example illustration above) contain very sharp curve bends (at 360 °) which will have very high values for the second derivative and thus add weight to the energy minimisation function preventing such configurations from reaching energy levels as low as those of genuine unique skeleton ACM configurations (pointed out by a green arrow). An entangled section of a root will further stretch out the entire ACM interface as it will take more distance to cover the root length compared to a non-entangled unique curve which will add further weight to the energy.

An alternative initialisation configuration for the ACM interface could for example be the output skeleton of the non-parametric curve-fitting method. This however will not be an ideal initialisation strategy as the curve-fitting method has been shown to contain highly entangled non-unique representations of certain sections of the root system (especially around the plant kernel or plant seed) as depicted in Figure 4.9 and as such will represent local minima in the energy minimisation function, potentially preventing the Simulated Annealing algorithm from evolving the ACM interface to the desired low-energy state of a singular unique skeleton representation. Another way to rationalise the above argument is that if the ACM interface was initialised using the outline of the curve-fitted skeleton output, the ACM contour will inherit all bad properties of the curve-fitted method, including non-unique descriptions of sections of roots where the same root segment is represented multiple times and due to the iterative nature of the Simulated Annealing algorithm, it may have limited capacity to mitigate or eliminate such artefacts in the final ACM skeleton output.

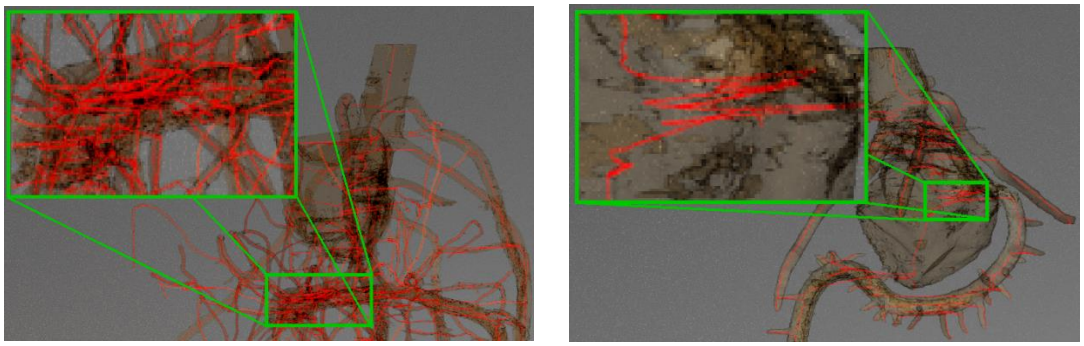


Figure 4.9 Highly-entangled sections of curve-fitted skeleton of two maize plants (left and right) demonstrating that the curve-fitted skeleton output is not suitable for initialisation strategy of the ACM interface as the goal of the Simulated Annealing algorithm is to eliminate such sections and yet the curve-fitting output will be introducing them to begin with.

RooTh uses an interactive GUI approach to let the user determine how well the skeleton fits the control points and deploys simulated annealing when needed by temporarily increasing and then decreasing the values of smoothness and stiffness whilst the ACM is running. Since the ACM simulated annealing method only requires a few annealing iterations to converge, often just a simple low-high-low sequence of steps is sufficient to eliminate any entangled sections. For that reason, an automated method can be used where the number of ACM iterations between each annealing step is pre-determined. Alternatively, a sum of minimum square distances can be used to automatically determine when the ACM interface converges and thus alternate between successive annealing iterations until that happens. The sum of square distances or tortuosity τ (Equation 4.5) can then be used to detect the presence of entangled sections and thus determine the number of annealing steps required to find a near-optimal solution.

$$\tau = \frac{N-1}{L} \cdot \sum_{i=1}^{N} \left(\frac{L_i}{S_i} - 1 \right)$$

Equation 4.5 Equation of Tortuosity – the curvature is divided into N parts and for each and for each part, the length of the curve L is used to calculate the local tortuosity and then all is summed up and averaged.

4.4 Spline Fitting and Trait Extraction

The snake points of the Active Contour Model need to be spaced out evenly along the length of the root and span the control points. In order to ensure both the stability of the ACM interface and that it reaches equilibrium after reasonable number of (a few hundred) iterations it is necessary for there to be multiple control points per snake point. A ratio of approximately 1:40 snake to control points has been found effective. This means that the skeleton itself cannot be sufficiently described by the snake points alone as there will be large gaps between neighbouring snake points. Instead, piecewise cubic spline interpolation is used to define a continuous curve between different sections of snake point intervals [56]. The splines spanning between every two adjacent snake points are each fully determined by the 4 parameters of the cubic curve equation:

$$y = ax^3 + bx^2 + cx + d$$

Equation 4.6 Cubic curve parametric canonical form

If there are *m* snake points there are *m-1* piecewise sections between those snake points and therefore *m-1* piecewise curves to be defined. The cubic curves are defined such that the first and second derivatives at each end of the corresponding adjacent curves are equal, ensuring a smooth transition from one curve to the next one at the joining snake points. These constraints, along with the requirement that the derivatives are equal to 0 at both ends of the ACM interface (as it is open ended) allows one to construct an *m x m* tri-diagonal matrix of linear equations, the solution of which gives the parameters of the piecewise cubic splines for the root skeleton.

$$\begin{bmatrix} 2 & 1 & & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & . & . & . & & \\ & & & 1 & 4 & 1 & \\ & & & & 1 & 2 & \end{bmatrix} \begin{bmatrix} D_0 \\ D_1 \\ D_2 \\ \cdots \\ D_{m-1} \\ D_m \end{bmatrix} = \begin{bmatrix} 3(y_1 - y_0) \\ 3(y_2 - y_1) \\ 3(y_3 - y_2) \\ \cdots \\ 3(y_m - y_{m-2}) \\ 3(y_m - y_{m-1}) \end{bmatrix}$$

Equation 4.7 Matrix form for calculating the cubic spline interpolation of the snake points

The piecewise splines are only capable of interpolating the root skeleton curve between the root branching points and the root tip. However, the root branching point as determined by the curve-fitted skeleton method In Chapter 3 doesn't always ideally match or connect with the parent root. In order to define a complete skeleton representation, it is therefore

desirable to extrapolate the piecewise curves from the emerging root branching root to the nearest parent root snake point. Since the distances are fairly small compared to the length of the branches making up a typical root system, it is sufficient to extrapolate using simple linear interpolation between the branching point and nearest snake point of the parent root connecting straight line as depicted in Figure 4.10 below.

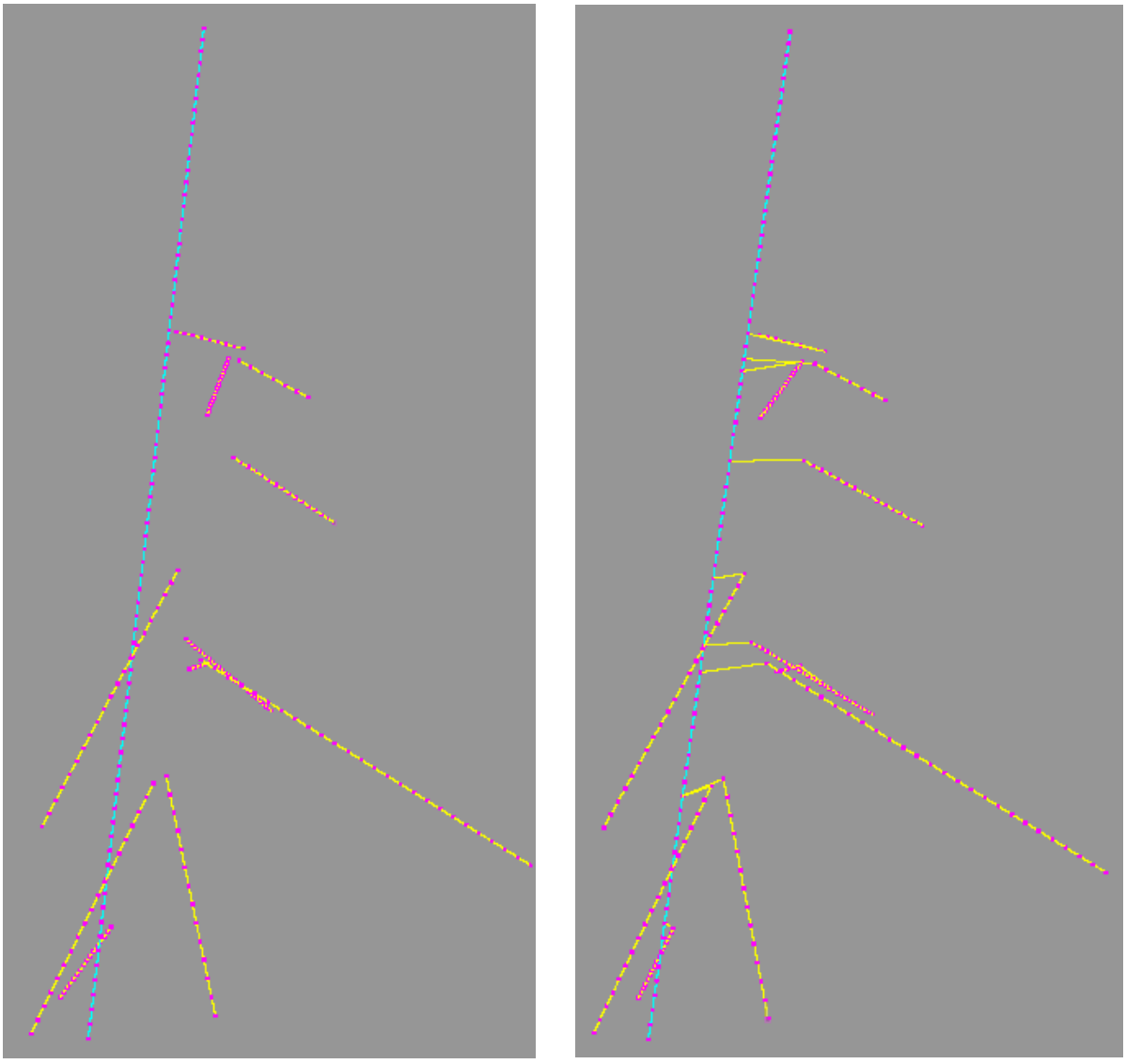


Figure 4.10 Illustrating extrapolating connecting lines (purple dots – snake points). Initial ACM interface is used in order to highlight the straight lines connecting the base of each lateral root (in yellow) with the primary root (in cyan) which otherwise they are too small to easily spot. The figure on the left lacks the connecting line and the figure on the right includes them to demonstrate their presence

Once the ACM skeleton has been computed, various traits can be derived from the piecewise curves representing the root system model. Root length can be calculated using finite integration by summing up the straight line distances between very small intervals

along the piecewise curves, and this can be computed very fast and efficiently on a modern CPU. Root width can be estimated by averaging out the distance between various points evenly spaced out along the length of the root skeleton and the nearest point on the root wall (twice that measurement in order to derive the diameter). Since the ACM skeleton has the property of being equidistant for all roots, including plagiotropic roots, it becomes possible to evaluate the local root width at every point along the length of the skeleton and output that to the RSML format description. Such high-dimensional root traits data could be very useful for running root simulation and mathematical models; whereas the average root width per root is more suited for QTL and other forms of more traditional phenotyping analysis such as QTL.

4.5 Results and Discussion

The segmented output of the data produced by the validation experiment presented in Chapter 3 was re-evaluated using the Active Contour method of skeletonisation proposed here. The results are shown in Figure 4.11.

As it can be seen, the ACM skeletonisation method has done a much better job at encoding the underlying root, providing a smooth and equidistant skeleton. If, for example, a closer look is taken into the kernel of the maize plant (which is problematic for the curve-fitting method as noted in Chapter 3) it can be observed that the Active Contour has largely been able to omit the high-variance spatial distribution of the control points caused by the large and irregular shaped maize seed. It has produced a smooth, curved skeleton path passing through it and smoothly transitioning into the primary root. It can further be observed upon close inspection of some horizontal sections of roots depicted in Figure 4.12 that when compared with the curve-fitting method, the ACM skeleton is equidistant from the root wall in all directions (horizontal and vertical) and is able to produce a smooth and complete skeleton irrespective of the direction of growth of the root.

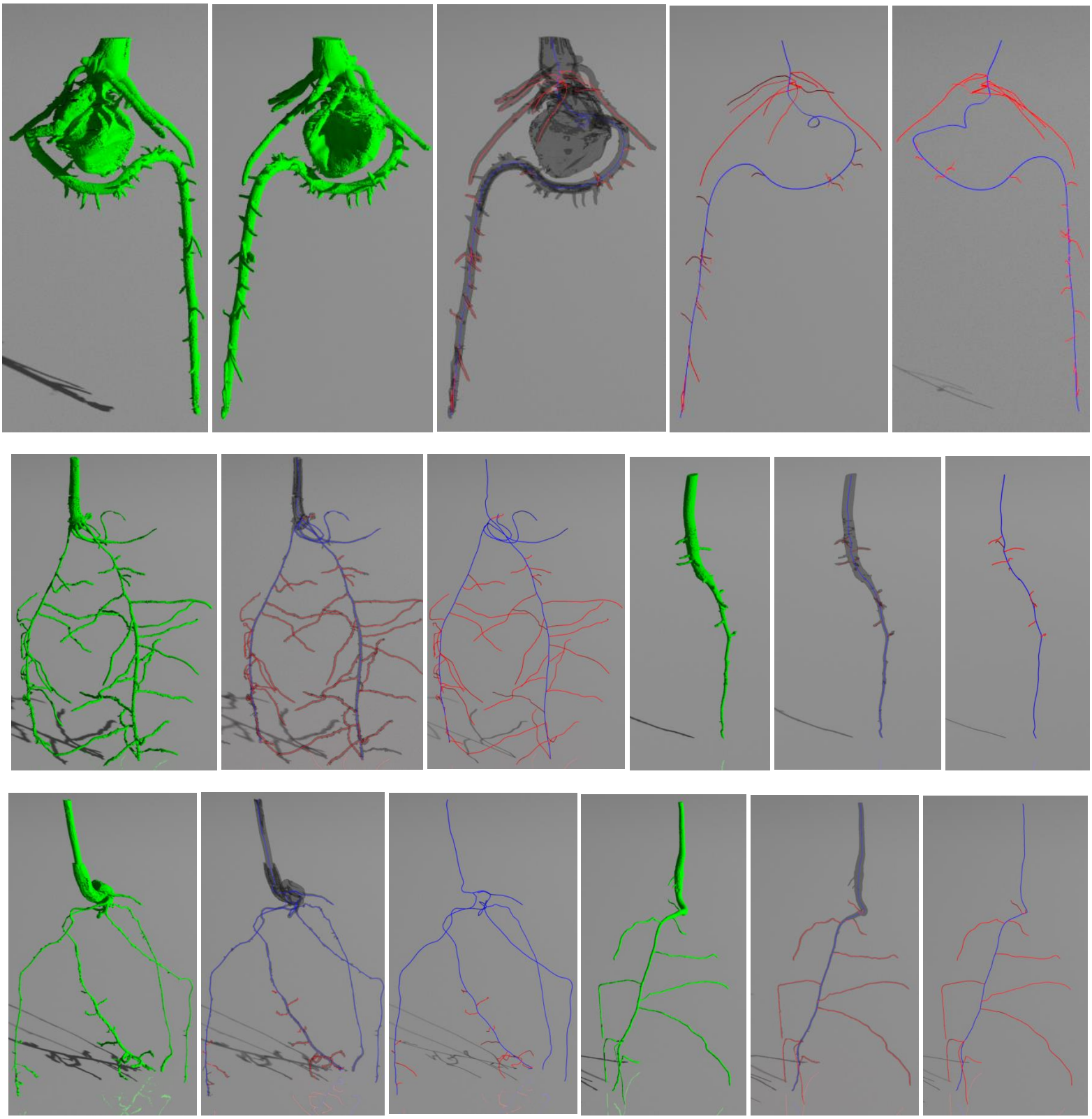


Figure 4.11 10 days old wheat plant scanned with the medium size scanner, segmented with RooTrak and skeletonised with RooTh. From top to bottom row – maize clay-loam, wheat sandy-loam, wheat clay-loam, tomato sandy-loam, tomato clay-loam. From left to right column – RooTrak segmented output, segmented output overlaid on top of ACM skeleton and ACM skeleton alone

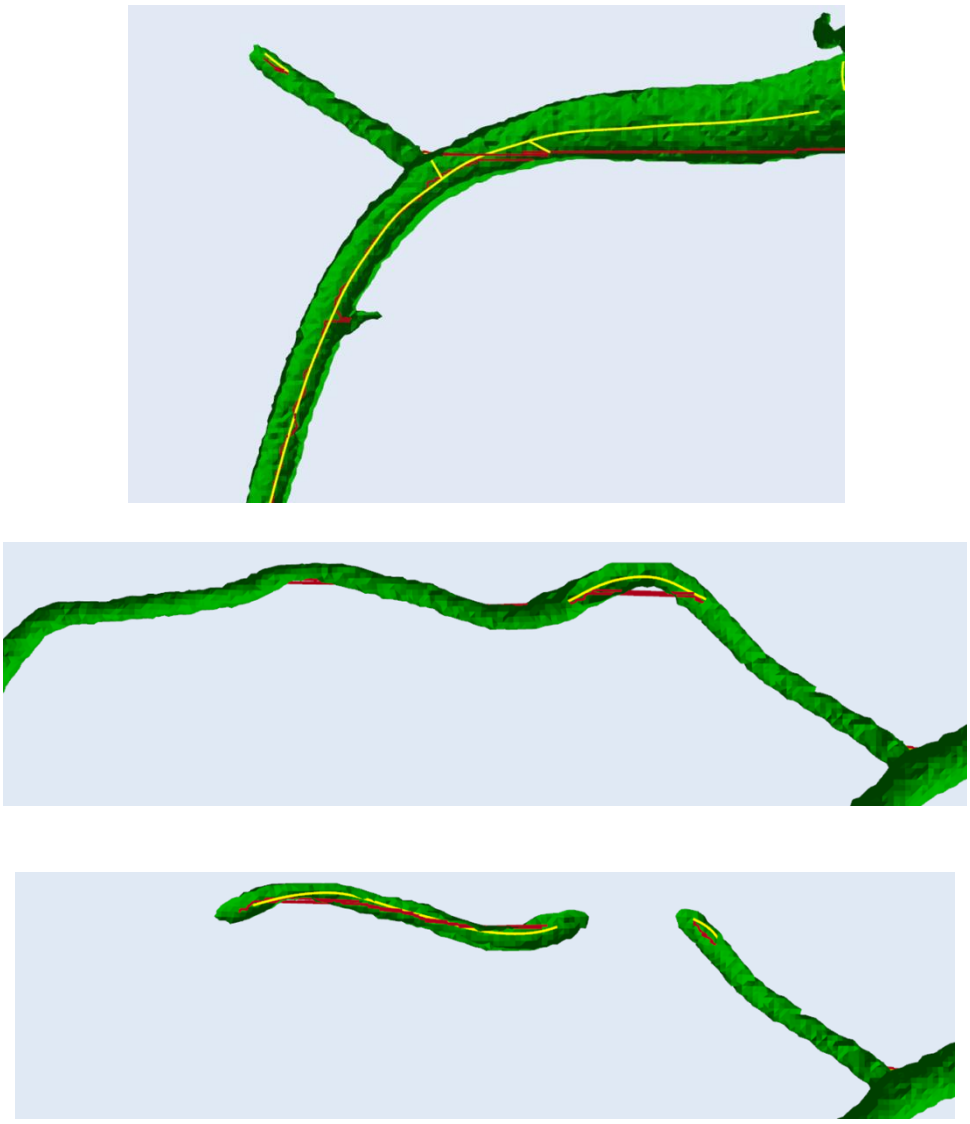


Figure 4.12 Comparison between curve-fitting and ACM skeleton illustrating the shortfalls of the curve-fitting method to handle horizontal sections of plagiotropic roots. The red highlighted skeleton is the output of the curve-fitting method and it fails to accurately follow the curves of the root in horizontal sections unlike the ACM skeleton highlighted in yellow which does a much better job at encoding horizontal sections of roots.

In a similar fashion to the method used to evaluate the curve-fitted skeleton in Chapter 3, the ACM skeleton output was compared against manually skeletonised ground truth points of individually selected root-cross sections and the Euclidian distance was used to estimate how accurate they are. Cross-sections were chosen such that they are roughly round in shape. Where roots are horizontal with respect to the Z axis of the volumetric stack, root cross-sections would appear elongated cutting across the length of the root. As the ACM skeleton is globally optimal and 3D spatially aware, unlike the 2D curve-fitted skeleton, it will weave in and out of the cross section resulting in multiple visible points making it hard to choose which one to compare with the ground truth point as illustrated in Figure 4.13.

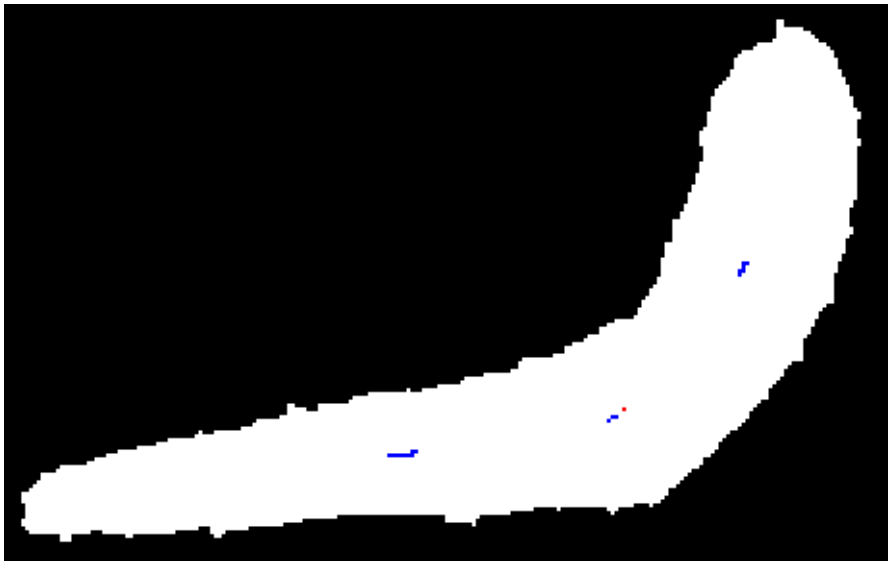


Figure 4.13 Comparison between ACM (blue lines) and curve-fitted skeleton (red dot) for a cross section of a horizontally growing root section of a root. It can be seen that the spatially aware globally optimum ACM skeleton will weave in and out of view, resulting in multiple readings to be taken if the data is to be used for correlation with a manually selected ground truth point.

In order to solve this problem, for horizontal sections of the root system, side (XZ and YZ front-back and left-right) views of the root systems were examined instead using Fiji's built-in Orthogonal Viewer ensuring that all measured root cross-sections are relatively round in shape and approximately perpendicular to the direction of growth of the measured root section. The results are shown in Table 4.7

Plant	Tomato	Tomato	Wheat	Wheat	Maize
Difference fitted curve	3.92	3.12	4.17	2.92	12.54
Difference ACM	2.23	3.45	3.42	1.89	11.20

Table 4.8 Estimate of the difference between the ground truth of manually selected skeleton points and the corresponding curve-fitted and ACM points for a sample of 15 root cross sections per plant for 5 validation experiment plants.

The table above shows that the ACM skeletonisation method performs consistently better than the curve-fitted method when evaluated against ground truth. This further backs the evidence seen from the validation experiment suggesting that overall the ACM approach produces a more accurate description of the underlying root system being skeletonised. The slightly poorer performance for the ACM method compared to the curve-fitted skeleton for the second plant could be justified if it is taken into account that this is the very last root

system in Figure 4.11 and as it can be seen from the illustration, it consists mostly of one vertical root with virtually no lateral roots. All manual measurements therefore had to be taken from the primary root, which is a downward growing root. The way the manual estimate of the ground truth was made would ordinarily put the skeleton ground truth point near the centre of the root-cross section and as such, for downward growing roots it would favour the curve-fitted skeleton which is locally optimal for each root cross-section instead of the ACM skeleton which is globally optimal. Due to the lack of horizontal root-cross sections being included in the manual skeletonisation dataset where we would expect the ACM skeletonisation algorithm to naturally outperform the curve-fitted method as discussed in Chapter 3 Conclusions, this could explain the results for this particular plant.

The very promising initial results from the validation experiment output of the ACM skeletonisation method meant that RooTh could be deployed for the analysis of a real-world biological experiment. The skeleton outputs in Figure 4.14 show the results of an experiment carried at the greenhouse at the Hounsfield Facility at the University's Sutton Bonington Campus. The purpose of the experiment was to study the Nitrogen Uptake efficiency of certain varieties of wheat plants such as Rialto and Savannah and involved 80 replicates of soil columns planted with germinated wheat seeds grown in a controlled environment and subsequently scanned with the medium-size scanner at 54μm resolution using high-quality settings (hour long scans). The main focus of the experiment was to quantify the number and the traits (length, width angles) of the plagiotropic lateral roots which made the ACM approach very well suited for this study. The reconstructed CT volumes were subsequently segmented using manual segmentation provided by VG Studio Max. This posed further challenges for the skeletonisation method due to the poor quality segmented output; however, the majority of complicated cases some of which are illustrated in the samples bellow were largely dealt with by the ACM method. An additional mandatory pre-processing step of morphological smoothing (opening followed by closing) for all scans was applied in order to alleviate to a certain extent the effect of the poor manual segmentation, to discard (absorb) some of the artefacts of the region growing technique and also to ensure that all sections of the segmented root system are fully connected (in some cases it was found that the region-growing segmentation method or the poor guidance of the user doing the manual segmentation would leave small gaps severing

roots into disconnected sub-sections). A side effect of the application of morphological operators was increased instances of touching or crossing roots for which reason a more extensive manual annotation of the fractal structure of the skeleton was necessary before running the ACM method in order to ensure that the initialised ACM interface is correctly hung between the branch point and its corresponding root tip for every root and to ensure the structure and integrity of the outputted root system skeleton.

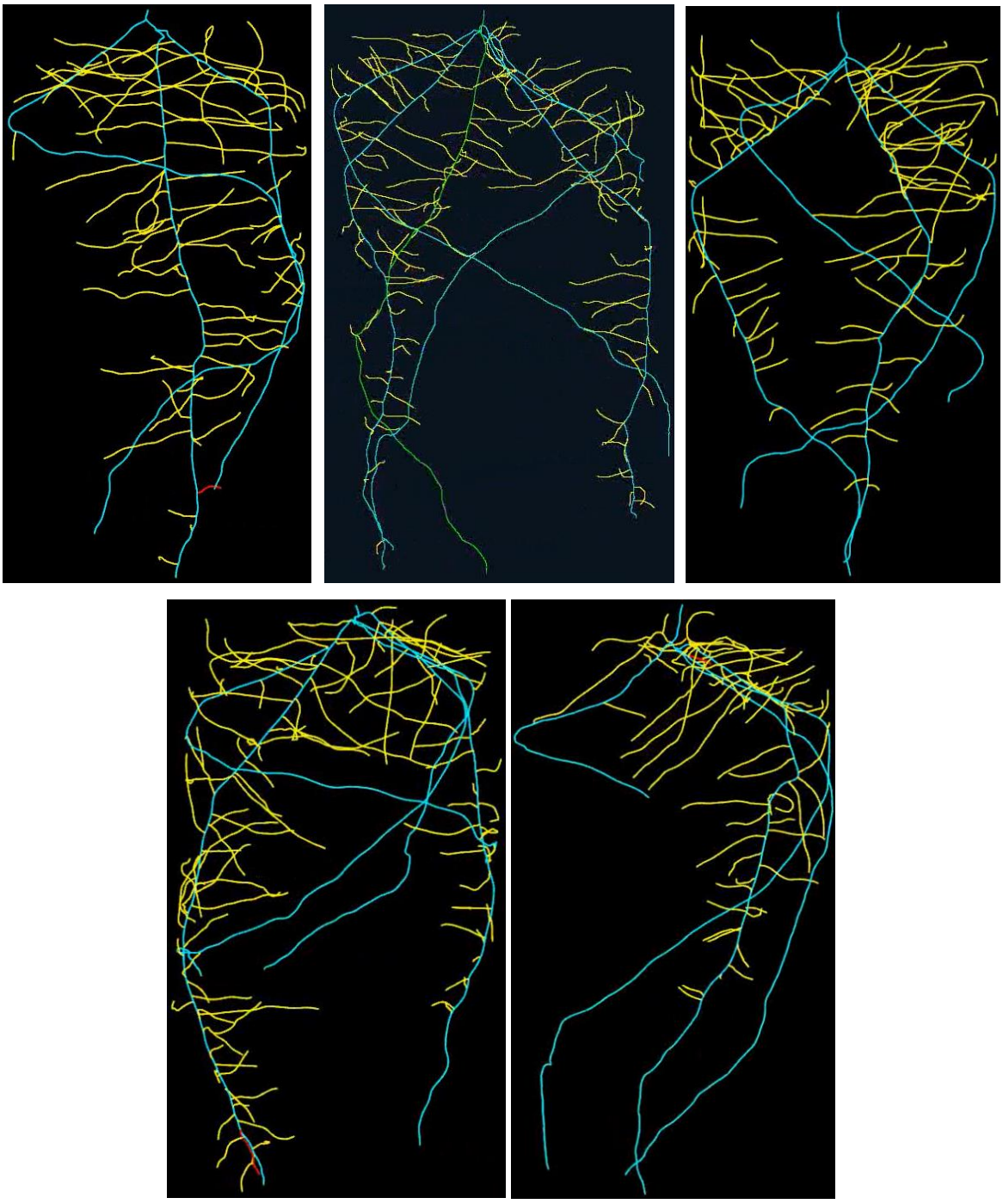


Figure 4.14 Sample of skeleton from the Nitrogen Uptake Efficiency experiment of wheat plants

Figure 4.15 presents a summary of some traits of interest quantified from the skeletonised data, such as root length. Some notable differences have been picked up by the ACM method between the low (red) and high (blue) nitrogen treatment for certain varieties/mutants plants.

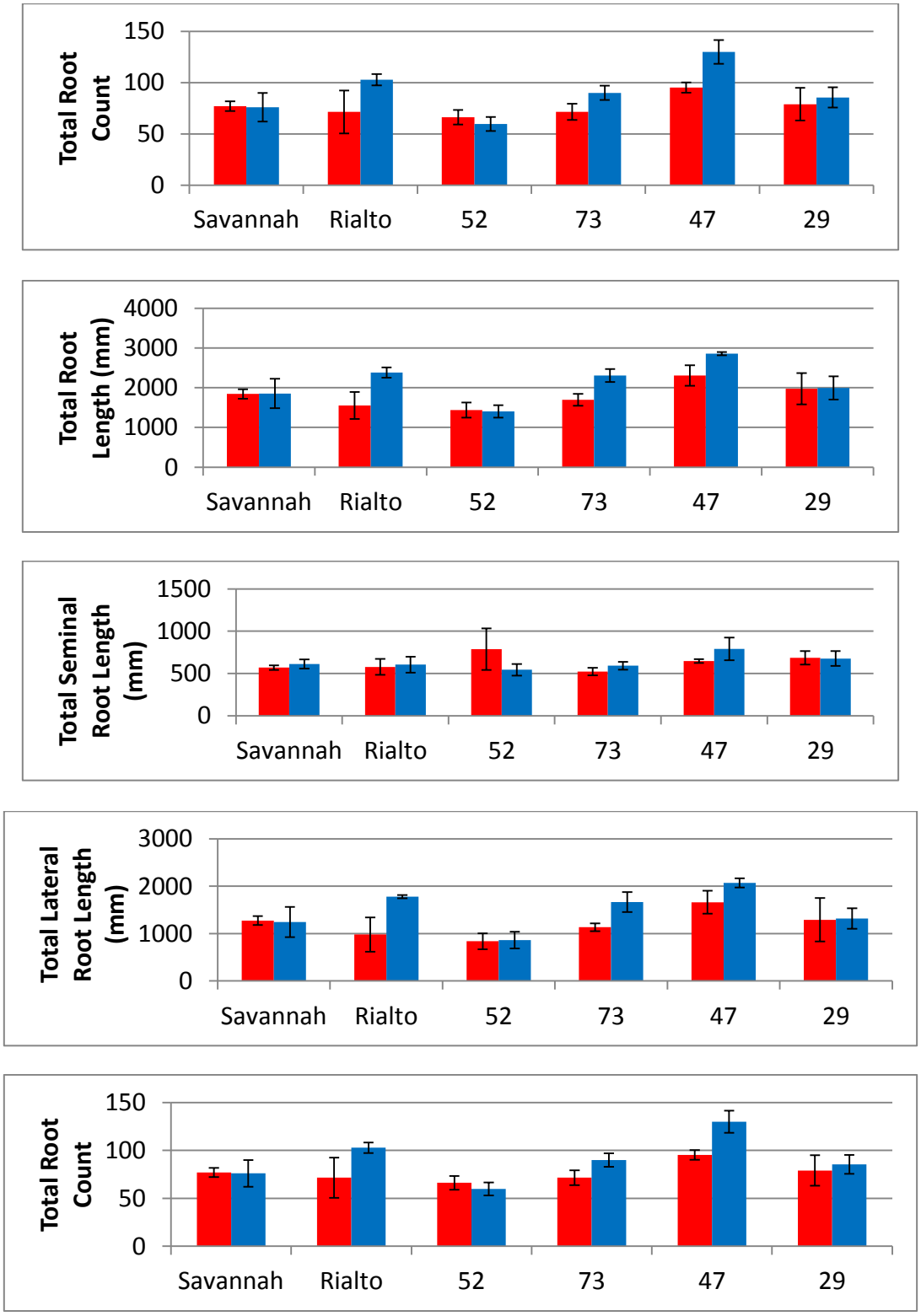


Figure 4.15 Biological Experiment of Anonymous Data showing the apparent differences between different treatments of plants (red – low nitrogen vs blue – high nitrogen)

The biological experiment suggests that higher nitrogen treatment of the soil allows the root system to develop more rapidly and penetrate a larger volume of soil, which is evident if the total root length or the total count of roots is considered. Using RooTh to measure root system traits such as root length, root angles and root volume allows plant scientists to detect minute differences in the underlying structure of the root system of mutants and wild types. This in turn allows them to produce a description of the phenotype of a plant species being investigated and map it to the genome of the plant.

4.6 Conclusions

This chapter presents a novel method for the skeletonisation of plagiotropic roots based on running an Active Contour over a force field generated by control points extracted from multi-directional sweeps of the image stack. The goal is to provide a smooth and equidistant skeleton description of plagiotropic roots. The method has demonstrated resilience with dealing with complicated and poor quality segmented data such as that resulting from manual segmentation of image stacks. The method was evaluated using a validation experiment and due to satisfactory results the method was used in a real-world biological experiment involving the phenotyping of wheat roots.

Chapter 5.
Plant Root Registration Methods

5.1 Motivation

The work reported in Chapters 3 and 4 produces descriptions of root system architectures from individual, segmented, CT stacks. Motivation for the development of methods capable of dealing with time-ordered sequences of CT images comes from two sources:

1. Study of growth. CT is non-invasive; it can provide 3D data on the same root at different times

2. Access to image sequences may ease the single image analysis problem. For example, one high quality image might be captured along with several lower quality ones, but more quickly obtained, images that are hard to segment. Data extracted from the one might be used to guide analysis of the rest.

We will return to point 2 shortly.

In order to evaluate the dynamic changes that occur in the root system architecture (RSA) during the development cycle of a plant, it is necessary to bring RSA representations extracted at various time points under the same coordinate system, a process known as registration. Since the RSA itself is the most highly varying feature within the 3D image stack, we formulate the problem of registering the RSA from two time points as a two stage problem. The first stage is a global, approximate registration of the CT scans of the soil columns using soil particles as feature points to determine the parameters of the registration. The second stage is then to use the approximately registered, segmented image stacks and skeletonised roots to guide a local registration which would allow the detection and analysis of small changes in the root traits of the registered RSAs, like root elongation or root thickening.

Section 5.2 describes a method for a global approximate registration of CT soil columns using large soil particles naturally present in the soil pot as feature points and Section 5.3 discusses and evaluates the results of the method. Section 5.4 presents a method for the

local registration of RSML skeletonised root system architectures and the extraction of 4D time-related traits such as rate of root growth. Section 5.5 looks at alternative ways of segmenting individual time-points using knowledge gained from RooTrak or manually segmented parent time points.

5.2 Global Registration using Soil Features

5.2.1 Image registration

When registering two images of the same scene taken at different time points, one of the images is used as a reference and the other image is called a sensed image. The process of registering a sensed image with a reference one involves generating a third image called the registered image which is directly derived from and contains all the features of the sensed image but is encoded in the reference image's coordinate system. According to [57] there are four fundamental steps when registering two 2D or 3D images: feature selection, feature correspondence, transformation and resampling of the sensed image. As the name suggests, feature selection involves detecting recognisable features in both the reference and the sensed image which can then be used to compare the two images. The features used should describe in some way the characteristics of the various objects appearing in the scenes of both images. They could be of any nature and could fall into one of a number of categories – points, lines, edges, corners, regions etc. It is important that any feature selection algorithm produces a set of features which are reproducible in both the reference and the sensed image. That is, if an object appears in both the reference and the sensed image scenes, the feature selection algorithm will generate similar features describing that object. Only then is it possible to compare the features of the two images.

Once a set of features are detected in both the reference and sensed image it becomes necessary to pair features in the reference image with their corresponding ones in the sensed image (i.e. a feature describing an object in the reference image with the feature describing the same object in the sensed image). The list of paired features can then be used to determine the parameters of the registration. In the simplest case – a rigid transformation – the sensed image could be shifted on its x, y or z axes with respect to the sensed image, it could be rotated by a set amount of degrees around any one of its major

axis or it could be scaled up or down. All three transformations described above are more formally defined by the similarity transformation function:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & T_x \\ 0 & S_y & 0 & T_y \\ 0 & 0 & S_z & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\gamma & \sin\gamma & 0 & 0 \\ -\sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & cos\,\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Equation 5.1 Similarity Transformation Function. The first matrix performs scaling S and Translation T. The remaining three matrices apply rotation along the X, Y and Z axis respectively. All coordinates are expressed in the Quaternion System

Provided that the feature selection algorithm generates a suitable list of points, it is sufficient to determine a set of four non-coplanar corresponding point pairs in order to recover all nine parameters of the similarity transformation function. Once the parameters of the similarity transformation function are known (the angles of rotation α, β, γ around the x, y and z axis respectively, the scaling column vector S and the translation column vector T), the transformation function can be applied to any Cartesian coordinate in the reference image to obtain the corresponding coordinate in the sensed image. The value of the sensed image at that coordinate can then be sampled to determine the respective value of the registered image pixel/voxel. The process is repeated for all voxel coordinates in the referenced image until a complete registered image is reconstructed.

5.2.2 Registering Roots in Soil

Feature point detection in 3D image stacks of soil columns is particularly challenging due to the high levels of noise present in the scans, as well as the relatively high level of complexity of the soil structure. The problem is further complicated by the fact that small and medium-sized soil particles tend to move up and down and propagate through the soil column during different phases of the water irrigation cycle, meaning that feature points based on fine soil structures would not necessarily match with one another. The registration method reported here therefore employs feature points of large soil particles.

Since large soil particles tend to exhibit lesser movement due to their comparatively larger size, they are the best candidates when looking for corresponding features in the reference and sensed image stacks. Their centres of mass are well-defined point features and so are used to support registration. The problem of calculating the coordinates of these points reduces to the problem of segmenting individual soil particles from the soil column,

evaluating their sizes and then calculating the centres of mass (averaging the coordinates that make up the region) of the largest soil particles.

An effective and efficient method of segmenting out soil particles is the watershed algorithm applied over the gradient of the 3D image stack. The watershed approach [42] involves sequentially visiting each voxel of the image stack in increasing order of their value. When visited, a given voxel's immediate neighbours are examined and, if none have been visited yet, the algorithm marks the current voxel as a new region. If one or some of the voxel neighbours are already marked as visited and they all belong to the same region then the current voxel is added to that region. If a voxel sits between more than one regions then it becomes a region boundary, or watershed.

High absolute values of the image gradient (computed from the partial 1^{st} derivatives in x, y and z directions at every voxel) indicate rapid changes in the intensity values of the underlying voxels, which occur at the boundary between soil particles and the surrounding water/air/organic material. A small gradient indicates areas of the image where little change occurs, like inside a soil particle or an air pore. Since the watershed algorithm starts to form initial regions around voxels with the lowest values, this implies that when it is used in combination with the image gradient, region boundaries will be from predominantly around the edges of soil particles, which is the goal of the segmentation algorithm.

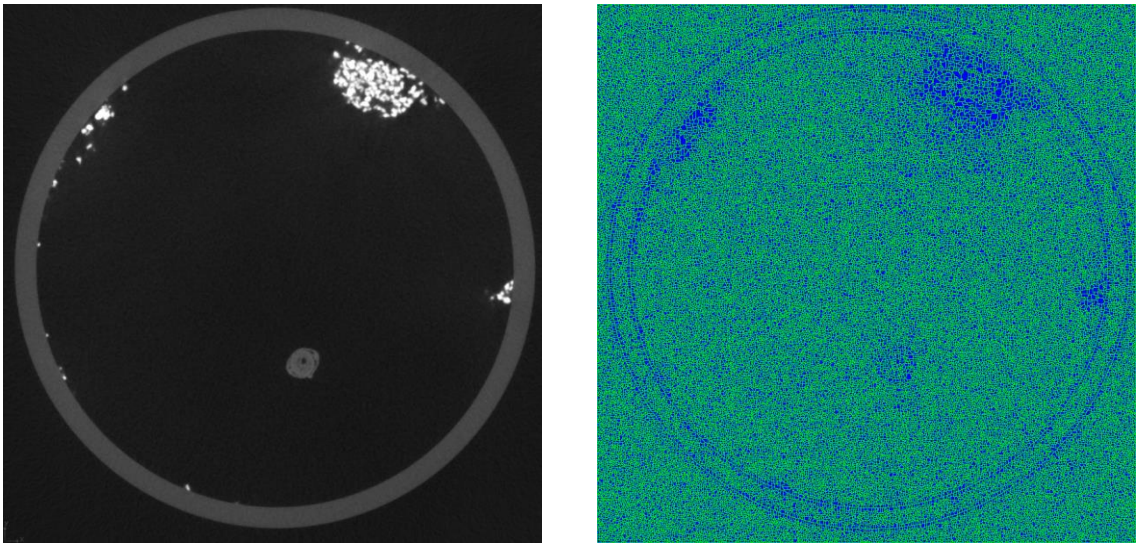


Figure 5.1 Watershed segmentation of the CT slice of a soil column

Unfortunately, the watershed approach tends to significantly over-segment images, especially those with high levels of noise as can be seen in Figure 5.1. There are different approaches to this problem which solve it with varying degrees of success. Common methods include pre-processing (e.g. Gaussian blurring) to minimise the effect of the noise and bi-level watershed where a threshold is used below which no new regions are formed even if a voxel sits on the boundary between multiple regions [58]. The solution proposed here is similar to the approach used in the split and merge method where a large number of regions are initially formed, then adjacent regions with similar properties are merged. In the case of the watershed segmentation, two adjacent regions are iteratively merged if the difference of the average intensities of their voxels is below a pre-set threshold. The value of the threshold determines the degree or merging with larger thresholds resulting in overall larger and smaller number of regions but more imprecise segmentation and smaller threshold producing more accurate segmentation at the expense of larger number of regions. After a small number of iterations, the resulting regions are significantly larger and much more accurately enclose large soil particles as it can be seen in Figure 5.2:

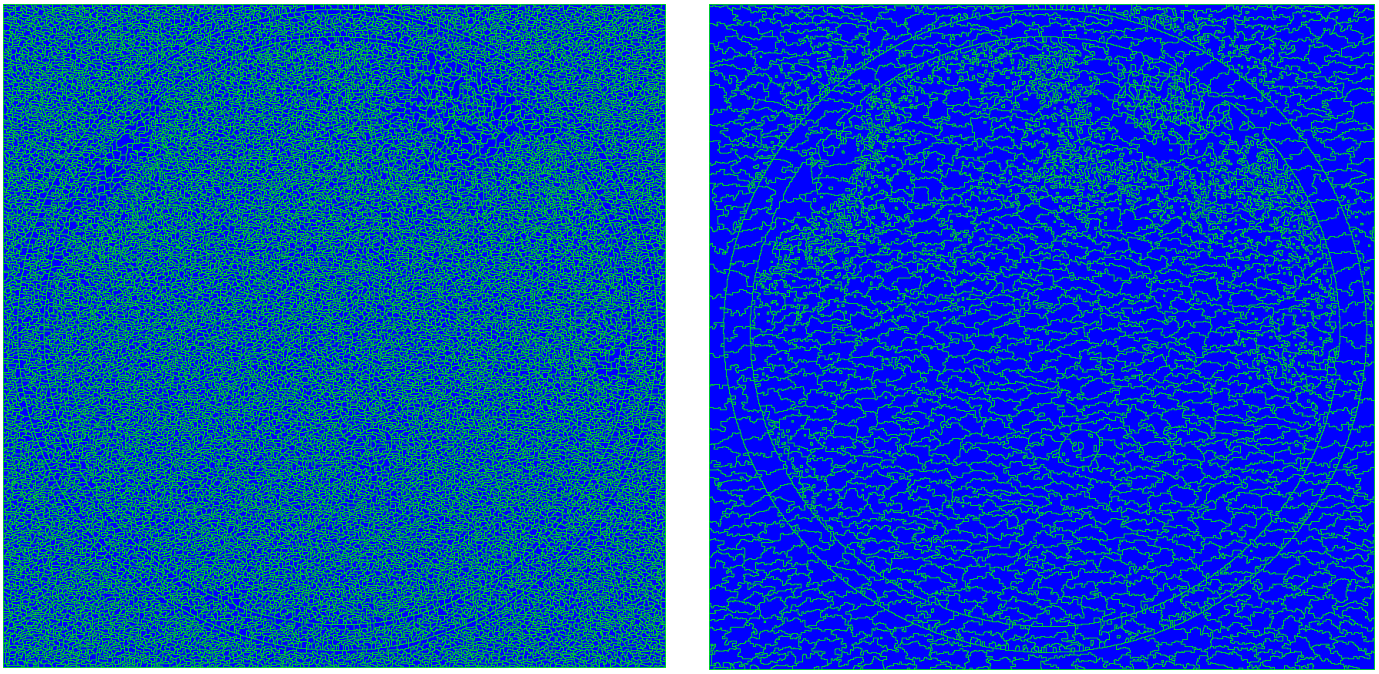


Figure 5.2 Split and merge algorithm used to reduce the number of over-segmented regions

The final step is selecting those regions likely to represent large soil particles. They are easily distinguished from other features in the soil column due to their high X-ray attenuation coefficient. Therefore, the average intensity value of their voxels is usually significantly

higher than that of other regions, meaning that a simple threshold classification could be used to detect them. Note that we only require features that can be matched between images, not an accurate semantic labelling of the image data.



Figure 5.3 Large Soil Particles in a soil column

The next step is to match soil particles appearing in the reference image with those in the sensed image. Due to various imperfections in the CT scans and errors in the segmentation process, many of the extracted soil particles are outliers – a feature point in the reference image might not be in the sensed one and vice versa. For instance, a threshold is used to select only the largest soil particles, however since soil particles come in various sizes, there would always be some who would be just above or just below the threshold. Further, the segmented description of a soil particle whose size is approximately that of the threshold could be the right size to be included in the feature set of one of the image stacks but not of the other. Instead of using feature points, whole regions could be used for feature selection based on their shape, volume etc. however there would still be the possibility that some might be present in the reference but not the sensed and vice versa. A feature correspondence method is therefore needed which takes into account the presence of outliers.

5.2.3 A Minimum Spanning Tree approach

The simplest approach as in many other cases is to use brute-force search. For every permutation of four non-coplanar points in the reference feature set, an equivalent permutation could be drawn from the sensed feature point set. The points in the set could then be paired with one another and they could be used to determine the transformation parameters. They can then be used to calculate the location of all feature points in the reference image with respect to that of the sensed image. If the selected four pairs all represent genuine feature points (i.e. identical soil particles appearing in both the reference and sensed images) then it could be expected that all other genuine feature points in the registered set will physically appear very close to the one in the sensed image. The same cannot be said for the outliers. A distance measure could be used to determine how well the registered points match the sensed ones and the permutation with the lowest score would be the one used to compute the registered image.

A major disadvantage of the above approach is the sheer number of combinations needed to be able to enumerate all possible four pair correspondence points. To be more precise, for a 3D image stack, if the reference image produces a set of n reference feature points and the sensed one a set of m points, it would be necessary to enumerate through n^4m^4 pairs of points before all possible combinations are exhausted, which would render the brute-force method impractical when used with large feature sets containing many hundreds of points.

As suggested by [59] a better approach is to use Euclidean Minimum-Spanning Trees (EMST) to determine the correspondence between different feature points. EMST have the property that a point set used to generate the tree is “similarity invariant and noise used in area of the set generally has no effect on the portions of the MST spanning other areas distant from the noise”[59]. This implies that the edges connecting genuine soil particles would rarely be affected by the presence of outliers in the feature set.

Prim’s algorithm [60] can be used to compute the minimum spanning tree of both feature sets independently. The algorithm is initialised by marking all points as unvisited. It then randomly picks one, calculates the Euclidean distance to all other unvisited points, chooses the point closest to the initial selected point and creates an edge between the two points; in effect forming an initial spanning tree. It then repeats the process by calculating the

distance to all remaining unvisited points from all points currently in the spanning tree and at each iteration always chooses the point with the smallest distance i.e. the point closest to the spanning tree. Once all points in the set are opened and connected with edges to other neighbouring points the EMST is complete.

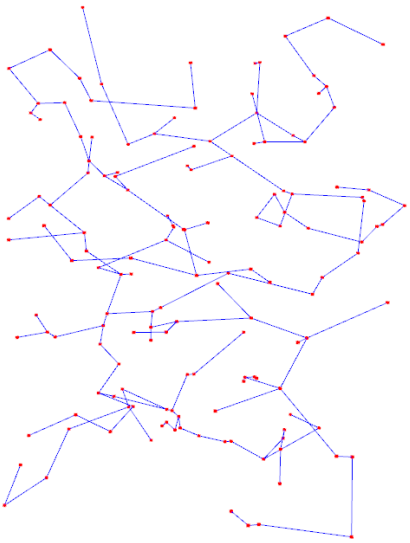


Figure 5.4 Minimum Spanning Tree of a feature set of soil particles' centre points

Using the EMST, the feature correspondence algorithm then looks for similarities between individual points in the reference and sensed feature set. This requires only *n x m* combinations of points in order to determine a set of four best matching points which would be approximately an order to twelve times faster than the brute-force method (assuming hundreds of feature points in each set). The angles and length ratios of the edges connecting a point to its neighbours in the EMST can be used as similarity invariant to compare individual points. Genuine feature points which are connected to other genuine feature points normally have connecting edges with similar angles and length ratios in the reference and sensed image, irrespective of the nature of the transformation (provided that it is a rigid transformation). Further assumptions can be made regarding the initial orientation of the soil columns when positioned in the CT scanner. Assuming that the images aren't flipped upside down, matching feature points will appear at approximately the same depth in the 3D image stack. All these can be put together in a minimisation term and calculated for every set of two feature points:

$$E = \sqrt{l^2 + a^2 + d^2}$$

Equation 5.2 Minimisation term for ranking the correspondence pair points generated by the MST algorithm:
a is the smallest angle between the edges connecting every node with other nodes in the EMST (as a coefficient between 0.0 and 1.0 where 0.0 is 0^o and 1.0 is 180^o)
l is the length ratio of the edges of the smallest angle
d is the ratio between the depth (z coordinate) of the control point and the total depth of the image stack

The above minimisation term is computed iteratively for all feature points in the reference and sensed set and the feature point coupled with the lowest score are paired. This is repeated until no points are left in either the reference or the sensed sets. The correspondence points are be ranked by score – points with a lower score indicate there is a good match between all terms, making them more likely to be genuine soil particles than outliers and vice versa.

Once the Minimum Spanning Tree produces the list of all ranked pair correspondence points, it is necessary to determine a tuple of 4 genuine correspondences to be used by the affine linear transformation to register the sensed image. There are a number of ways to test if the list contains the required minimum of 4 pair points and to determine which ones they are with the most basic approach being brute-forcing all possible combinations. Since the algorithm only needs to look at combinations of 4 points, a brute-force algorithm is polynomial in time complexity with a Big-Oh of $O(n^4)$. Whilst the brute-force algorithm is almost always exponential in running time in real-world applications, it should be noted that in this case it is polynomial. This makes using the brute-force approach feasible, however, it is still undesirable as it will take significant amount of processing power on a conventional desktop computer system.

As it is often the case with optimisation algorithms, the problem could be solved by being modelled to run on a Probabilistic Turing Machine. One such algorithm is RANSAC (Random Sample Consensus) [61]. The RANSAC algorithm iteratively evaluates randomly selected samples of 4 correspondence pair points and it estimates how well each subset matches the rest of the feature points in the entire set (using Hausdorff distance [57]). Subsets which produce good registration models are kept and those who don't are considered outliers and are thrown away. Figure 5.5 Illustrates running a RANSAC algorithm over a random set of points plotted in 2D space, half of which follow are inliers and follow a linear model and the other half are randomly positioned outliers which are not related to the model. The RANSAC

algorithm is able to successfully recover and fit the linear model using the inliers and discarding the outliers without being given any prior information or knowledge about whether each point belongs to either class.

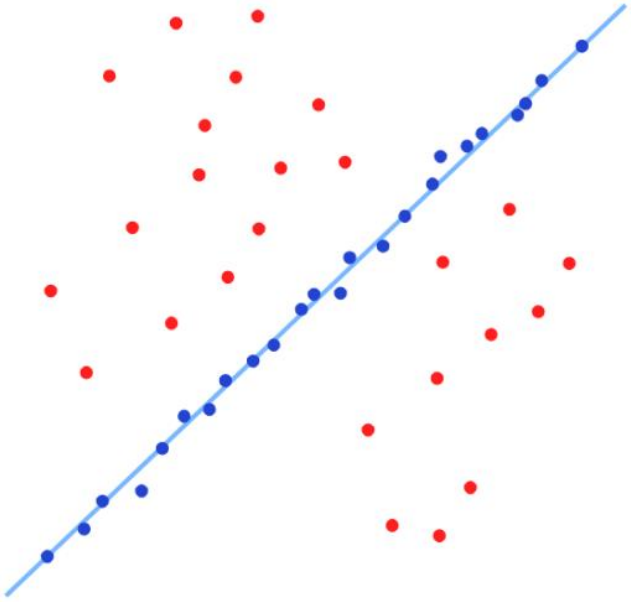


Figure 5.5 RANSAC regression of set of 2D points. The blue points have been plotted according to a linear model and the red dots are plotted randomly on the 2D plane. The RANSAC algorithm is able to successfully separate the two classes and generate its own linear model by correctly using only inliers and discarding the majority of outliers. (Credits: Picture[22] by Msm[23] licensed under CC BY-SA 3.0[24])

The RANSAC algorithm needs to run *T* interactions in order for there to be sufficient probability that it will successfully identify a set of 4 point pairs if such a subset exists in the output of the MST step. However, it can be shown that for the practical problem of registering soil columns, the RANSAC algorithm will need to run as many iterations as an exhaustive brute-force search in order for it to reach an acceptable level of probability of success which is due to the fact that the number of genuine pair correspondence points (inliers) is very small in relation to the number of outliers as it has been observed in the sample data set presented in Figure 5.6 discussed later in this chapter.

The number of combinations a brute-force algorithm has to perform in the worst case scenario is given by the equation of Combinations in 5.3 below.

[22] https://en.wikipedia.org/wiki/File:Fitted_line.svg
[23] https://commons.wikimedia.org/w/index.php?title=User:Msm&action=edit&redlink=1
[24] https://creativecommons.org/licenses/by-sa/3.0/deed.en

$$C = \frac{n!}{k!\,(n-k)!}$$

Equation 5.3 Number of brute-force combinations

n *is the total number of samples (in the example given above n=40 pair points in Figure 5.6 further down)*
k *is the number of elements in the set. For the purposes of the registration algorithm k is a constant and k=4 (minimum 4 pair points required by the affine transformation algorithm to work)*
C *is the number of combinations/worst-case number of iterations the brute-force algorithm needs to perform*

The number of iterations which the RANSAC algorithm will have to carry out in order to satisfy a given probability of success on the other hand is given by the following equation:

$$T = \frac{\log(1-p)}{\log\left(1-\left(\frac{i}{n}\right)^k\right)}$$

Equation 5.4 Number of iterations RANSAC algorithm performs

n *is the total number of samples (n=40 for the given example in Figure 5.6 further down)*
p *is the probability of success the RANSAC algorithm is desired to achieve*
i *is the expected number of inliers out of the total number of sample points. i can be given as an absolute value or as a proportion of the total number of points*
T *is the number of iterations the RANSAC algorithm runs*

The limit $\lim_{n\to\infty}(C/T)$, which represents how much slower the brute-force algorithm is compared to RANSAC, is constant for a given probability *p*. Table 5.5 gives the values of the ratio for common probabilities of interest:

Probability	95%	99%	99.9%	99.99%	99.999%
Ratio *C/T*≈	3.54	2.3	1.53	1.15	0.92

Table 5.5 List of ratios of Brute-force vs RANSAC performance for given common desired probabilities

For example, if we desire RANSAC to be successful in identifying corresponding point pairs 95% of the time, we might expect a constant speed-up factor of 3.5x. However, a 95% success rate wouldn't really be acceptable if a batch of multiple image stacks need to be registered as it will imply that 1 in 20 fail to register. A more rational choice would be 99.9% certainty. In practical terms however, the above table should be taken with caution as there are other factors to be considered when taking into account the performance of the two algorithms. For instance, the RANSAC algorithm will have additional overhead at each iteration as it needs to generate new random combinations of 4 pairs; many standard implementations would do that using pseudo-random number generators based on

calculating hash functions. Calculating a hash function will be slower than simply updating a counter as in the brute-force method and if both need to perform comparable number of iterations of the same order of magnitude, the RANSAC algorithm may end up being slower despite needing fewer iterations as each iteration will take longer to complete. Another disadvantage of using the RANSAC algorithm in this instance will be that unlike a brute-force exhaustive search, it doesn't guarantee correctness – whilst a valid solution may in fact exist in the dataset calculated by the MST algorithm in the previous step, the RANSAC instance may simply be unlucky and not find it. An exhaustive search of the available search space performed by the brute-force algorithm on the other hand guarantees that if a solution exists, it will be found and if multiple solutions exists, the optimal one (the one which provides the most accurate registration once the affine linear transformation is applied to the referenced and sensed image stack) will be found.

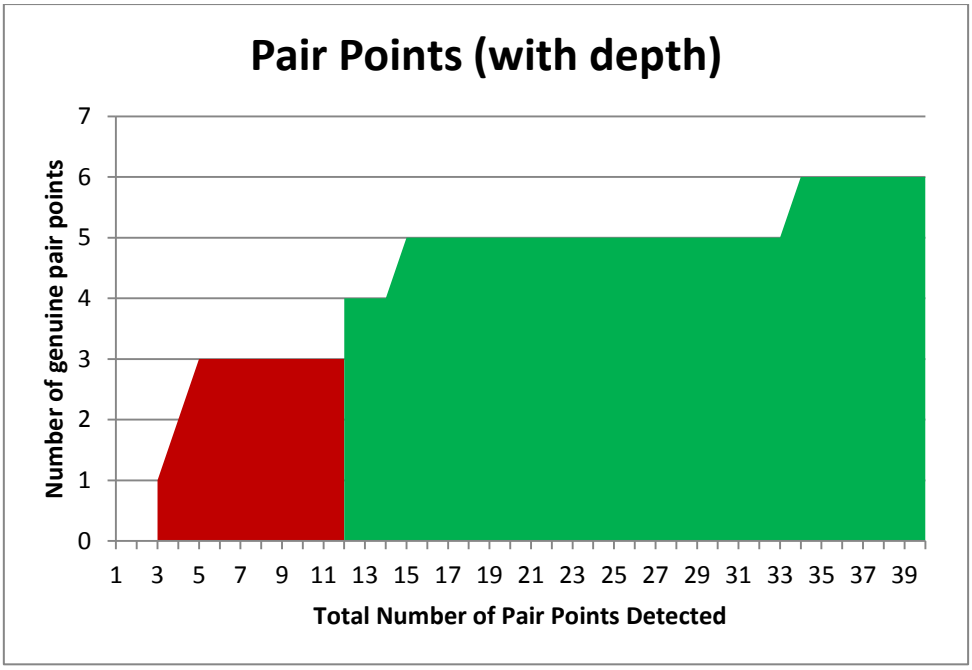
For the reasons outlined above, the registration algorithm in this thesis uses exhaustive brute-force search of the list of corresponding point pairs by iteratively testing each combination. It should be noted, however, that future implementations which need to register larger datasets (such as those generated by the Large CT scanner outlined in the Background Chapter) will likely need to deal with thousands if not tens of thousands of feature points per volume and may well need to consider using a probabilistic algorithm similar to RANSAC in order to successfully filter out the genuine correspondent points from the outliers. In order to do that, a future improved implementation of the registration method presented in this chapter will need to significantly improve the success rate of the ranking algorithm used to sort pair points and evaluate how likely they are to be genuine. The calculations shown above which imply that RANSAC is no better than brute-force are only valid if assumed that due to the poor performance of the matching algorithm, the ratio between inliers and outliers in the set of pair points generated by the MST is quite low. If the number of inliers is comparable to that of outliers than a hypothetical implementation of the RANSAC algorithm will drastically outperform any practical implementation of brute-force.

The corresponding point pairs are selected sequentially, choosing those with higher ranks first. All possible combinations of 4 points among them are considered to determine the best 4 candidates to be used to derive the transformation parameters. For every 4

correspondence pair points the transformation parameters are calculated and used to register all remaining points. The Euclidean distance between the registered and sensed points is used as a measure of how good the 4 candidate correspondence points represent the underlying transformation.

Various scoring methods can be used to determine the best match – according to [57] a small Hausdorff distance (the maximum of the minimum distances between the registered and sensed points) is a good indicator that estimates if the transformation parameters are correct, however, in the presence of large number of outliers it could often fail. The problem could partly be resolved by introducing a threshold above which the minimum distances are not taken into consideration since in most cases registered outlier points will lie far away from other sensed points. The solution adopted is to use the sum of all inverted minimum distances. This way, a good set of 4 pair points which accurately register most genuine feature points will result in large total sum. Incorrect registrations on the other hand will likely set both the genuine points and outliers far away from one another resulting in a much smaller sum. The combination of 4 pair points with the largest sum is therefore used for the transformation of the sensed image.

Figure 5.6 demonstrates the effectiveness of the use of minimisation terms in the MST correspondence pair matching algorithm in order to improve the rate of successfully matched points. Not all point pairs matched by the MST algorithm will be genuine corresponding points even though the minimisation term may provide the lowest possible values. However, it is sufficient for the MST method to identify 4 genuine corresponding feature points out of the many pair points in order for the subsequent Hausdorff distance algorithm to accurately determine the correct registration parameters. Any combination of 4 matching points which includes point pairs which do not correspond to one another is likely to result in severely deformed registration coefficients, in effect sending the majority of registered points far away from their intended corresponding reference points and thus scoring low by the Hausdorff metric. The higher the number of genuine matching points, the richer set the Hausdorff distance algorithm has to choose from, further improving the accuracy of the registration. The difference between genuine and false-positive matching correspondence pairs is illustrated in Figure 5.7:



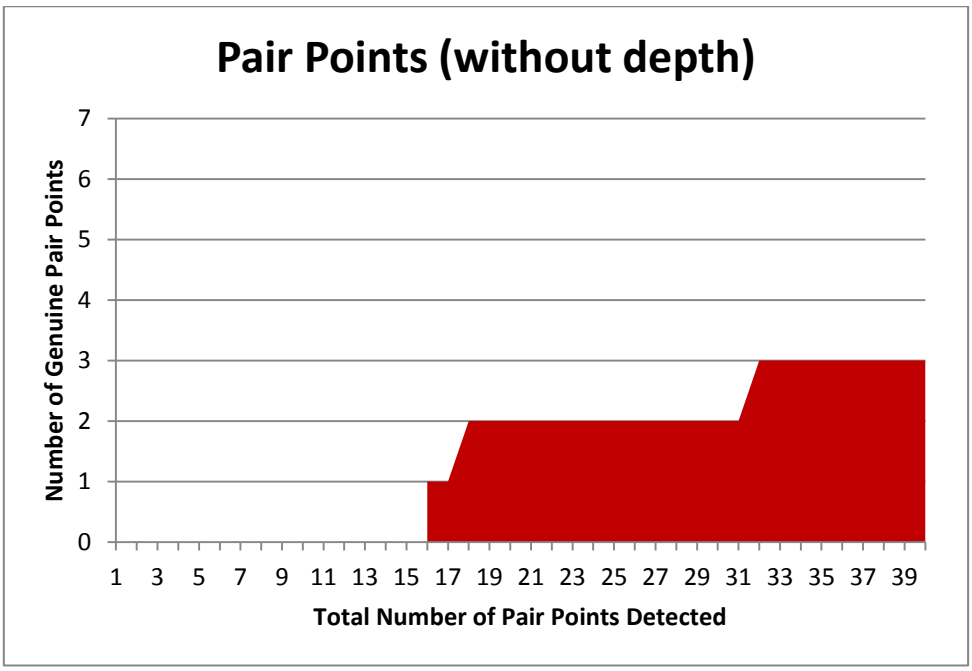


Figure 5.6 Correspondence pair points between the reference and sensed feature sets matched by the MST algorithm. The set of control points on the graphs above shows the number of genuine correspondent pair point out of the total number of pair points when ranked by their similarity. A minimum of 4 genuine correspondent points are needed for the 4D registration to be successful. The top graph shows the ranking of genuine correspondence points when depth is used as a feature vs if depth is not used as a feature as is the case of the bottom graph. It can be seen that using depth ensures that 4 genuine correspondence points can rapidly be detected as soon as the first 12 pair points are examined and any further exhaustive search will increase the pool of available genuine correspondence points (area in green) thus improving the accuracy of registration. If no depth is used as a ranking feature, the algorithm fails to reach the minimum of 4 genuine correspondence pair points as it can be seen in the graph below (red area) and as such the algorithm fails to register the 4D image stacks.

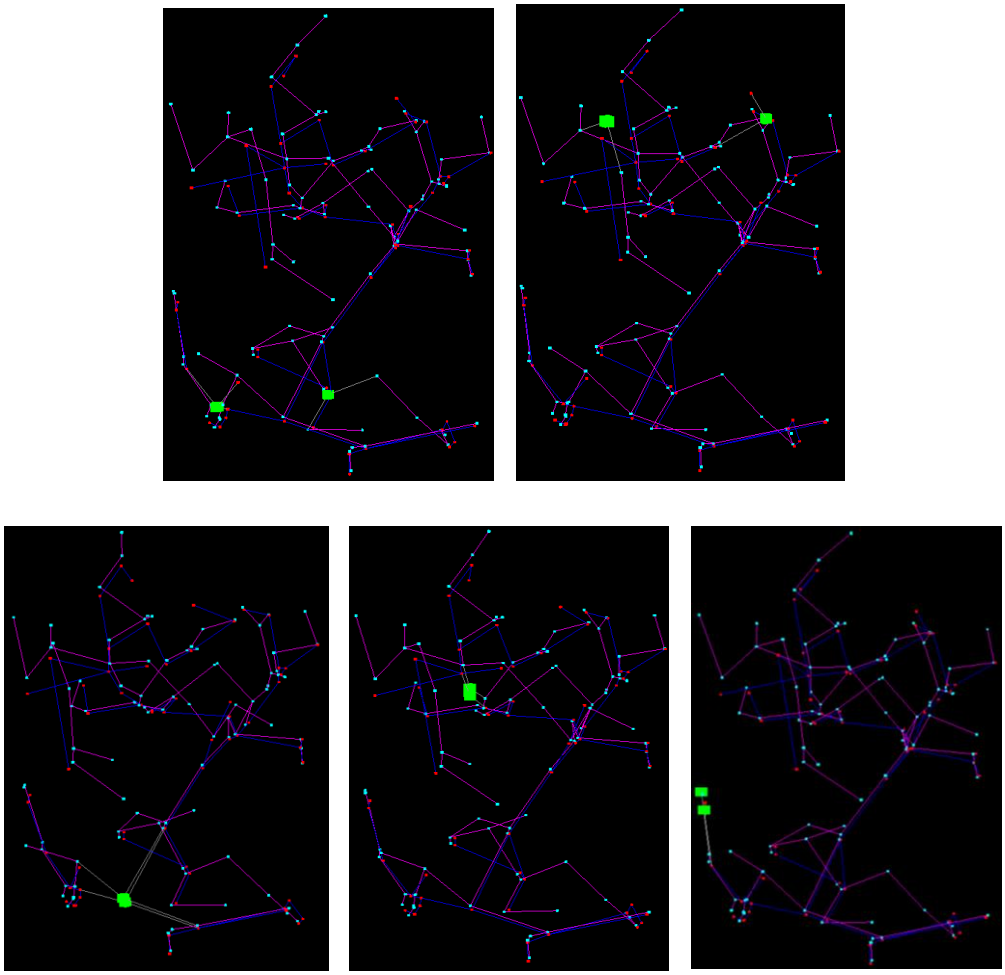


Figure 5.7 Difference between false positive (first row) and genuine pairs of matching correspondence pair points

As can be seen from the data in Figure 5.6, the first metric which used depth as a feature visualised in the upper graph provides twice as many genuine correspondence pair points as that of the bottom graph where no depth information was used. In total the depth ratio metric produces 6 genuine correspondence pair-points vs only 3 for the one lacking the term. Overall, after enumerating through all combinations of pair points, the Hausdorff algorithm correctly identifies the 4 out of 6 best matching points out of the total of 40 pair points as the correct corresponding pair points and uses them to register the image stacks. In the second case, since only 3 of the pair points are genuine and the remaining ones are not, the algorithm incorrectly selects 4 pair points at random which are used to register the image stack, resulting in an incorrect and deformed registered image as illustrated in Figure 5.8.

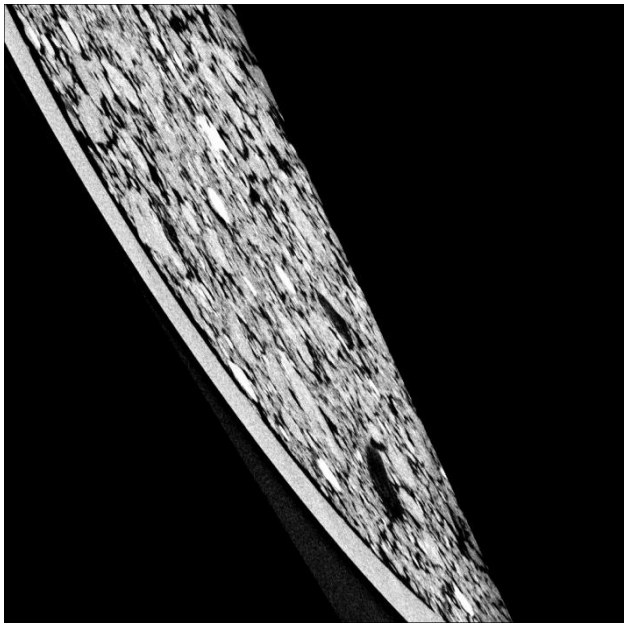


Figure 5.8 Cross-section of incorrectly registered 3D volume resulting from incorrectly selecting the pairs of correspondence points

The nature of the underlying transformation affecting the sensed image with respect to the reference image can be described using different categories of mathematical functions. The method proposed here employs the affine linear transformation, defined by a set of 12 independent parameters as follows (Equation 5.6):

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & j & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Equation 5.6 Affine Linear Transformation – the parameters *d*, *h* and l encapsulate the translation just as in the similarity transformation and the parameters *a*, *b*, *c*, *e*, *f*, *g*, *I*, *j* and *k* the rotation and scaling.

When determining the four best candidate correspondence points from the list of correspondence pair points, the affine linear transformation is used to register the reference points and calculate the difference with respect to the sensed points. The resulting 12 parameters are then used to register every voxel coordinate in the reference image to that in the sensed image.

Since the resulting voxel coordinates are floating-point numbers and the references and sensed image have integer only coordinates, trilinear interpolation is used to sample the voxel values of the sensed image in order to determine the voxel value of the registered image. Instead of sampling a single voxel value, for every registered voxel coordinate a total

of 8-neighbouring voxels (forming a cube) are sampled from the sensed image and the weighted sum of their values is used to determine the exact value of the registered voxel. The weight of each voxel is the degree of overlap between the floating-point and integer coordinates respectively [57].

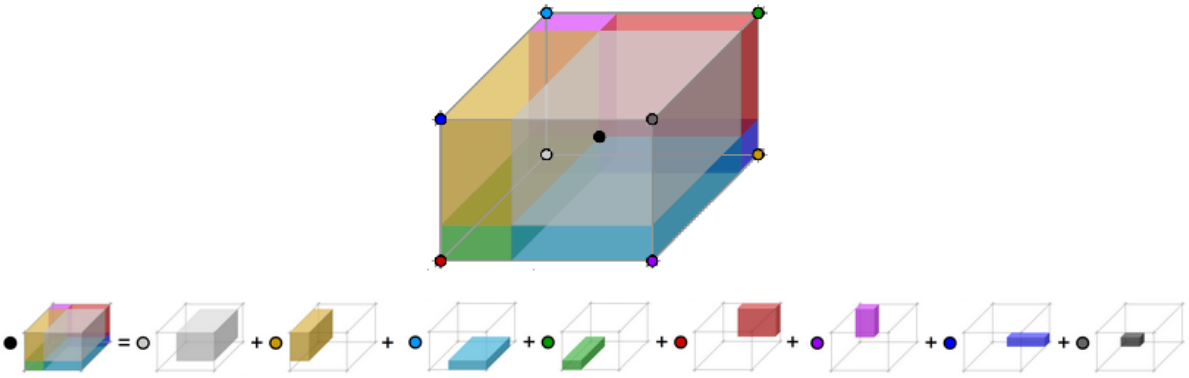


Figure 5.9 Weighted sum of trilinear interpolation (Credits: Photo[25] by Cmglee[26] licensed under CC BY-SA 3.0[27])

5.3 Global Registration Results

As discussed above, a major issue for any registration algorithm is the accurate selection of feature and correspondence points in both the reference and sensed images. Due to the very large number of discrete features present in the 3D scans of soil columns, a generic feature selection algorithm could fail to determine meaningful features which are relevant to the registration process. In order to evaluate the performance of the registration approach based on large soil particle extraction the registered results were compared with those provided by the TurboReg Fiji plugin [62] which is a popular tool used in the plant science community. CT scans of soil columns planted with tomato plants were used, which were taken 4 days apart and watered at least one between the scans. Since TurboReg only registers individual 2D slices in the image stack one by one, before the data was loaded the 3D stack was manually shifted along the z axis in order to approximately align them and allow the 2D registration algorithm used by TurboReg to receive similar images.

[25] https://en.wikipedia.org/wiki/File:Trilinear_interpolation_visualisation.svg
[26] https://commons.wikimedia.org/wiki/User:Cmglee
[27] https://creativecommons.org/licenses/by-sa/3.0/deed.en

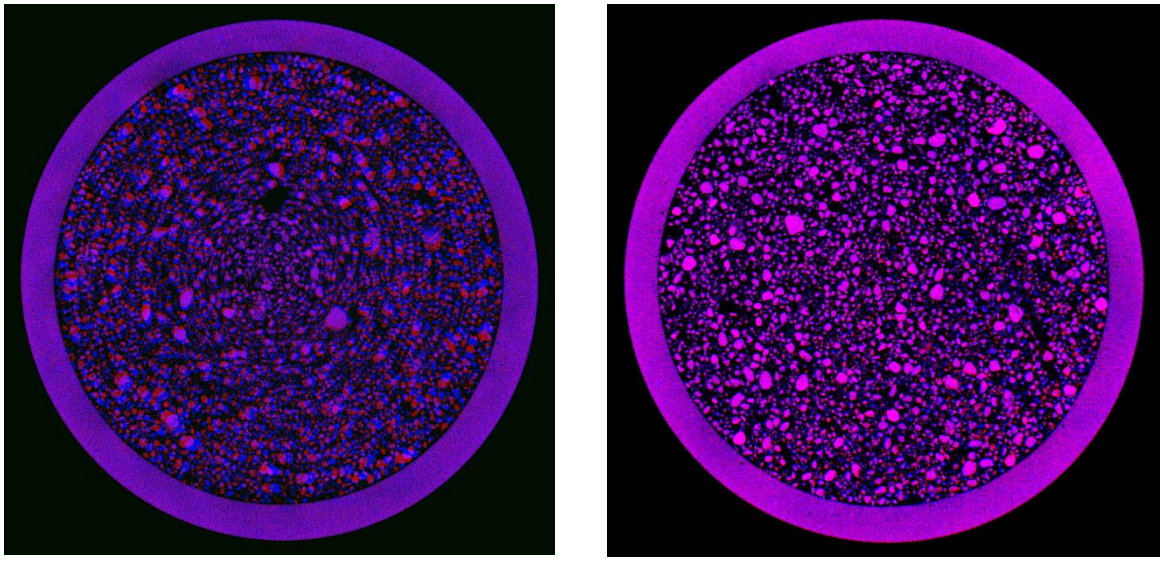


Figure 5.10 Comparison of TurboReg (Left) and 3D soil particle registration (Right)

TurboReg offers a number of different modes of registration which can affect the accuracy of the registered image at the expense of the amount of computational time the registration algorithm will take to complete the task. Namely, TurboReg operates in the following modes: translation mode – the registration can only detect and correct offset in either the X, Y directions; rigid body registration – as well as correcting for translation, the registration process can also account for rotation along the Z axis; scaled rotation – the algorithm can account for zoomed in or zoomed out images, for example, in the case of CT scans, the imaged soil may have purposefully or accidentally being placed at a different distance from the detector making the same feature appear bigger or smaller when two different time points are compared; affine transformation – it can account for complex description of the scene and it uses parameterised linear transformation to register the images; bilinear transformation mode [63].

Since RootShed used affine linear transformation to register the reference and sensed image stacks, in order to provide a fair comparison, TurboReg was run using Affine Transformation in high-quality mode. The result of registering the example 4D CT volume stacks mentioned earlier are shown in Figure 5.10. The figures visualises the differences between a sensed image slice (blue) and a registered image slice (red) at the same depth for both the TurboReg and soil particle registration methods. As can be seen from the left image, the 2D registration method has failed to determine the rotation parameter of the transformation function. More interestingly, close inspection of the right image shows

numerous small red-only or blue-only coloured soil particles. Those are soil particles which appear in the referenced image but don't or only partly appear in the referenced image and vice versa. This could largely be attributed to the movement of soil particles due to external forces (shaking, stress etc.) and internal forces (water movement, pore space contraction).

In order to provide formal evaluation of the registration accuracy and correctness of RootShed, a simple experiment was conducted whereby a CT image stack of a soil column was artificially transformed by a known amount and then registered back with itself. The transformation of the stack was performed manually using a standard off-the-shelf 2D image editor and three types of transformations were applied to the image stack – translation, rotation and scaling. The translation was performed along the z-axis by manually deleting the first and last 20 images of the stack for the referenced and sensed 3D volumes respectively, in effect, emulating a translation corresponding to 20 voxels in a vertical direction (in real-life this corresponds to the operator of a CT scanner manually adjusting the height of the table, sitting between the x-ray beam and image detector on top of which the soil column in positioned). A +90° rotation was then applied to every single individual image in the stack in batch mode emulating rotation by 90° along the Z axis (in real world, this corresponds to the CT scanner operator positioning the column facing the detector at a different angle before the scanning process is initiated). Finally, every image was scaled by 85% in order to replicate real world differences in the resolution of CT image scans which may arise from various reasons such as different quality of scans or positioning the soil column closer or further away from the detector. The resulting transformed stack was used as a referenced image and the original stack as the sensed image (the one being transformed to reproduce the registered image). The cumulative transformation matrix of the above process is shown below as calculated based on Equation 5.7:

$$\begin{bmatrix} \mathbf{0} & \mathbf{1.18} & \mathbf{0} & \mathbf{0} \\ \mathbf{-1.18} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{20} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} 1/0.85 & 0 & 0 & 0 \\ 0 & 1/0.85 & 0 & 0 \\ 0 & 0 & 1/0.85 & 20 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 90° & \sin 90° & 0 & 0 \\ -\sin 90° & \cos 90° & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 90° & 0 & -\sin 90° & 0 \\ 0 & 1 & 0 & 0 \\ \sin 90° & 0 & cos 90° & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 90° & \sin 90° & 0 \\ 0 & -\sin 90° & \cos 90° & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 5.7 Translation matrix used for the artificial transformation of a sample image stack for the purposed of validating the 4D registration accuracy of RootShed.

The above transformation however will translate the entire image into negative coordinate space because the 90° rotation is applied around the corner of the image where the 0,0 coordinate lies as opposed to a coordinate system where the 0,0 is positioned in the centre of the image as illustrated in Figure 5.11 Since many common GUI APIs operate in whole

unsigned integer arithmetic to calculate the coordinates of individual pixels, including the GUI API used by RootShed (written natively in Windows Graphic Device Interface), RootShed naturally supports only whole positive integers to encode voxel coordinates. For this reason RootShed will apply one final transformation along the Y axis to correct for the negative shift of the transformed image and bring it back into the positive quadrant of the coordinate system.

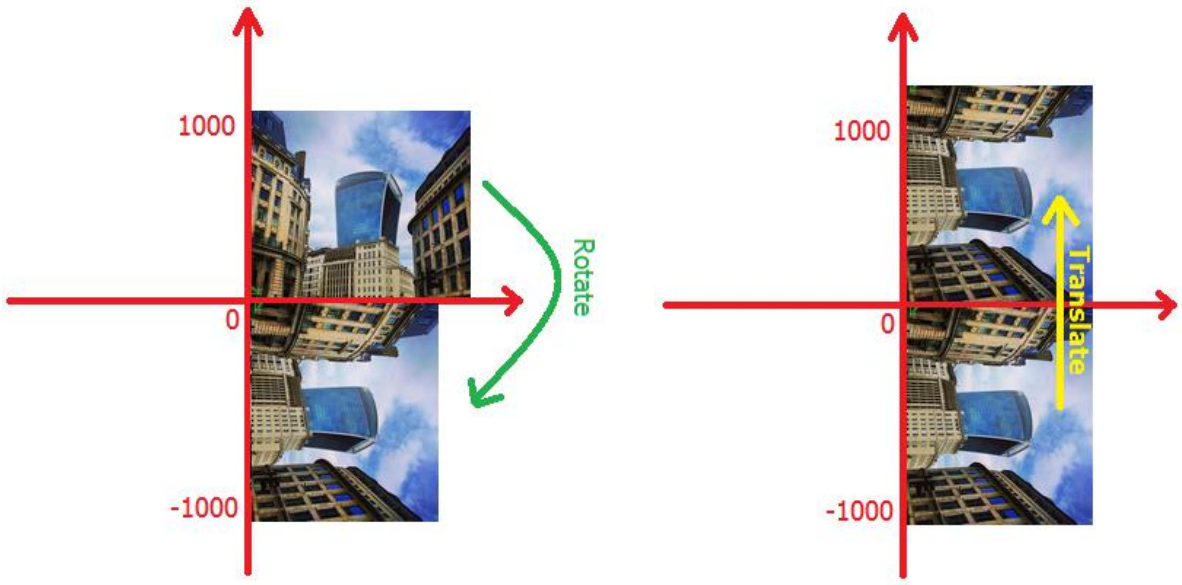


Figure 5.11 Illustration of the transformation process of the artificially generated image stack. After the image is rotated, it is positioned on the negative side of the Y axis. In order to restore positive coordinates for every image pixel, the entire image need to be normalised by translating every pixel by the height of the image.

Taking into account the above correction and the dimensions of the transformed image stack (1000x1000x500) the final transformation matrix of the artificial transformation performed on the sample image stack should look as follows:

$$\begin{bmatrix} 0 & 1.18 & 0 & 0 \\ -1.18 & 0 & 0 & 1000 \\ 0 & 0 & 1 & 20 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 5.8 Adjusted transformation matrix used to artificially transform a sample image stack

Next, the referenced and sensed image stacks were registered with using the registration method described in this chapter and the resulting transformation matrix as calculated by RootShed is shown below:

$$\begin{bmatrix} -0.01939160 & 1.168220000 & -0.00435027 & 10.4586 \\ -1.17045000 & -0.00776534 & 0.000466851 & 999.244 \\ -0.01174930 & -0.03937310 & 1.045670000 & 18.9884 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 5.9 Transformation parameters generated by RootShed's registration algorithm

The results of the transformation are shown in Figure 5.12:

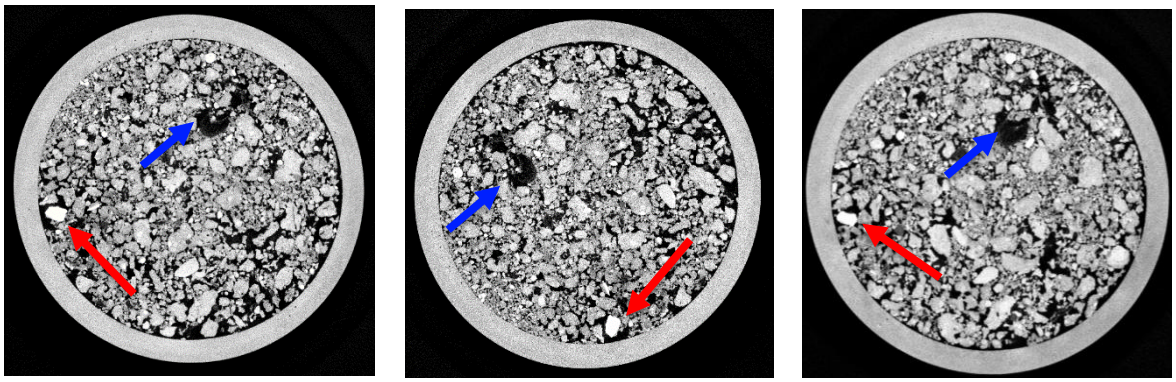


Figure 5.12 Results of validation registration of artificially transformed image stack (not to scale) which is then registered back with itself. Left image – a single slice of the original stack; Middle image – CT slice as same depth from the manually transformed stacked. Right image – slice at the same depth taken from the registered stack produced by RootShed. Red and blue arrow indicate identical features in each of the images respectively. The results indicate that the registration has been largely successful.

The affine transformation parameters estimated by the RootShed registration method are very similar to the ground truth parameters used to register the artificially transform the original stack. This is further backed up by examining individual CT slices as shown in Figure 5.12 from which it is apparent that the transformation has accurately identified the right registration parameters to be used in order to restore the original 3D image stack.

5.4 Local registration and 4D quantification of Root System Architectures

Once the registration parameters of the CT volumes are known, it becomes desirable to use the parameters to register sequences of 3D image stacks for the purpose of extracting dynamic traits from the scanned root system. As mentioned in Section 5.1 there are two main cases of evaluating dynamic traits – one where both the reference and sensed images have been segmented independently of one another either using RooTrak or any other suitable segmentation method such as manual segmentation and subsequently skeletonised with RooTh producing RSML descriptions of the different time points, and one where only a single time point has a full segmentation and RSML description of the root system and extracting static and dynamic traits from alternative time points requires using the information from the skeleton description of that time point to guide segmentation and/or skeletonisation in the other time points. The second option is discussed in Section 5.5.

There are four different ways of applying the registration parameters to the sensed skeletonised time point data in order to extract dynamic traits:

- The registration parameters could be directly applied to the raw CT volume to produce a registered CT volume which can subsequently be analysed with the segmentation-skeletonisation pipeline i.e. using RooTrak to obtain a segmented output and then RooTh to generate the RSML description and traits. A potential problem with this approach is that poor interpolation could result in image artefacts which disrupt the operation of the level-set method deployed by RooTrak, resulting in an incorrect segmentation. A further disadvantage of this approach is that in many cases it may be altogether impractical as it would imply registration needs to be run prior to segmenting the image stacks which in itself is often time-consuming and CPU intensive process and as such may not be applicable to a dataset which has already been segmented.
- The registration parameters can be used to transform the RooTrak segmented output of the sensed volume stack which can then in turn be skeletonised with RooTh. Similarly to the first case, poor interpolation could result in image artefacts in the registered segmented output which can hinder the skeletonisation algorithm used by RooTh (for instance by generating very large number of false-positive root tips or by resulting in discontinued segmentation output). The problem is further amplified by the segmented output being a binary image. One way to smooth the presence of such artefacts is to use morphological opening and closing as a post-processing step.
- The registration parameters can be applied directly to the control points generated by the minimum values of the distance map of the sensed image volume. This approach has the significant advantage that any slight changes in the relative location of control points in the registered image stack with respect to one another (as a result of floating point arithmetic for instance) are likely to be smoothed out by the Active Contours used by the skeletonisation algorithm.
- Finally, the registration parameters can be applied directly to the skeleton points of the RSML description of the sensed root system skeleton. Care should be taken if the resulting registered skeleton is used to evaluate static or dynamic traits as small

changes in the relative location of spline points with respect to one another could result in significant oscillations of the underlying piecewise spline curves due to Runge's phenomenon [64]. A possible solution to this problem is to use smoothing splines in place of interpolation splines.

In order to illustrate the sensitivity of interpolating splines to misalignment of snake points as a result of imprecise registration, a small amount of uniform pseudo-random noise generated using a hash-based random number generator provided in the C++ standard template library was added to a set of control points of a plant over which set an ACM skeleton was run. Similar noise was also independently added directly to ACM snake points, producing different skeletons. Both skeletons were used to evaluate root system traits and they were compared against the original unaltered skeleton. The results are shown in Table 5.10 and the corresponding root system skeletons used in the experiment are visualised in Figure 5.13.

Measurement	**Total Root Length (mm)**
Original	432.9
Control Points	431.6
Snake Points	458.0

Table 5.10 Total root length estimate of the effect of Runge's phenomenon on the accuracy of image registration. The first row gives the output of the original, unaltered skeletonisation, the second row – results after applying pseudo-random noise to the set of control points and the third – applying noise directly to snake points. The results suggest that snake points are far more sensitive to noise than control points.

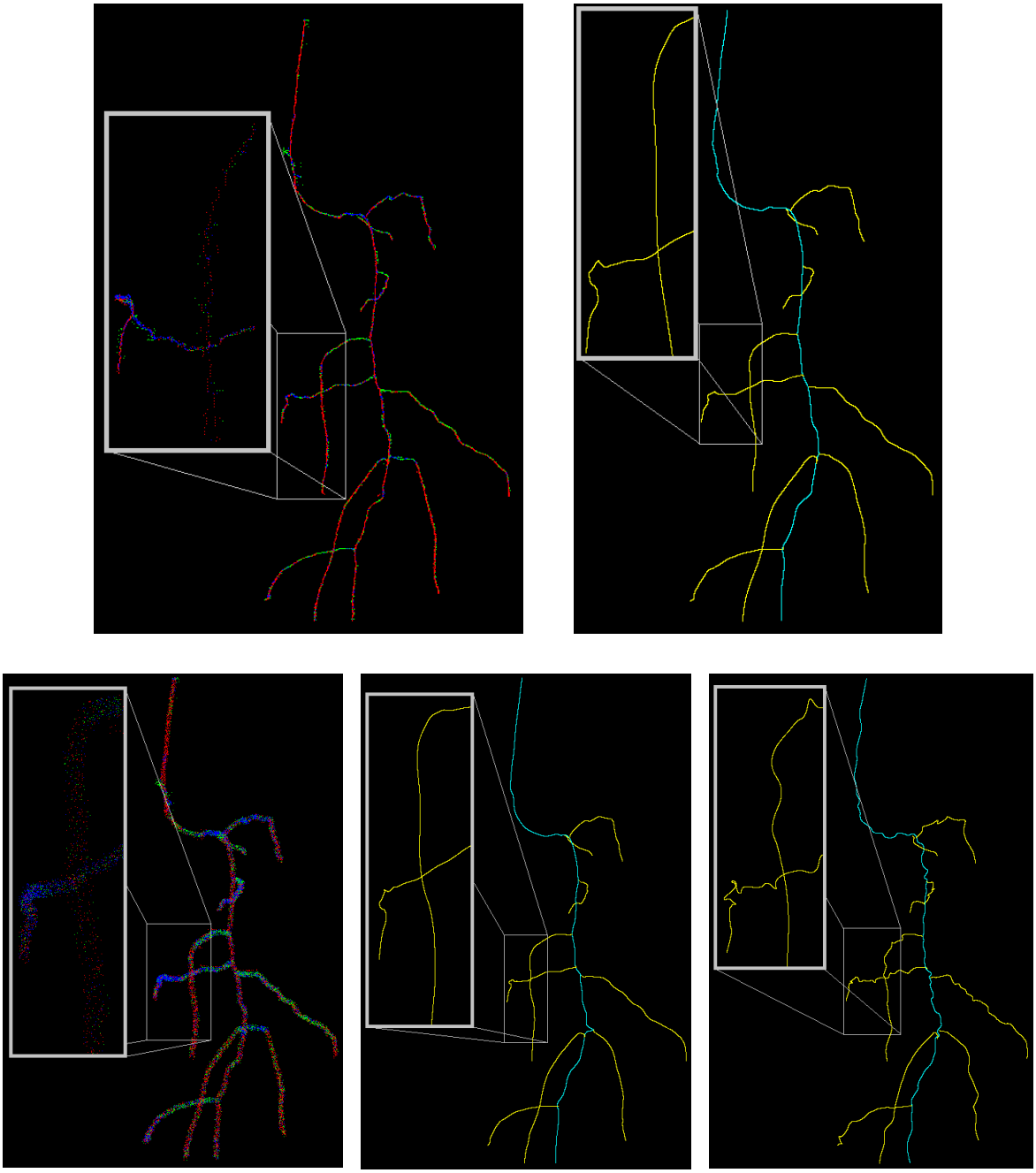


Figure 5.13 Runge's phenomenon effect on the accuracy of image registration. Top row – original referenced set of control points (left) and resulting ACM root skeleton (right). Bottom row – pseudo-random distorted set of control points (left) has little impact on the accuracy of the resulting image skeleton (middle). When pseudo-random distortion was applied directly to the snake points (right), the structure of the resulting skeleton was significantly affected as can be seen from the appearance of unusually large number of wave artefacts which are likely to hamper accurate estimation of root traits.

As predicted, the results indicate that the piecewise defined ACM curve is more sensitive to misalignment of snake points than it is to misalignments of control points. The set of control points to which random noise was added generated a root skeleton which only slightly underestimates the total root length. This is exactly as expected as the more uncertainty there is about the location of control points, the smoother and less accurate the resulting ACM skeleton will be, in effect taking shortcuts when following the outline of the underlying root and underestimating its overall length. On the other hand, adding random noise to snake points which are directly used to define the piecewise contour of the skeleton will give rise to the Runge's phenomenon resulting in large number of artefacts in the form of "waves" and "wrinkles". As a "curly" ACM curve will take longer path to cover the same distance of root length it is entirely expected that the results of the second experiment will overestimate and not underestimate the total root length. The second case overestimates the overall root length to a greater degree than the first. For that reason, when registering 4D time-series, RootShed applies the transformation parameters to the set of control points as described in option 3 in the list above as opposed to the set of snake points in order to minimise any further distortion to the extracted 4D root traits.

Once the registration parameters are applied to the root system skeleton and the reference and sensed root system descriptions are brought together, the problem of evaluating dynamic traits is reduced to the problem of identifying corresponding pairs of roots and calculating the difference between static traits such as root length, root width, root volume etc. Assuming that the global registration of the reference and sensed root system has correctly registered them and the majority of roots mostly overlap one another, the average of the minimum distance between the closest points of the reference root to every point in the sensed root could be used as a metric to determine the matching pairs. In rare circumstances where a root appears in one of the time points but not the other either due to low temporal resolution resulting in the root system being imaged before some roots have emerged or due to inconsistencies in the segmentation or skeletonisation algorithm resulting in a root being successfully skeletonised in one time point but not the other, it becomes necessary to use pre-set heuristics in the form of a threshold applied to the average distance metrics defined above. Roots which fall too far apart from any other root

in the corresponding time point are considered to be outliers and as such are omitted from the dynamic trait evaluation.

The RSML root system descriptions of three time points of a sample 4D manually segmented data set of a tomato plant at days 4, 6 and 8 were evaluated using the method defined in point 4 above and the results of the dynamic traits output are shown in Table 5.11 below and Figure 5.14.

Root	Day 4 – Day 6	Day 6 – Day 8	Day 4 – Day 8
1.	-0.11mm	-0.07mm	-0.19mm
2.	3.5mm	10.5mm	14mm
3.	0.7mm	4.3mm	5.1mm
4.	1.1mm	5.7mm	6.8mm

Table 5.11 Dynamic traits evaluation of root elongation of three point time-series of a manually segmented tomato plant. Resolution of the CT image stacks is assumed to be 25µm. Root 1 is the primary root and roots 2-4 are the lateral roots.

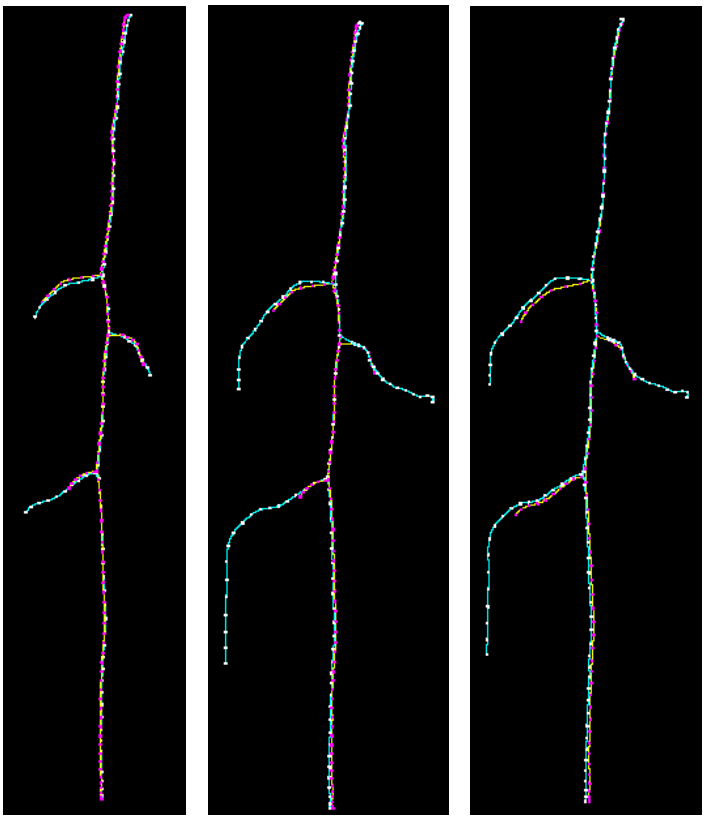


Figure 5.14 Overlapped time-series of a tomato plant illustrating the different stages of growth and the elongation of lateral roots (the primary root is cut-off by the scanning range). From left to right: day 4-6, day 6-8, day 4-8

5.5 Conclusions

Chapter 5 presents a novel, application-specific approach to the registration of 3D image stacks of soil columns for the purpose of 4D trait evaluation in the absence of external markers attached to the pot. The proposed methods allow for the natural abundance of soil particles within the soil column to be used as feature points in the image registration algorithm. As a result the registration process can account for movement and deformation occurring within the soil as a result of multiple irrigation cycles, biological activity and other complex processes taking place within the soil environment, which would not otherwise be possible to detect if one relied solely on an external marker attached to the body of the soil column. The proposed methods were implemented in a 4D registration tool named RootShed and preliminary work was carried out to evaluate the efficiency and accuracy of the registration algorithms put in place. A small number of experiments using artificially generated 4D data were used to evaluate the correctness of RootShed and overall the conclusion is that the results are promising.

Chapter 6.
Towards Time-Series assisted segmentation of Root Systems

6.1 Watershed based supervoxel segmentation

This chapter presents preliminary work exploring the possibility of using RooTrak segmented and RooTh skeletonised output to assist the segmentation of images acquired elsewhere in 4D time series. We suggest that, instead of running RooTrak independently on each image in a sequence, the knowledge acquired at previous or successive time points can be used to aid the segmentation of a given image. In essence, we hypothesise that a root architecture description obtained at a different time-point can be used to support local adaptive thresholding of a CT image – the region over which adaptive thresholding is applied being determined by the transformed root architecture.

The root material found inside a CT scan of a soil column accounts for only a small portion of the total number of voxels, largely due to the fact that plant roots are long, narrow tubes. This can be illustrated if the global histogram of an image stack is considered as shown in Figure 6.1. The two peaks in this histogram are known to arise from air pockets and soil: root material does not provide a noticeable peak as it represents a very small percentage of the total image stack volume. Any segmentation algorithm based on a global histogram of the image stack would therefore fail to distinguish the roots from the surrounding soil.

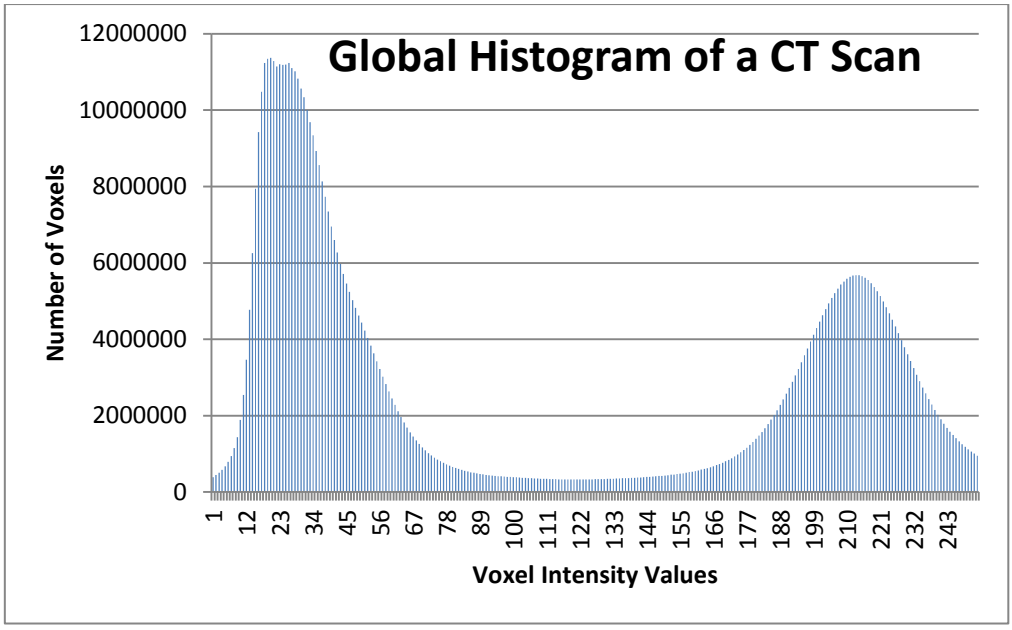


Figure 6.1 Global Histogram of an Image Stack of a soil column containing root system. Two distinguishable peaks are present – the lower band one accounting for pore space and the higher band one accounting for soil particles. Root material and organic matter is somewhere in the middle and not clearly distinguishable.

For histogram based segmentation of the root system to be successful in segmenting the root a new method is needed which focuses attention on very constrained regions of interest with a high likelihood of containing roots. The histogram of such a region is more likely to support root-soil segmentation as the numbers of root and soil voxels are likely to be more balanced, revealing any systematic differences between the two classes. This is where skeletonised root output from previous or successive time-points, registered with the current image stack due to be segmented is of value: the root system skeleton can indicate the expected location of roots in the new image. A fixed width region around the skeleton of each root could be examined for root matter. If a histogram is plotted of voxels within a fixed distance of the skeleton of a transformed root system (fixed width-tube), we observe that 3 main peaks appear (Figure 6.2). Examination of the voxels contributing to the middle peak shows that this represents the root material.

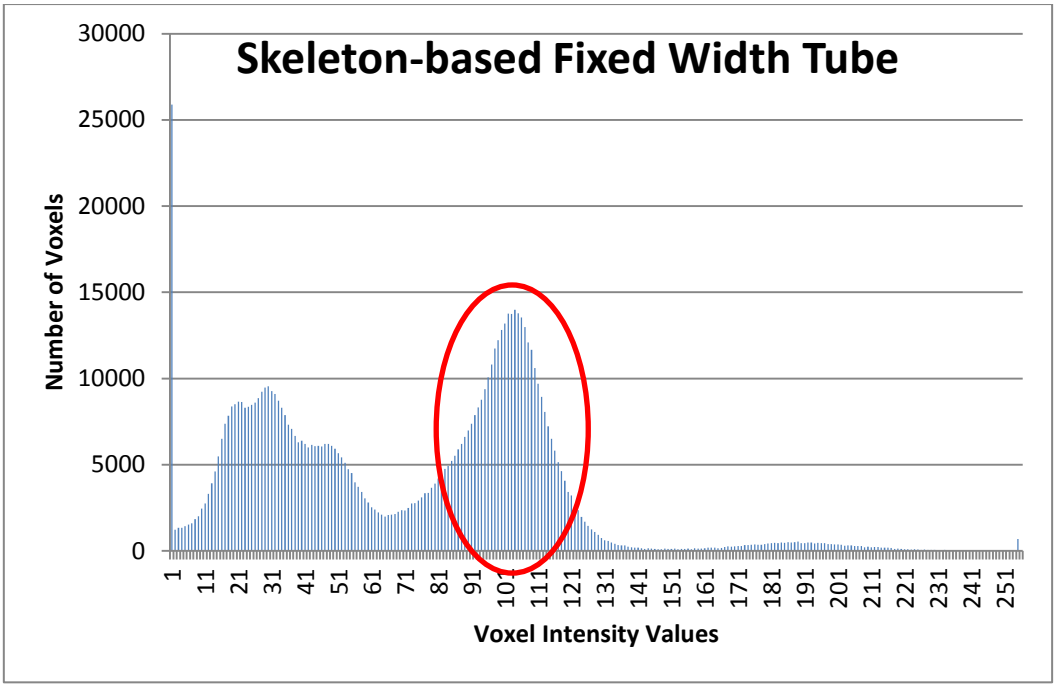


Figure 6.2 Fixed Width Tube Histogram. The middle peak, which is distinct from surrounding peaks, represents root material

This would allow for the root system to be segmented by classifying all voxels within a certain distance of registered root skeleton either as part of the segmented root material or soil. However, due to the presence of noise in the raw CT data, many voxels may lie out of the range and thus result in very grainy segmentation. In order to improve the segmentation, the super-voxels output by the watershed algorithm described in Section 5.2 are used as a basis of classifying the CT column volume into root and non-root material. The mean value of the voxels making up each super-voxel is used to determine its inclusion or exclusion into the root material boundary. The centre points of the supervoxels (the average of the coordinates of the underlying voxels contained in the supervoxel) are used to calculate their distance from the nearest point of the root skeleton and if they lie within fixed width tube threshold, they are considered for classification as either root or non-root material based on the average intensity value of the supervoxel as mentioned above.

The resulting segmentation significantly reduces the effect of inconsistent classification due to the noise variation of pixel intensity values resulting in less grainy segmented output. Figure 6.3 illustrates the result of applying the histogram thresholding over a set of supervoxels and the resulting segmented root-cross sections. The search region is defined around a set distance from every point lying on the root skeleton and is deliberately defined

to be larger than the expected root width in order to allow for some tolerance in cases where the global registration doesn't perfectly align the registered RSML skeleton with the actual location of the referenced root system in the raw CT data.

However, a small number of misclassified super-voxels are still present around the root walls which fall within the search region but which don't belong to root material. In order to further minimise the presence of any segmentation artefacts, connected-components are used to cluster all segmented supervoxels. The majority of the root material will naturally cluster in very large regions in size encapsulating whole roots (the whole root system in an ideal case) whereas small parasitic segmented supervoxels around the root wall remain isolated. Simple threshold classification based on the resulting clustered region's size is used to discard all parasitic segmented artefacts and leave the segmented root system only. The use of supervoxel classification versus classifying individual voxels further aids the connected-components algorithm, allowing for more complete segmentation of the output than the one that would be possible if thresholding was used directly on top of simple voxels.

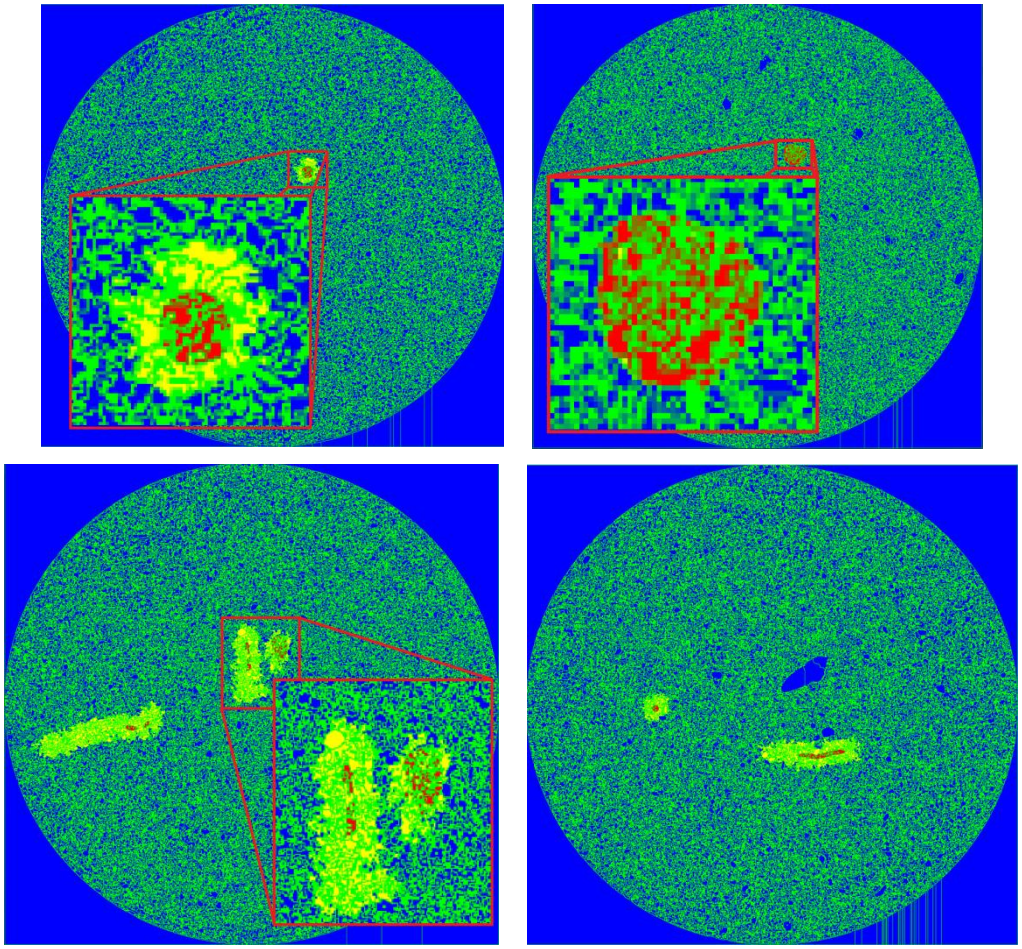


Figure 6.3 Example of 4D assisted root segmentation. Green borders represent supervoxels formed after running the split and merge watershed segmentation algorithm. Yellow regions colour all supervoxels lying within a certain distance of the root skeleton. Red regions are supervoxels which have been classified as root material based on their average intensity values and the global histogram of the yellow regions.

According to [46], the intensity values of the root material voxels change with the depth of the root system. This undermines the use of a single threshold in order to classify supervoxels into root and non-root material. Instead, a better solution is to compute local thresholds from a sliding window running along the length of each root skeleton. The histogram range is calculated locally for that window and neighbouring supervoxels lying within the classification distance away from the root systems are classified according to that histogram. Figure 6.4 shows the sliding window histogram at different positions along of the main root. It is evident that the intensity of the root material changes as the sliding window traverses along the length of the root.

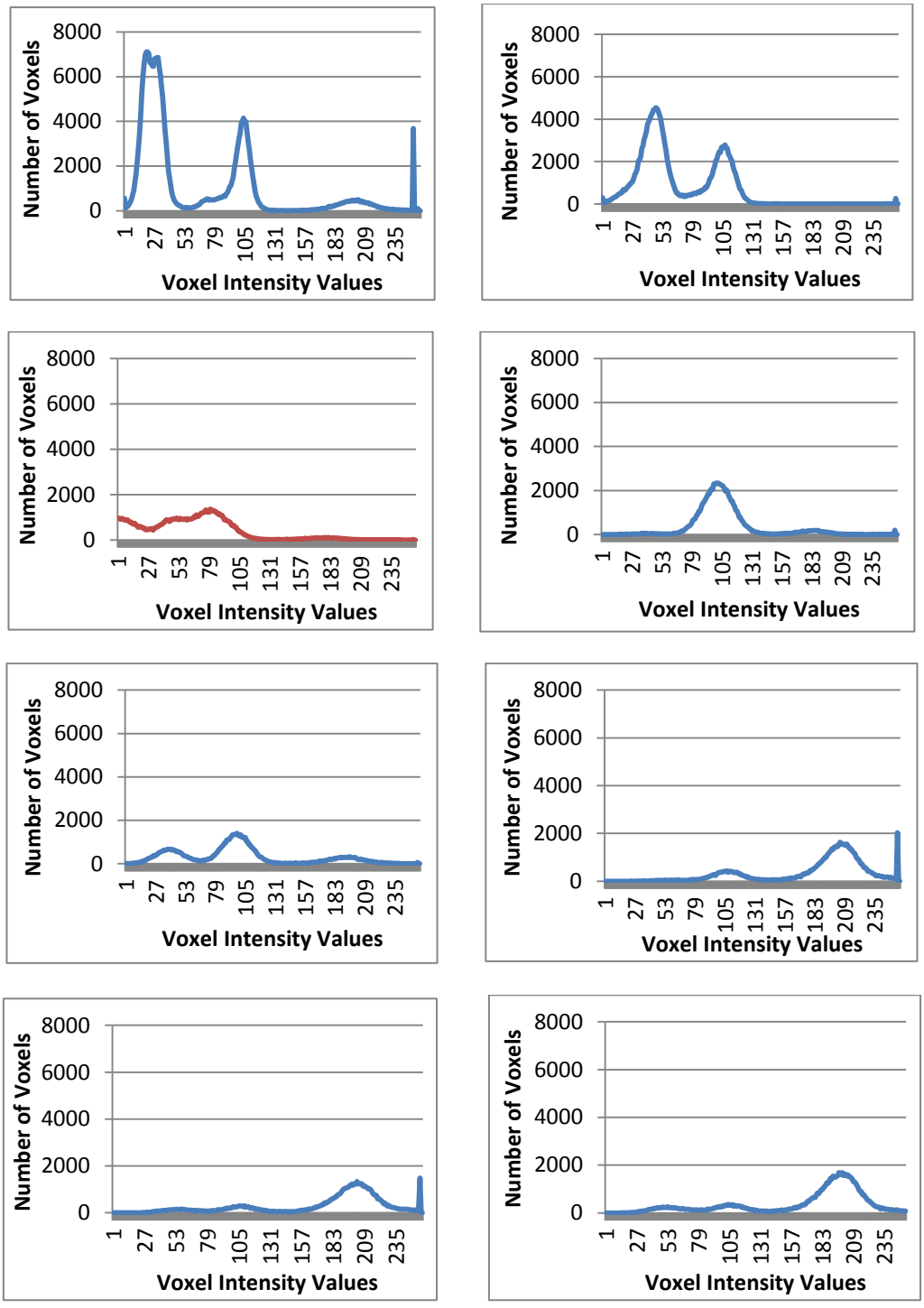


Figure 6.4 Local histogram charts of sliding windows in a single section of a primary root. It is clear that the intensity values of the root material change along the length of the root emphasising the need for an adaptive histogram approach in lieu of a global histogram. X axis – voxel intensity values. Y axis – number of voxels having a specific intensity value in each bar. The red-coloured graph illustrates a section of the CT image stack where the boundary between the root material and surrounding soil is less clear (no clear peaks) due to overlapping.

The use of local histograms creates its own problems in certain cases due to the reduced number of voxels used to make up the histogram. From the sliding window example in Figure 6.4, it is noticeable that the peak in the red-marked histogram of a section of the root is less clear; this is due to larger than usual overlapping of the intensity ranges associated with roots and adjacent pore space. A straight-forward solution to such cases is to use a weighted histogram which amplifies the root peak and silences surrounding non-root material voxels as demonstrated in Figure 6.5 by comparing it to a fixed-width histogram for the same sliding window (the red-labelled histogram) in Figure 6.4. The weighted histogram weighs each voxel based on how far it is from the nearest point lying on the root skeleton and it logically gives higher weight to those voxels lying closer to the skeleton.

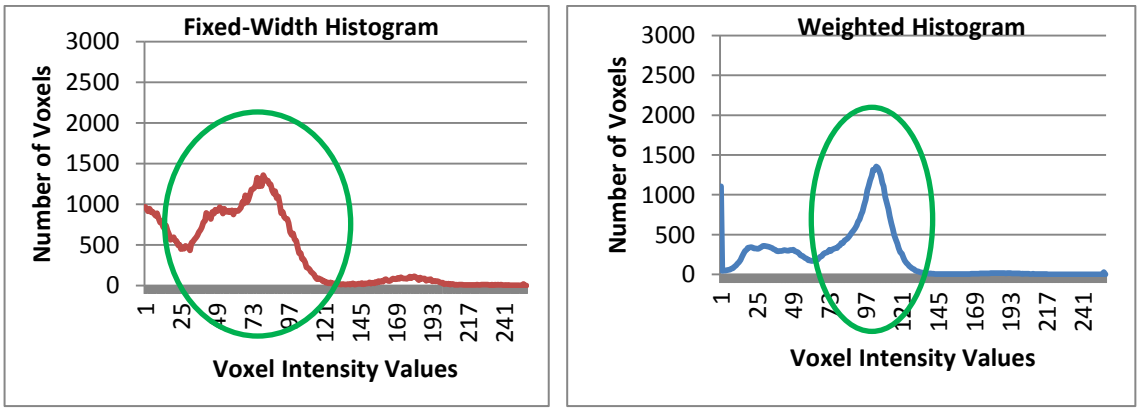


Figure 6.5 Comparison of Fixed-width and Weighted Histogram of a primary root of a tomato plant. The weighted histogram significantly improves upon the separation of root material from the surrounding soil as can be seen by comparing the shapes of the global maxima in the two histograms which is representative of root material (indicated by green ovals).

The above weighted histogram was calculated over the same section of the primary root and after cropping the values of 0 and 255, the root peak becomes much more visible. A coefficient can be applied to the inverted distance from the corresponding root skeleton and used as a weight coefficient and that would smooth out the histogram. Using too much or too little weighting would either be equivalent to applying a fixed width tube (i.e. all voxels lying within a fixed distance away from the skeleton) or applying a root width tube (all voxels lying within a certain distance away from the skeleton according to the width of the root at that skeleton point as measured by RooTh) so the right balance would lie somewhere in the middle. The goal of this method is to use an appropriate amount of weighting such that assuming the root is not too far off the expected location in the sensed image stack compared to the reference one, it will accurately appear on the histogram as

the only major peak and a simple normal distribution can be used to approximate the range of intensity values for the root matter as opposed to having to deal with multiple peaks if a non-weighted histogram is used.

Based on the observations made above, the proposed method for the segmentation of time series of CT scans involves using an adaptive sliding window histogram to determine the mean and standard deviation of the root material around a fixed-width tube encapsulating the skeletonised RSML output. The peak and standard deviation of the root material is determined using half-width Gaussian Distribution [65]. Figure 6.6 visualises a 3D rendered segmentation of root using the above method. For the purposes of testing the proposed segmentation method as a proof of concept, the referenced raw CT volume and the sensed RSML skeleton are that of the same time-point meaning that they are perfectly registered by default making the task of segmentation more straight-forward.

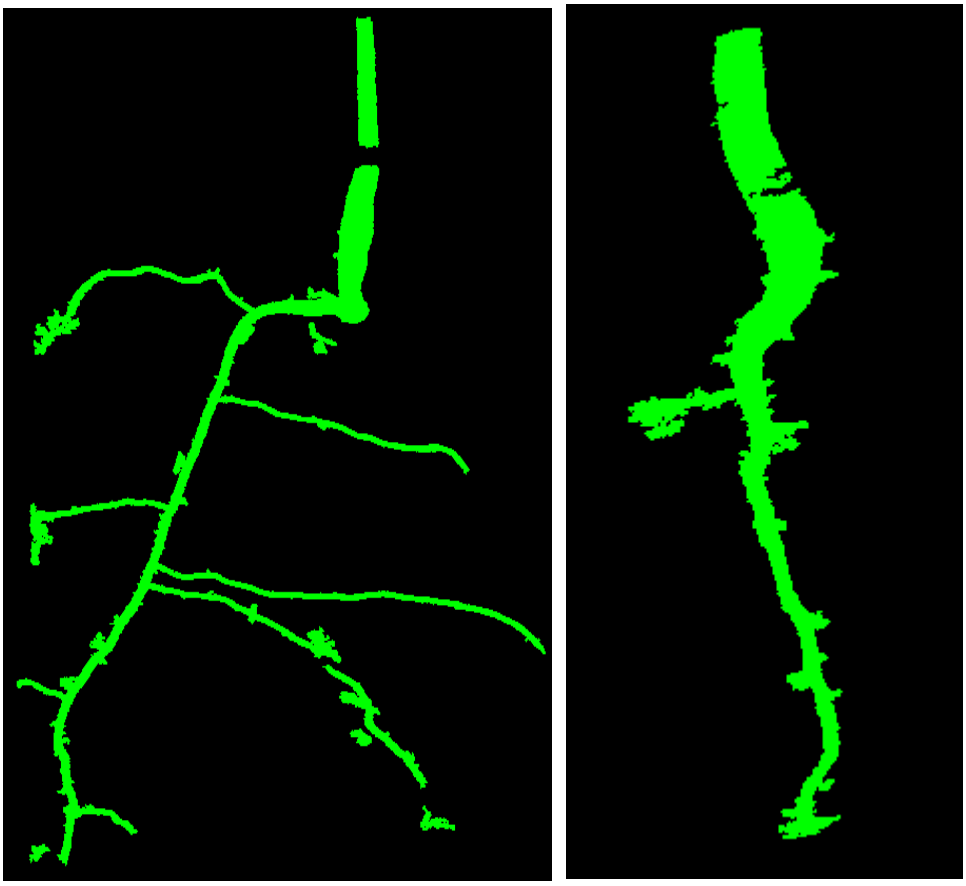


Figure 6.6 Results of 4D assisted Segmentation of the root system of a tomato plant

6.2 Discussion

The RooTrak segmented histogram of the reference image can be used to estimate the root distribution by constraining the root position and width. The following histograms represent just that, with the RooTrak histogram being applied to the whole root system and the skeleton histogram again looking at a single section:

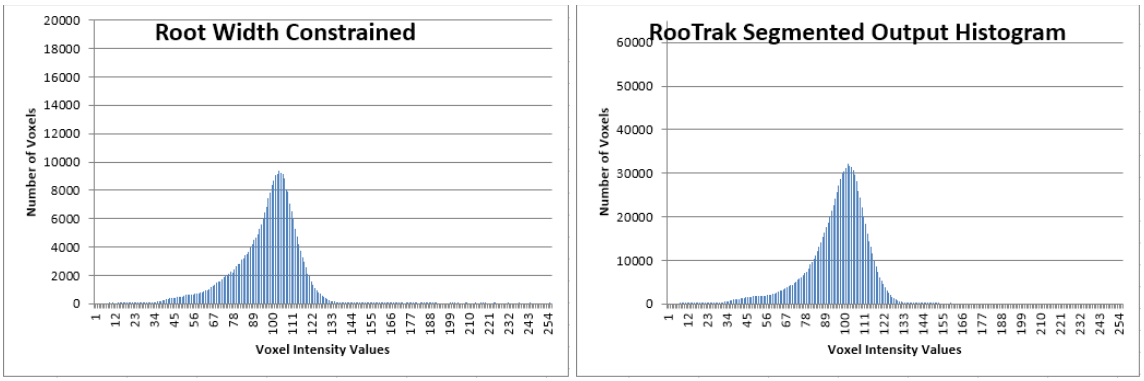


Figure 6.7 Comparison between Root Width Constrained and RooTrak Segmented outputs (scaled up to make them match)

Not surprisingly, the two histograms have a very similar shape (and in both cases there is only one visible peak. More interestingly, we can see that in both cases the bell-shaped curve is non-symmetric. A practical short-cut to deal with this non-standard shaped Gaussian curve is to fit a standard Gaussian distribution and then take different amounts of standard deviations on the left and the right respectively to account for the data observed in real life. There are a number of different reasons why the skewed distribution of intensity values is observed. It may simply be the case of an image artefact i.e. darkened areas of the root system showing up. It could be down to the fundamentals of the CT scanner where the x-ray beam or the detector is more sensitive to lower attenuation readings or it could be that fundamentally the root material bears a different distribution such as the Poisson distribution, for instance. Nevertheless, it is something that a future implementation based on this framework should take into account in order to further refine the quality of the segmented output.

In real terms, when it comes to which approach would have the biggest improvement over the segmented result, undoubtedly that would be the RooTrak segmentation histogram as it provides the best possible ground truth. The real term improvement would come from the fact that the sliding window could be made arbitrarily small and still preserve the shape of

the resulting histogram (no matter how few voxels each sliding window encapsulates, there will always be some root voxels) which would be one way of dealing with the small darkened sections and thus producing a fully connected or near-fully connected segmented output. Arguably, this might not greatly improve the method, as one could interpret this approach as doing what RooTrak does already – looking at the image stack slice by slice in a constrained area of interest and segmenting out the region with the highest probability of being root material - and crucially using the RooTrak output itself to determine the probability.

In practical terms, when dealing with true 4D time-series where there are different reference and sensed image stacks there are two main approaches emerging when calculating the root material intensity distribution. First, make the assumption that the global image stack histogram is near identical in all scans, which may well be the case when dealing with very short temporal resolution of just a few hours, and making sure that the exact same settings for 3D reconstructions and cropping are used. In that case, it would make most sense to calculate the histogram from the reference image as that way it will be known exactly where the root material is and the RooTrak histogram would be most appropriate for that purpose especially if one is interested in deriving the global histogram of the root system. The root width constrained histogram is better suited for the case where one needs to calculate the distribution locally for the sliding window approach as the skeleton allows to segregate individual roots although perhaps a hybrid version of both could be devised for optimal performance where the root skeleton width is used to look at the area around the root for the corresponding sliding window, but only RooTrak segmented voxels are considered.

In cases where no assumption can be made regarding the global histogram distribution, as in the case of low grade temporal 4D series where different treatments may have been applied between consecutive scans, then the method needs to derive the distribution of the intensity values from the sensed image stack by looking at larger windows around the skeleton. If the method were to assume that in all but a very few cases the global rigid 4D registration would make the roots match then this would in itself undermine the fine local registration step as there will be nothing to register, so this makes this assumption self-excluding. In this case, the combination of weighted histogram and multi-peak histogram

distribution would be the best approach. One can also argue that an alternative is to use the segmented RooTrak output itself and for instance dilate it morphologically in order to capture the surrounding roots, this however would bypass the skeleton to an extent and would be somewhat a different method. Another advantage of using the skeleton is that it gives the structure of the root system, making the local adaptive histogram segmentation by the means of the sliding window much easier to implement in a low level programing language such as C++.

In conclusion, the RooTrak histogram approach is best suited for verification purposes. It provides reliable ground truth, whereas for real-world 4D time series the best approach would be a combination of root width constrained histogram and some form of adaptive confidence interval where if the current distribution of the underlying sliding window deviates too much from the global histogram, the search area is widened to search for the root. The method itself doesn't guarantee to generate a fully connected segmented output like RooTrak, however this could be an advantage in some cases. Even if there are areas where the root disappears for a long time, it will still continue to look further down instead of relying on everything being chained down. Thus, the method only intends to complement the RooTrak output and not to generate a better or an independent version of it. Regarding dealing with the missing sections problem beyond using RooTrak as post-processing tool, possibilities range from further refining the sliding window adaptive histogram segmentation using the above findings although fundamentally connected output is not in the DNA of the method the same way it is with RooTrak so it may only have a limited success, performing image pre-processing to deal with issues like beam hardening for instance or other reconstruction artefacts in cases where the segmentation struggle solely as a result of image artefacts or performing a morphological post-processing of the segmented output in the form of dilating and eroding the segmented output in order to close the gaps and then using the original output as a mask in order not to lower the quality of the original segmentation or introduce touching roots again with potentially limited success.

6.3 Conclusions

The preliminary skeleton based segmentation method described in this chapter could be used for the evaluation of 4D time-series of plants where data at only one time points is obtained via a high quality slow CT scan (and thus possible to segment with RooTrak) and the remaining time points are scanned using fast scan mode producing relatively low quality images. The theoretical method and results presented in this chapter have been developed into a prototype version of RootShed which implements a basic skeleton-based segmentation extension, written in C++ with a basic GUI similar to that of RooTh. The purpose of the tool is not to be a completed version or improved version of RooTrak but to act as a proof of concept and to allow the testing and evaluation of the segmentation methods. The results of initial evaluation by registering and segmenting the same image stack with itself and comparing the segmentation to that of a RooTrak output are promising and suggest that further improvements may lead to real-world applications of the proposed methods.

Chapter 7.
Conclusions and Future Work

7.1. Review of Aims and Objectives

This thesis presents a set of methods for the quantification and extraction of root traits from CT images of plants grown in soil columns. Attention focuses on means of skeletonising segmented outputs of root systems, and extracting root system architecture traits from the skeletons obtained. The methods proposed were implemented in a root skeletonisation tool (RooTh) which was subsequently used to evaluate and validate the methods as well as being used for the analysis of real-word biological data produced by experiments carried out at the Hounsfield Facility at The University of Nottingham. The following sections are a review of the original aims and objectives at the start of the project and a discussion of the extent to which they were fulfilled.

7.1.1. Semi-automated skeletonisation tool

RooTh has largely met its goal of providing automated tools for the quantification and phenotyping plant roots imaged using of non-invasive CT scans of soil columns. The original aim was to develop a fully-automated software tool which can work in batch mode, i.e. given a list of dozens or hundreds of image stacks it can run as a process continuously in the background producing a complete description of all image stacks when it finishes. Whilst RooTh requires some user interaction with the GUI in order to produce a morphologically accurate skeletonised description of the root system, the amount of user interaction is minimal. Experience with the semi-automated functionality of RooTh has demonstrated its value for plant scientists.

The original curve-fitting skeletonisation method described in Chapter 3 is almost fully automated apart from initialisation of a small number of parameters (the smoothing parameter or morphological erosion and dilation parameter) which must be set for each individual image stack before the process is run. However, given an experiment with a large number of replicate plants which have undergone identical treatment and growth conditions, it is likely that the same set of parameters would apply equally to all image

stacks thus enabling the root skeletonisation algorithm to be run in batch mode very much like the original aim intended. Due to the relatively large number of false-positive root tips generated by the root tip detection algorithm described in Chapter 3, a post-processing user-guided step may be necessary to remove all resulting false-positive rootlets from the skeleton descriptions, however, parallel processing in batch mode will still be possible.

The ACM-based skeletonisation algorithm which was the topic of Chapter 4 requires the root system output generated by the curve-fitting steps to be fractalised, which is the main semi-automated user-driven step in RooTh as described in further detail in section 3.1.2.8. Since processing each root system required supervision and input from the user, the user becomes the main bottleneck in the algorithm and as such parallelisation on this scale is no longer possible. Some global traits, such as total root length, may however be obtained without the fractalisation step.

7.1.2. Morphologically accurate approximation of the skeletal ground truth

A naturally-flowing aim of software tools in the field of computer vision is to produce output that is as close as possible to some ground truth measurement of the scene. Any root skeletonisation tool as such aims to generate skeleton and associated trait measurements that are an accurate reflection or close approximation of the skeletal ground truth of the underlying root system. The specific criteria used in the development of RooTh to evaluate how accurate the skeleton descriptions produced by the tool score compared to an ideal ground truth model were discussed in details in section 3.1. Overall, the ACM-based skeletonisation method was shown to be more accurate when compared to the curve-fitting algorithm. Both methods were evaluated using real-world biological data consisting of segmented root image stacks of various plant species as discussed in the results and discussion sections of Chapters 3 and 4 respectively.

Another important aim of the output produced by RooTh was to provide local description of the underlying root system as opposed to global one in the form of numerical root traits of individual roots from base to root tip. The root fractalisation algorithm and corresponding manual annotation interface presented in Chapter 3 allow for the complete fractalisation of the root system by encoding individual roots as separate entities in the hierarchical

description of the root system. Automatic fractalisation of root architectures, however, remains an unsolved problem.

7.1.3. Root image analysis pipeline integration

The original objective for RooTh was to be developed as a plug-in for RooTrak [31]. This would have enabled a user of RooTrak to seamlessly process a soil column and directly produce a list of trait measurements and retrieve the root system description. Integrating RooTh as part of RooTrak would have also allowed users to take advantage of internal variables and data structure representations which may have aided the skeletonisation process. However, throughout the development of the project, the outline of the CT image analysis pipeline became more apparent, starting with the preparation of a biological experiment, through scanning soil columns and reconstructing the 3D volumes, to segmenting and skeletonising the root system and using the resulting traits for statistical analysis of the phenomic and genomic traits [66]. At every stage of the process multiple tools could be used to perform the task and it became necessary that RooTh was compatible with any generic root segmentation software as well as generating the resulting root traits and root system description in an interoperable format which can be exchanged with other similar root analysis tools.

RooTh was designed to read its input from a sequence of raw uncompressed black and white images highlighting the distribution of root material within the soil column. It is common to use such binary images to represent masks or segmented data which makes this format suitable for reading input from tools other than RooTrak, such as manual segmentation performed using VG Studio Max. It can be expected that the vast majority of 3D root segmentation tools that have been developed or will be developed in future will easily support such a format. Whilst importing segmented data from plain binary images allows great compatibility between RooTh and any third party segmentation tool, any future utilisation of RooTh with other software should take into account the assumptions RooTh makes about the nature of the segmented input. These are described in detail in Chapter 3. In particular, its reliance on the segmented root system being fully connected as is the output produced by RooTrak and RooTh's particular sensitivity to false positives i.e. fragments and artefacts marked as root material which are really not. If RooTh encounters

such ROIs it will default to skeletonising them as if they were roots. It will therefore be imperative for any future segmentation tool which uses RooTh for its skeletonisation and traits analysis to also minimise the number of false positive roots present in its output.

In some cases the skeletonisation computed by RooTh may be the end result, but it is more likely that it is intended for use by other tools further down the pipeline which are tasked with performing some form of statistical analysis. The complete fractalised description of the root system is encoded using the RSML format [45], making it compatible with a large number of existing root skeletonisation tools some of which were reviewed in Chapter 2. This high-level descriptive data is invaluable for simulation research into the structure and dynamics of the root systems of plants and the fractalised local quantitative traits provide higher quality data for QTL analysis than existing global trait quantification tools. The RSML output of a large scale experiment could be used to parameterise a root growth simulation such as that generated by SimRoot [44] and allow plant scientists to access new levels of detail when looking at trait expression, even beyond that allowed by current observation technology such as CT scanners.

7.2.Future improvements and applications

Future improvement of the methods used in RooTh could focus on improving the root tip detection algorithm and potentially removing the need to use heuristics. Exciting opportunities are arising in the recently emerged field of Convolutional Neural Networks (CNNs) [67]. Once a sufficiently large number of root CT scans are imaged and analysed in facilities such as the Hounsfield Facility, it may become possible for artificial intelligent techniques to be deployed to automatically learn to recognise the location of root tips within a raw CT volume with high degree of accuracy [68].

Another potential application of deep learning towards improving the accuracy of the RooTh's skeleton output and removing the need to manual annotation of the input data is the problem of root system fractalisation. If a sufficiently large amount of annotated data becomes available, a neural network could be trained to automatically classify individual root branches as belonging to primary and lateral roots and in turn this classification metadata could be used to determine the branching at every point of the root system hierarchy by making local decisions about which way roots are connected.

This thesis also presents a proposed method for the automatic registration and quantification of dynamic root traits from 4D time-series CT data as well as a proposed method for the segmentation of root systems using knowledge from different time-points. The proposed segmentation method does not provide a solution for locating the root tips of in the segmented data. Future work is needed to automatically detect the ends of roots in images captured at earlier time-points when an RSML description of an older root is used.

Bibliography

[1] R. Hopfenberg, 'Human Carrying Capacity Is Determined by Food Availability', *Popul. Environ.*, vol. 25, no. 2, pp. 109–117, Nov. 2003.

[2] Joel Mokyr, 'Great Famine | History, Causes, & Facts', *Encyclopedia Britannica*. [Online]. Available: https://www.britannica.com/event/Great-Famine-Irish-history. [Accessed: 29-Jun-2018].

[3] 'World Population', United Nations Department of Economic and Social Affairs/Population Division, New York, 2004. [Online]. Available: http://www.un.org/esa/population/publications/longrange2/WorldPop2300final.pdf

[4] M. A. Hanjra and M. E. Qureshi, 'Global water crisis and future food security in an era of climate change', *Food Policy*, vol. 35, no. 5, pp. 365–377, Oct. 2010.

[5] William M.Denevan, '2 Prehistoric agricultural methods as models for sustainability', *Adv. Plant Biol.*, vol. Volume 11, pp. 21–43, 1995.

[6] Leon Hesser, *The Man Who Fed the World: Nobel Peace Prize Laureate Norman Borlang and His Battle to End World Hunger*. 2006.

[7] P. B. R. Hazell, *The Asian Green Revolution*. Intl Food Policy Res Inst, 2009.

[8] D. G. Dalrymple, 'The Development and Adoption of High-Yielding Varieties of Wheat and Rice in Developing Countries', *Am. J. Agric. Econ.*, vol. 67, no. 5, pp. 1067–1073, 1985.

[9] B. Khanna, 'What are the advantages of high yielding varieties of crops?' [Online]. Available: http://www.preservearticles.com/201012312124/advantages-of-high-yielding-verities-of-crops.html. [Accessed: 29-Jun-2018].

[10] C. S. Gasser and R. T. Fraley, 'Genetically Engineering Plants for Crop Improvement', *Science*, vol. 244, no. 4910, pp. 1293–1299, Jun. 1989.

[11] G. Conway, *The Doubly Green Revolution: Food for All in the Twenty-First Century*, 1 edition. Ithaca, N.Y: Comstock Publishing Associates, 1999.

[12] *NIAB: The science behind superwheat*. 2013. [Online]. Available: https://www.youtube.com/watch?v=M7bN2C1Pjs0

[13] 'News and Events - NEWS: Breakthough in wheat breeding science offers greater yields', 13-May-2013. [Online]. Available: http://www.niab.com/news_and_events/article/282. [Accessed: 31-Jul-2018].

[14] S. J. Mooney, T. P. Pridmore, J. Helliwell, and M. J. Bennett, 'Developing X-ray Computed Tomography to non-invasively image 3-D root systems architecture in soil', *Plant Soil*, vol. 352, no. 1–2, pp. 1–22, Mar. 2012.

[15] M. J. Bennett, 'Malcolm Bennett on UK Plant Science', *The Centre for Plant Integrative Biology*. [Online]. Available: https://www.cpib.ac.uk/outreach/public-engagement/media/garnet-mjb-youtube/. [Accessed: 29-Oct-2016].

[16] E. G. Jobbágy and R. B. Jackson, 'The distribution of soil nutrients with depth: Global patterns and the imprint of plants', p. 28.

[17] B. Ehdaie *et al.*, 'Root System Size Influences Water-Nutrient Uptake and Nitrate Leaching Potential in Wheat', *J. Agron. Crop Sci.*, vol. 196, no. 6, pp. 455–466.

[18] J. P. Lynch and K. M. Brown, 'Topsoil foraging – an architectural adaptation of plants to low phosphorus availability', *Plant Soil*, vol. 237, no. 2, pp. 225–237, Dec. 2001.

[19] S. Paula and J. G. Pausas, 'Root traits explain different foraging strategies between resprouting life histories', *Oecologia*, vol. 165, no. 2, pp. 321–331, Feb. 2011.

[20] Y. Ren *et al.*, 'Major quantitative trait loci for seminal root morphology of wheat seedlings', *Mol. Breed.*, vol. 30, no. 1, pp. 139–148, Jun. 2012.

[21] J. Kumar, A. Pratap, and S. Kumar, 'Plant Phenomics: An Overview', in *Phenomics in Crop Plants: Trends, Options and Limitations*, Springer, New Delhi, 2015, pp. 1–10.

[22] ‘Introduction to Common Root Phenotyping Methods’. [Online]. Available: http://www.plantstress.com/methods/Root%20Methods%20PORTAL/Root%20methods.htm. [Accessed: 29-Jun-2018].
[23] S. Zappala *et al.*, ‘Effects of X-Ray Dose On Rhizosphere Studies Using X-Ray Computed Tomography’, *PLOS ONE*, vol. 8, no. 6, p. e67250, Jun. 2013.
[24] J. L. Chotte, J. Y. Laurent, and J. P. Rossi, ‘A modified hydropneumo-elutriation apparatus for quantitative root separation from large soil core samples’, *Commun. Soil Sci. Plant Anal.*, vol. 26, no. 15–16, pp. 2703–2709, Sep. 1995.
[25] J. C. O’Toole and Soemartono, ‘Evaluation of a simple technique for characterizing rice root systems in relation to drought resistance’, *Euphytica*, vol. 30, no. 2, pp. 283–290, Jun. 1981.
[26] G. Zeng, S. T. Birchfield, and C. E. Wells, ‘Automatic discrimination of fine roots in minirhizotron images’, *New Phytol.*, vol. 177, no. 2, pp. 549–557, Jan. 2008.
[27] ‘WinRHIZO’, 2013. .
[28] L. G. Tumlinson, H. Liu, W. K. Silk, and J. W. Hopmans, ‘Thermal Neutron Computed Tomography of Soil Water and Plant Roots’, *Soil Sci. Soc. Am. J.*, vol. 72, no. 5, pp. 1234–1242, Sep. 2008.
[29] Vgs. MAX, ‘VGSTUDIO MAX 3.0 − STATE OF THE ART FOR CT DATA’. [Online]. Available: https://www.ita-polska.com.pl/upload/documents/VGStudioMAX_30_en0.pdf.
[30] P. . Sahoo, S. Soltani, and A. K. . Wong, ‘A survey of thresholding techniques’, *Comput. Vis. Graph. Image Process.*, vol. 41, no. 2, pp. 233–260, Feb. 1988.
[31] S. Mairhofer *et al.*, ‘RooTrak: Automated Recovery of Three-Dimensional Plant Root Architecture in Soil from X-Ray Microcomputed Tomography Images Using Visual Tracking’, *PLANT Physiol.*, vol. 158, no. 2, pp. 561–569, Feb. 2012.
[32] D. Leitner, B. Felderer, P. Vontobel, and A. Schnepf, ‘Recovering Root System Traits Using Image Analysis Exemplified by Two-Dimensional Neutron Radiography Images of Lupine’, *PLANT Physiol.*, vol. 164, no. 1, pp. 24–35, Jan. 2014.
[33] G. Lobet, L. Pagès, and X. Draye, ‘A Novel Image-Analysis Toolbox Enabling Quantitative Analysis of Root System Architecture’, *Plant Physiol.*, vol. 157, no. 1, pp. 29–39, Sep. 2011.
[34] D. Ristova, U. Rosas, G. Krouk, S. Ruffel, K. D. Birnbaum, and G. M. Coruzzi, ‘RootScape: A landmark-based system for rapid screening of root architecture in Arabidopsis thaliana’, *Plant Physiol.*, p. pp.112.210872, Jan. 2013.
[35] M. P. Pound, A. P. French, J. A. Atkinson, D. M. Wells, M. J. Bennett, and T. Pridmore, ‘RootNav: Navigating Images of Complex Root Architectures’, *Plant Physiol.*, vol. 162, no. 4, pp. 1802–1814, Aug. 2013.
[36] H. Blum, ‘A Transformation For Extracting New Descriptors of Shape’, 1964.
[37] R. Clark *et al.*, ‘3-Dimensional Root Phenotyping with a Novel Imaging and Software Platform’, *Plant Physiol.*, p. pp.110.169102, Jan. 2011.
[38] J. Wu *et al.*, ‘RhizoChamber-Monitor: a robotic platform and software enabling characterization of root growth’, *Plant Methods*, vol. 14, no. 1, p. 44, Jun. 2018.
[39] O. Symonova, C. N. Topp, and H. Edelsbrunner, ‘DynamicRoots: A Software Platform for the Reconstruction and Analysis of Growing Plant Roots’, *PLOS ONE*, vol. 10, no. 6, p. e0127657, Jun. 2015.
[40] ‘Morphology - Skeletonization/Medial Axis Transform’. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/HIPR2/skeleton.htm. [Accessed: 29-Jun-2018].
[41] ‘Skeletonization’. [Online]. Available: http://www.inf.u-szeged.hu/~palagyi/skel/skel.html. [Accessed: 29-Jun-2018].
[42] R. C. Gonzalez and R. E. Woods, *Digital Image Processing: International Edition*, 3 edition. Upper Saddle River, NJ: Pearson, 2008.
[43] S. R. Tracy, J. A. Roberts, C. R. Black, A. McNeill, R. Davidson, and S. J. Mooney, ‘The X-factor: visualizing undisturbed root architecture in soils using X-ray computed tomography’, *J. Exp. Bot.*, vol. 61, no. 2, pp. 311–313, Jan. 2010.

[44] J. A. Postma *et al.*, 'OpenSimRoot: widening the scope and application of root architectural models', *New Phytol.*, vol. 215, no. 3, pp. 1274–1286.
[45] G. Lobet *et al.*, 'Root System Markup Language: toward a unified root architecture description language', *Plant Physiol.*, p. pp.114.253625, Jan. 2015.
[46] S. Mairhofer *et al.*, 'Recovering complete plant root system architectures from soil via X-ray µ-Computed Tomography', *Plant Methods*, vol. 9, p. 8, Mar. 2013.
[47] S. Mairhofer, C. Sturrock, D. M. Wells, M. J. Bennett, S. J. Mooney, and T. P. Pridmore, 'On the evaluation of methods for the recovery of plant root systems from X-ray computed tomography images', *Funct. Plant Biol.*, vol. 42, no. 5, pp. 460–470, May 2015.
[48] D. Silver, N. D. Cornea, and P. Min, 'Curve-Skeleton Properties, Applications, and Algorithms', *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 3, pp. 530–548, 2007.
[49] R. W. Schafer, 'What Is a Savitzky-Golay Filter? [Lecture Notes]', *IEEE Signal Process. Mag.*, vol. 28, no. 4, pp. 111–117, Jul. 2011.
[50] 'Cross Validation'. [Online]. Available: http://www.cs.cmu.edu/~schneide/tut5/node42.html. [Accessed: 29-Jun-2018].
[51] Y. Waisel, A. Eshel, T. Beeckman, and U. Kafkafi, *Plant Roots: The Hidden Half, Third Edition*. CRC Press, 2002.
[52] M. Kass, A. Witkin, and D. Terzopoulos, 'Snakes: Active contour models', *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, Jan. 1988.
[53] J. Ivins and J. Porrill, 'EVERYTHING YOU ALWAYS WANTED TO KNOW ABOUT SNAKES (BUT WERE AFRAID TO ASK)', p. 36.
[54] A. Benson and D. J. Evans, 'A normalised algorithm for the solution of symmetric general five-term linear systems of semi-bandwidth M', *J. Comput. Appl. Math.*, vol. 5, no. 4, pp. 299–304, Dec. 1979.
[55] B. Vöcking *et al.*, 'Simulated Annealing', in *Algorithms Unplugged*, Springer Science & Business Media, 2010, pp. 393–401.
[56] R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann, 1987.
[57] A. A. Goshtasby, *2-D and 3-D Image Registration: for Medical, Remote Sensing, and Industrial Applications*. John Wiley & Sons, 2005.
[58] M. P. Pound, A. P. French, D. M. Wells, M. J. Bennett, and T. P. Pridmore, 'CellSeT: Novel Software to Extract and Analyze Structured Networks of Plant Cells from Confocal Images', *Plant Cell*, vol. 24, no. 4, pp. 1353–1361, Apr. 2012.
[59] C. Zahn, 'An Algorithm for Noisy Template Matching', 1974.
[60] B. Vöcking *et al.*, 'Minimum Spanning Trees (Sometimes Greed Pays Off...)', in *Algorithms Unplugged*, Springer Science & Business Media, 2010, pp. 325–332.
[61] M. A. Fischler and R. C. Bolles, 'Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography', *Commun ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.
[62] P. Thevenaz, U. E. Ruttimann, and M. Unser, 'A pyramid approach to subpixel registration based on intensity', *IEEE Trans. Image Process.*, vol. 7, no. 1, pp. 27–41, Jan. 1998.
[63] 'TurboReg'. [Online]. Available: http://bigwww.epfl.ch/thevenaz/turboreg/. [Accessed: 29-Jun-2018].
[64] E. W. Cheney and W. A. Light, *A Course in Approximation Theory*. American Mathematical Soc., 2009.
[65] M. E. Dunn and S. H. Joseph, 'Processing poor quality line drawings by local estimation of noise', in *Pattern Recognition*, 1988, pp. 153–162.
[66] S. Mairhofer *et al.*, 'X-Ray Computed Tomography of Crop Plant Root Systems Grown in Soil', *Curr. Protoc. Plant Biol.*, vol. 2, no. 4, pp. 270–286.
[67] H. H. Aghdam and E. J. Heravi, *Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification*. Springer, 2017.

[68] P. Kumar, C. Huang, J. Cai, and S. J. Miklavcic, ‘Root phenotyping by root tip detection and classification through statistical learning’, *Plant Soil*, vol. 380, no. 1–2, pp. 193–209, Jul. 2014.