

Reeb Graph Based Image Representation for Phenotyping of Plants

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Computergraphik und Digitale Bildverarbeitung

eingereicht von

Ines Janusch

Matrikelnummer 0525109

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: o.Univ.-Prof. Dipl.-Ing. Dr. Walter G. Kropatsch

Wien, 15.03.2014

(Unterschrift Verfasserin)

(Unterschrift Betreuung)

Reeb Graph Based Image Representation for Phenotyping of Plants

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Computergraphics and Digital Image Processing

by

Ines Janusch

Registration Number 0525109

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: o.Univ.-Prof. Dipl.-Ing. Dr. Walter G. Kropatsch

Vienna, 15.03.2014

(Signature of Author)

(Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Ines Janusch
Reithofferplatz 12/11, 1150 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Verfasserin)

Acknowledgements

Optional acknowledgements may be inserted here.

Abstract

While genotypes are defined as the set of genes an organism holds, its phenotype is defined as the set of its observable characteristics. To determine the correlation of genotype and phenotype or how a phenotype is affected by environmental conditions, an evaluation on large datasets is needed. An automatic analysis of image data and extraction of characteristics allows for large-scale evaluations.

This thesis presents a comparison of two types of graph-based image representations: medial axis transformation and Reeb graphs and evaluates the feasibility of using these representations in image based plant phenotyping.

A presegmented binary image of roots (of the plant *Arabidopsis thaliana*) is the basis for generating the well-known medial axis and the Reeb graphs. For phenotyping of plants their root structure is analyzed. The main characteristics used here are branching points, branch endings as well as the length and width of individual branches. These characteristics are captured by the presented graph representations. For the computation of the Reeb graphs two different Morse functions are used: height function and geodesic distance.

As the roots are pictured as 2D image data, the projection of a 3D structure to a 2D space might result in an overlap of branches in the image. One major advantage, when analyzing roots based on Reeb graphs, is posed by the ability to immediately distinguish between branching points and overlaps in the root structure as the overlap introduces a cycle and thereby a certain type of node (saddle - merge) in the Reeb graph. This differentiation is not as easily possible by a medial axis representation or by an analysis solely based on contours.

In order to use the advantages of different representations and the characteristics provided by them, a possibility to combine different graph representations of one root image is needed. Therefore the equality of graphs is evaluated. This thesis shows that all three representations of a root are either isomorphic graphs or isomorphic subgraphs. For isomorphic graphs the characteristics, the nodes are attributed with, such as length or width, can be directly combined for matching nodes. For isomorphic subgraphs only the attributes of the matching subgraphs can be combined.

Kurzfassung

Ähnlich zu den, als Menge der Gene eines Organismus definierten Genotypen, sind Phänotypen definiert als die Menge der beobachtbaren Eigenschaften dieses Organismus. Um den Zusammenhang von Genotypen und Phänotypen sowie den Einfluss von Umweltbedingungen auf den Phänotyp zu bestimmen, werden große Datensätze evaluiert. Eine automatische Analyse von Bilddaten und Extraktion von Charakteristiken ermöglichen die Auswertung dieser großangelegten Studien.

Diese Diplomarbeit widmet sich dem Vergleich zweier graphbasierter Bild-Repräsentation: Medial Axis Transformation und Reeb Graphen und analysiert die Einsatzmöglichkeit dieser Abbildungen in der bildbasierten Pflanzen Phänotypisierung.

Ein segmentiertes Binärbild der Wurzeln (der Pflanze *Arabidopsis thaliana*) bildet die Grundlage zur Berechnung der bekannten Medial Axis und der Reeb Graphen. Zur Phänotypisierung von Pflanzen wird ihre Struktur analysiert. Die wichtigsten Charakteristiken dabei sind: Verzweigungen, Enden von Zweigen, sowie die Länge bzw. Dicke der einzelnen Zweige. Diese Eigenschaften werden von den verwendeten Graph-Repräsentationen erfasst. Zur Berechnung der beiden Reeb Graphen werden zwei unterschiedliche Morse Funktionen verwendet: die Höhenfunktion sowie die geodätische Distanz.

Da die Pflanzen als 2D Datensatz abgebildet werden, findet eine Projektion der 3D Wurzelstruktur in den 2D Raum des Bildes statt. Diese Projektion kann in Überlappungen einzelner Wurzelzweige enden. Ein bedeutender Vorteil der Reeb Graph Repräsentationen ist durch die Möglichkeit, Verzweigungspunkte sofort von Überlappungen unterscheiden zu können, gegeben. Dies beruht auf der Eigenschaft der Reeb Graphen, dass durch die Überlappungen der Wurzeln im Bild, Zyklen in der Graphstruktur entstehen. Diese Zyklen erzeugen einen Knoten vom Typ saddle (merge) im Graph. Eine Unterscheidung zwischen Überlappungen und Verzweigungen ist für eine Darstellung basierend auf der Medial Axis oder einer Analyse der Wurzelstruktur anhand der Kontur nicht so leicht möglich.

Um die Vorteile verschiedener Darstellungen und die entsprechenden Charakteristiken zu verwenden, wird eine Kombination der unterschiedlichen graphbasierten Repräsentationen benötigt. Dazu muss die Gleichwertigkeit der Graphen bewertet werden. Im Rahmen dieser Diplomarbeit zeigt sich, dass die Graphen aller drei Darstellungen entweder isomorph sind oder isomorphe Teilgraphen enthalten. Für isomorphe Graphen können Attribute für jeweils übereinstimmender Knoten direkt für alle Knoten kombiniert werden. Für isomorphe Teilgraphen ist dies nur für die Knoten der übereinstimmenden Teilgraphen möglich.

Contents

1	Introduction	1
1.1	Problem Statement	2
1.2	Aim of the Work	2
1.3	Contribution	3
1.4	Structure of the Thesis	3
2	Graph based Shape Representation	5
2.1	Graphs and Skeletons	5
2.2	Medial Axis Transformation	9
2.3	Morse Theory and Reeb Graphs	10
3	State of the Art	15
3.1	Image Based Plant Phenotyping	15
3.2	Graph Based Shape Description	19
4	Dataset and Pre-Processing	23
4.1	Image Acquisition	23
4.2	Pre-Processing	25
5	Graphs on Root Images	27
5.1	Computation of Medial Axis Graph	29
5.2	Computation of Reeb Graph (Height Function)	30
5.3	Computation of Reeb Graph (Geodesic Distance)	33
5.4	Improvements on the Graphs	35
6	Results	37
6.1	Pruning Effect	39
6.2	Overlapping Branches	42
6.3	Comparison: Root System Analyser	43
7	A Normalized Root Representation	47
7.1	Equality of Graphs	47
7.2	Combined Root Representation	51

8 Conclusion	53
List of Figures	55
List of Tables	57
Bibliography	59

Introduction

The problem of shape analysis and representation within the context of image based plant phenotyping is addressed in this thesis.

A first definition of phenotypes was given by Johannsen in 1911 in [19]: “All “types” of organisms, distinguishable by direct inspection or only by finer methods of measuring or description, may be characterized as “phenotypes”.” While an organism’s genotype is defined as the set of genes this organism holds, its phenotype is defined by its observable characteristics (from Greek: *phainein* = to show).

Gregor Mendel first introduced the concept of dominance, which stated that some inherited traits were dominant over others as a heterozygote (genotype consisting of two different alleles) offspring could show the same phenotype as its homozygote (genotype consisting of two identical alleles) parent. However, the relationship between genotypes and phenotypes proved to be more complex than this simple dominant pattern described by Mendel [27].

Moreover, while the phenotype of an organism is based on its genes, it is just as well influenced by environmental conditions. While the genome of certain plants or animals (so called model organism) is completely characterized [33], their phenotype (the full set of phenotypes of an individual) can never be completely characterized as phenotypes vary strongly [17].

Apart from this application in molecular biology, this thesis is set within the field of image processing, more precisely within the area of topological image analysis and shape representation. The following keywords describe the overall context:

- Graph Theory
- Topological Image Analysis
- Shape Extraction, Representation and Comparison

1.1 Problem Statement

To address this major questions in biology, how genotypes translate into phenotypes, the ability to phenotype a large number of individuals is needed. Therefore an efficient extraction and representation of characteristics is needed.

For phenotyping of plants their root development, the architecture of their root systems and thereby root characteristics such as branches and branch endings are analyzed. The root of the small plant *Arabidopsis thaliana* is excellently suitable for large-scale non-invasive phenotyping because it can be grown on transparent media in large numbers and its projections of the young root essentially capture all the important biological features at the organ level. For analysis of the plant development, the plants are imaged on several successive days of their growth cycle.

The root characteristics used in plant phenotyping, for example branching points or length of branches, can be captured and described by a graph based shape representation.

Branches in the root structure depict a change in the topology of the structure. Reeb graphs describe these changes in topology in the represented structure. As the 3D structure of the root is projected to the 2D image space, overlaps of branches may occur in the root images. These overlaps need to be distinguished from branching points. Here an advantage of Reeb graphs is that such overlaps can be immediately detected, as they introduce a particular type of node in the graph. Reeb graphs are based on Morse theory and analyze the image content according to a function (Morse function). An important question in this context is to determine how much an where two or more functions defined on the same model differ [4]. A normalized graph representation overcomes these differences of various representations and allows to combine the advantages of multiple representations.

In the context of plant phenotyping, a normalized representation is needed to compare plants or their development based on various characteristics and thereby various representations.

1.2 Aim of the Work

In order to simplify the examination of root characteristics and enable an efficient comparison of roots, a representation of imaged root data by Reeb graphs is introduced in this thesis.

Reeb graphs capture the topology of the represented structure - in this case the locations of branches and branch endings of the roots - and form a skeletal graph representation of the underlying image data in this way. The computation of the Reeb graphs on the root images is based on a Morse function. For the root images a Reeb graph representation based on the height function and the geodesic distance is presented. The well known medial axis transformation was applied for a third representation. Based on these three representation the following characteristics of the roots can be acquired:

- number of branches
- number of branch endings
- length of branches

- width of branches
- branching pattern (according to a main stem)

To allow for an efficient comparison of different plants or of one plant in different stages of its development, a normalized representation is defined.

Graphs provide a representation that allows to compare and match shapes. However, for such comparisons it is usually assumed that all graphs were computed based on the same method. In order to combine characteristics based on all three mentioned graph representations, to obtain the final normalized representation, the equality respectively similarity of representations based on different functions is measured.

1.3 Contribution

Based on three graph based shape descriptors an approach to represent root structures is described in this thesis. A popular shape descriptor for branched structures such as roots, is the medial axis transformation. In this thesis a representation of branched structures based on Reeb graphs is introduced. In order to combine characteristics of different representations, the equality respectively the difference between various types of representations is evaluated. A definition for the comparison and matching of two different graph representations forms a central part of this work. The derived root representations are compared to the representations computed with the recently published tool “Root System Analyser” [24].

1.4 Structure of the Thesis

An overview on graph based shape representations in general and the theoretical background particularly needed for the introduction of Reeb graphs are given in Chapter 2. Based on this knowledge the state of the art regarding (Reeb) graph based representations as well as the state of the art in image based plant phenotyping are discussed in Chapter 3. Chapter 4 presents the dataset used within the scope of this thesis. Details on the presented approach for Reeb graph based root representations are given in Chapter 5 while results are discussed in Chapter 6. A comparison with the recently published tool “Root System Analyser” [24] is given in Section 6.3. The possibility of a combination of the different representations is discussed in Chapter 7. Chapter 8 concludes this diploma thesis and gives an outlook to future work.

Graph based Shape Representation

2.1 Graphs and Skeletons

In shape representation a skeleton is a thinned version of a shape with equal distance to the boundaries of the shape. Morphological skeletons, well known as medial axis, are the most common skeletons. Sometimes the terms skeleton and medial axis are used interchangeably as synonyms. However, Figure 2.1 shows a comparison of two different types of skeletons. Figure 2.1a shows an example for a medial axis skeleton, while Figure 2.1b shows another type of skeleton: a straight line skeleton.

A graph G is a tuple (V, E) where V is the set of vertices (nodes) of the graph and E the set of edges connecting the vertices in the graph. Edges in a graph may be directed or undirected. A digital image is formed by a set of pixels. A graph on such an image can be defined by map-

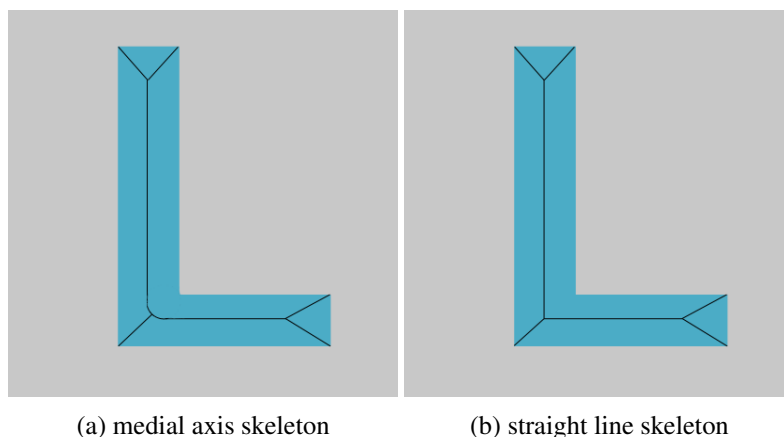


Figure 2.1: Shape representation: skeleton

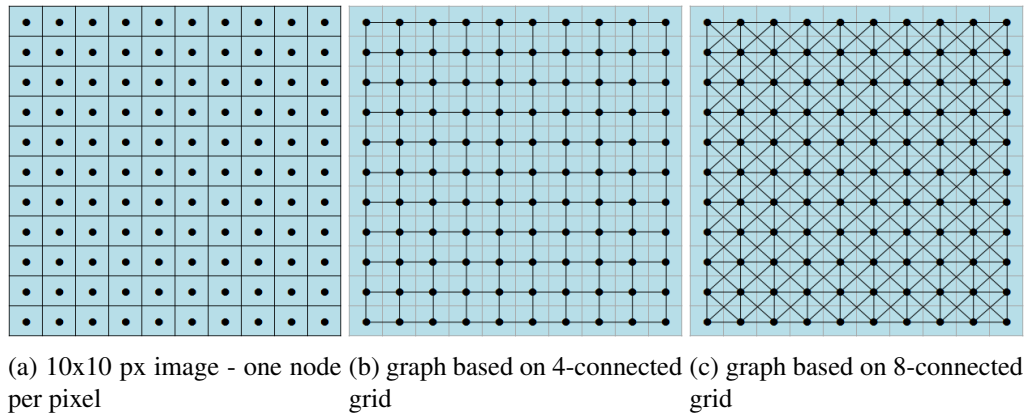
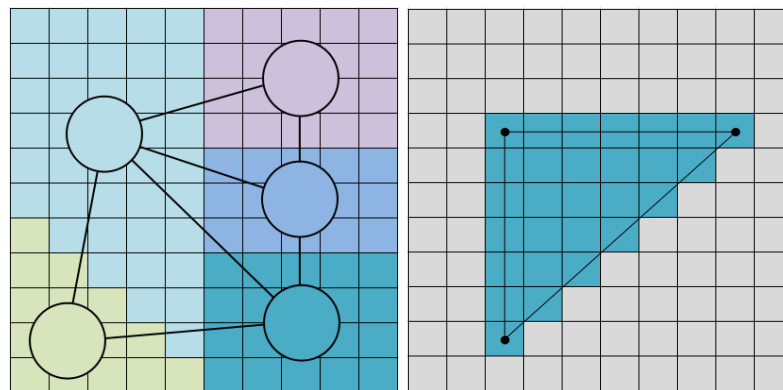


Figure 2.2: Graphs on image pixels.

ping a node to each pixel and connecting nodes of neighboring pixels with edges. Figure 2.2 shows an example for such a graph assuming a 4-connected pixel grid and an 8-connected one.

For graph-based image representations nodes may not be assigned to every pixel in the graph. Pixels are rather grouped and each group is represented by a single node in the graph. The arrangement of pixels into groups can be based on similarity of an assigned label or value as for example their color-value. An example for such a graph is shown in Figure 2.3a. This graph is a region-adjacency-graph (RAG): each region is represented by one node, the edges represent adjacency between regions. In such a way graphs can for example be used in segmentation approaches.

For the description of shapes, nodes are placed at characteristic positions of a shape and connected according to it. The triangle image in Figure 2.3b is represented by 3 nodes at the vertices of the triangle and edges according to the triangle edges.



(a) RAG - nodes represent regions, edges their adjacency
 (b) representation of shape - nodes describe characteristic locations

Figure 2.3: Two examples for graph based image representation.

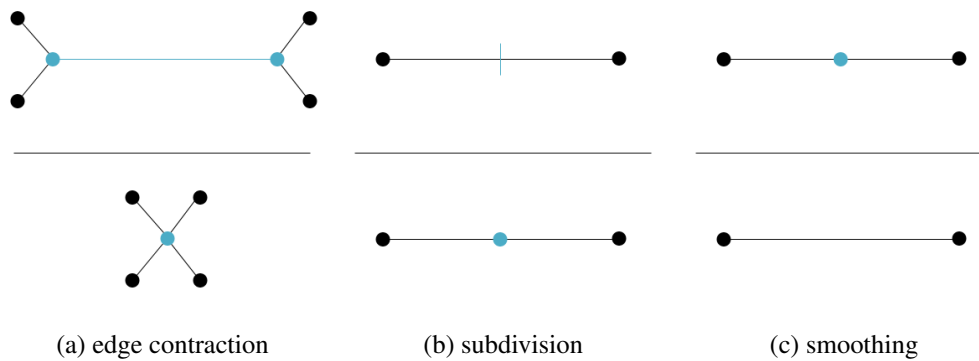


Figure 2.4: Operations on graphs. The top half shows the original graphs with changing parts labeled blue, the bottom half of the images show the altered graphs.

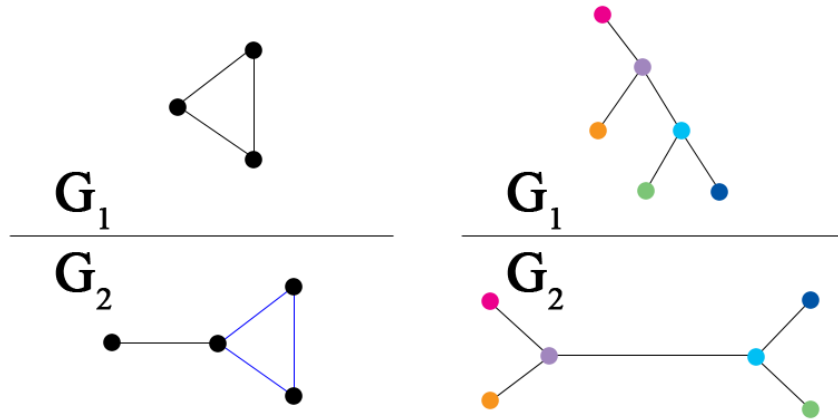
Operations on graphs

To remove or insert vertices or edges in a graph the following operations are defined (Figure 2.4 shows examples):

1. Edge contraction:
removes an edge from the graph by merging the two vertices that were connected by the edge (see Figure 2.4a).
2. Subdivision:
divides an edge into two edges by introducing a new vertex that is adjacent to the two vertices of the original edge (see Figure 2.4b).
3. Smoothing:
removes a vertex from the graph by removing the edges connecting it to other vertices and establishing edges between these vertices (see Figure 2.4c). This operation is only possible for nodes of degree 2 (nodes adjacent to two other nodes).

Subgraphs and Graph Isomorphism

- Graph isomorphism:
Graph isomorphism is defined as a bijection between the vertex sets of two graphs G_1 and G_2 . The bijective function $f : G_1 \mapsto G_2$ is called isomorphism [6]. The two graphs in Figure 2.5b are isomorphic graphs, corresponding nodes are colored the same way.
- Subgraphs:
The vertex set of a subgraph $G' = (V', E')$ of a graph $G = (V, E)$ is a subset of of the vertex set of G . The edge set of the subgraph G' is the subset of the edge set of G restricted to the vertices in V' : $V' \subseteq V$ and $E' = E \cap (V' \times V')$ [6]. Figure 2.5a shows graph G_1 as a subgraph of G_2 , the corresponding edges of the subgraph are colored in blue.



(a) G_1 is a subgraph of G_2 (edges of the subgraph are marked in blue) (b) G_1 and G_2 are isomorphic graphs

Figure 2.5: Definition of subgraphs and graph isomorphism

- Isomorphic subgraphs:

A subgraph isomorphism from G to G' is defined as the injective function on the two graphs $G = (V, E)$ and $G' = (V', E')$: $f : G \mapsto G'$ if there exists a subgraph $S \subseteq G'$ so that f is a graph isomorphism from G to S [6]. In Figure 2.5a there is a subgraph isomorphism from G_1 to G_2 .

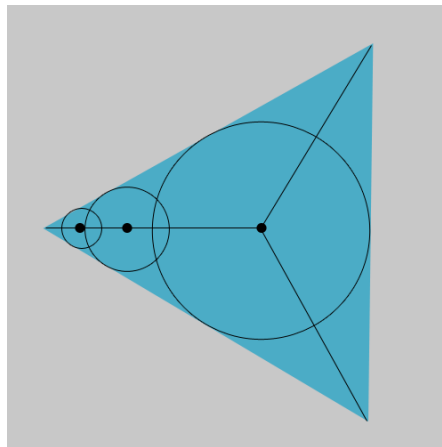


Figure 2.6: Computation of a medial axis by inscription of maximal circles.

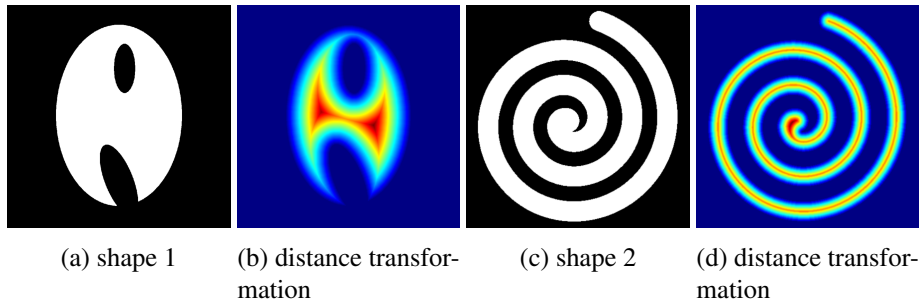


Figure 2.7: Distance transformation computed for two shapes. Red indicates high distance values, blue indicate low distance values.

2.2 Medial Axis Transformation

The medial axis transformation (MAT) was introduced by Blum in 1967 [5] and has since been widely used in shape analysis and representation.

A skeleton (the medial axis) is computed on a shape. For a shape S (for example a polygon) the medial axis $M(S)$ is defined as the set of points $\{p\}$ inside S , for which at least two points in the boundary of S are closest and equidistant to $\{p\}$ [23]. The skeleton derived based on the MAT therefore is a symmetric axis of the shape.

Using a geometric description: the medial axis is formed by the center of maximal circles that are inscribed into the shape at any position. Figure 2.6 shows an example. Therefore the MAT further provides a measurement of width, as for each point on the medial axis, a radius (the distance to the boundary of the shape) is associated with this point [23]. The shape can be reconstructed based on the medial axis and the width information stored with each skeleton point.

A distance transformation on a shape labels each foreground pixel with the distance to the next background pixel. The medial axis corresponds with the ridges formed by the local maximum values in the distance transformation. Figure 2.7 shows the values of a distance transformation computed for two shapes.

The medial axis skeleton can be used as an intermediate step to build a graph representation of a

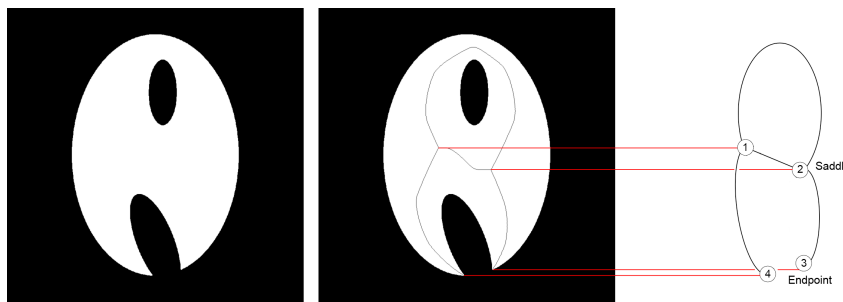


Figure 2.8: Graph representation of a shape derived based on the medial axis skeleton.

shape. In order to compute a graph based on the skeleton, branching points and endpoints in the skeleton need to be detected. These are used as the nodes in the graph. For the medial axis graph the nodes are connected by edges according to the skeleton. The medial axis skeleton computed on a shape, along with the corresponding graph is shown in Figure 2.8.

2.3 Morse Theory and Reeb Graphs

Reeb graphs are compact shape descriptors that preserve the topological characteristics of the described shape [4]. Reeb graphs are named after the French mathematician Georg Reeb and are based in Morse theory. The topological characteristics of interest are the number of connected components and the number of holes in the 2D structure. These two characteristics are the first two Betti numbers b_0 and b_1 . b_0 is defined as the number of connected components of a topological space and b_1 as the number of its 2-dimensional holes.

Based on critical points according to a scalar function a Reeb graph describes the topological structure that is the connectivity of level sets of e.g. 2D or 3D content [10]. In order to build a Reeb graph, critical points, of the structure to be represented, need to be computed.

A point (a, b) of a function $f(x, y)$ is called a critical point if both derivatives $f_x(a, b)$ and $f_y(a, b)$ are equal to 0 or if one of these partial derivatives does not exist [34].

Such a critical point either be degenerate or non-degenerate. These two cases can be distinguished via the Hessian matrix. The determinant of the Hessian matrix at a critical point x is then called the discriminant. If this determinant is zero then x is called a degenerate critical point of f (or non-Morse critical point of f). Otherwise it is non-degenerate (or Morse critical point of f).

A smooth, real-valued function $f : M_d \rightarrow \mathbb{R}$ is called a Morse function if it satisfies the following conditions for a d manifold M_d with or without boundary:

- all critical points of f are non-degenerate and lie inside M_d ,
- all critical points of f restricted to the boundary of M_d are non-degenerate,
- for all pairs of distinct critical points p and q , $f(p) \neq f(q)$ must hold [9].

Critical points of such a real-valued function are those points where the gradient becomes zero. The topological information of a shape described by a Reeb graph based on a function is related to the level sets of this function on the shape [4]. A change in topology appears with a change in the number of connected components in a level set. At regular points no topology changes occur. Topological changes occur at critical points only.

Vertices of the Reeb graph correspond to critical points of the function (points where the topology of M changes), edges describe topological persistence [4]. In other words: All nodes having the same function value are represented by one node in the graph, connections between nodes describe connections between segments of the underlying structure. A first example for such a Reeb graph is shown in Figure 2.9. The white foreground region is here analyzed top down. With a change in the number of connected components at a certain height a node is introduced

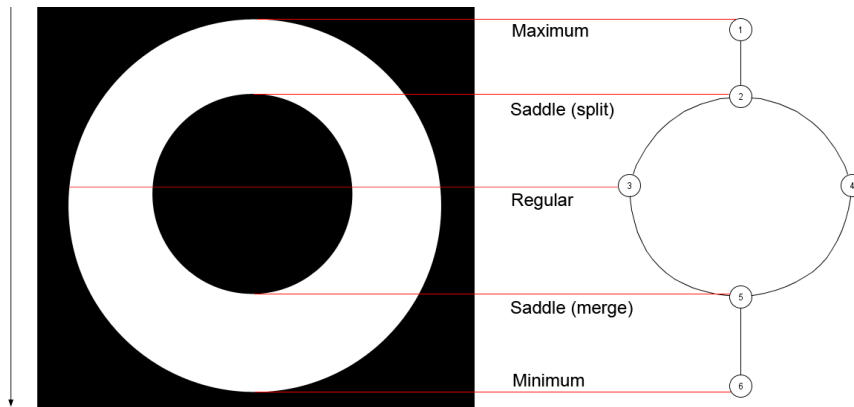


Figure 2.9: Critical points computed based on the height function (downwards) and corresponding Reeb graph. The white image region shows the foreground region described by the Reeb graph, black parts are background.

in the graph.

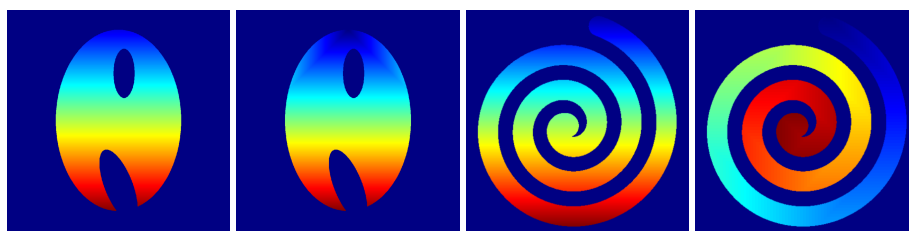
Reeb graphs are originally defined for the continuous space, but have been extended to the discrete domain: Here the Reeb graph is defined on a piecewise linear Morse function [10]. As the approach presented in this thesis provides an analysis of 2D image content, it is based in the discrete domain (image pixels). The Reeb graphs that are built on the root images are therefore discrete Reeb graphs and are based on the following definitions. In order to define a discrete Reeb graph, connective point sets and level-set curves are defined first:

- Two point sets are connected if there exists a pair of points (one point of each point sets) with a distance between these two points below a fixed threshold.
- If all non-empty subsets of a point set, as well as its complements, are connected, such a point set is called connective.
- A group of points that have the same Morse function value and that form a connective point set, is called a level-set curve [35].

The nodes in a discrete Reeb graph represent level-set curves, the edges connect two adjacent level-set curves, therefore the underlying point sets are connected [35].

The critical points are computed on the shape according to a Morse function. The most popular function in this scope is the height function. But distance functions as for example the geodesic distance are suitable as well.

- The height function in 2D is defined as the function f that associates for each point $p = (a, b)$ of a function $f(x, y)$ the value b as the height of this point p : $f(x, y) \mapsto y$.



(a) height function (b) geodesic distance (c) height function (d) geodesic distance

Figure 2.10: Example images for the two Morse functions: the height function is computed top-down, the seed point for the geodesic distance is in the center of the topmost pixel line of the foreground. The input images are shown in Figure 2.7a and 2.7c. Red indicates high function values, blue low function values.

- The geodesic distance is defined as the shortest distance measured between two points. Within the scope of this paper the geodesic distance is measured between any point of the root and the starting point of the root which is located at the center pixel of the topmost foreground pixel-line.

A comparison of the function values generated by these two Morse functions is shown in Figure 2.10. While for Figure 2.10a and 2.10b the function values are very similar, the Figures 2.10c and 2.10d show an example where the function values for height function and geodesic distance vary strongly.

In 2D critical points and corresponding nodes in the Reeb graph are minima, maxima or saddles [9]. The saddle nodes can be further distinguished: a saddle node that appears with a reduction in the number of connected components is further called merge (saddle) node, a split (saddle) node describes an increase in the number of connected components. When considering these two different types of saddle nodes that might appear in a Reeb graph, four different types of critical points and according nodes in the graph can be distinguished: maximum node, minimum node, split (saddle) node, merge (saddle) node. Besides these nodes corresponding to critical points, regular nodes can be added at any position and along any edge in the Reeb graph as they do not describe a change in topology. Nevertheless regular nodes can, for example, be used to describe changes in the color of the foreground region (see [3]).

Figure 2.9 shows an example for a Reeb graph based on a height function, containing all five types of nodes and the actual image the graph was computed on. Each edge in the Reeb graph describes a connected component. Therefore the edges of a Reeb graph are formed by connecting the node representing the birth of a connected component to the corresponding node representing the death of this component.

Another example Reeb graph containing all possible types of nodes is shown in Figure 2.11. The nodes in this graph correspond to the critical points of two Morse functions: the height function (going downwards) as well as the geodesic distance (the source pixel is located in the center of the top foreground pixel line) both result in this set of nodes.

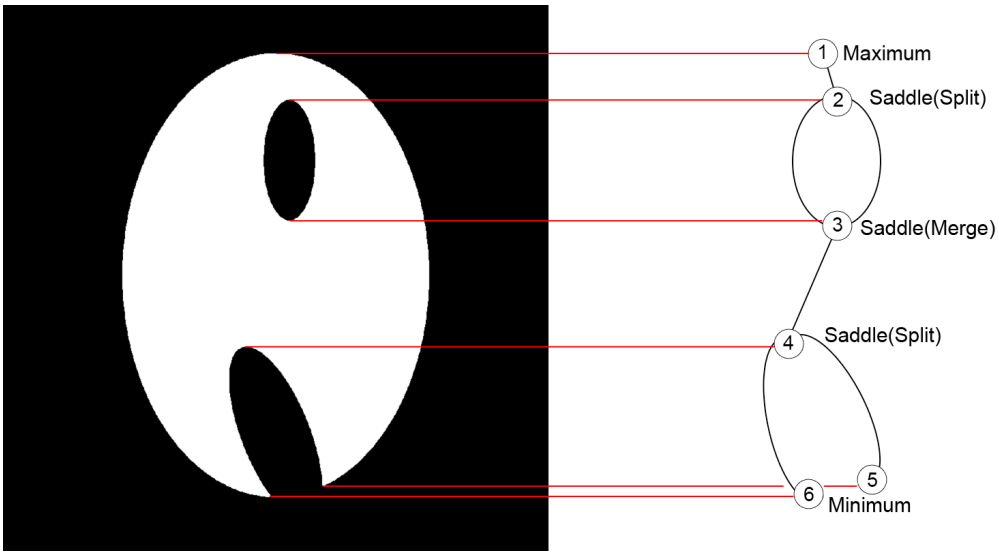


Figure 2.11: Reeb graph, computed for the white foreground region, according to height function (going downwards) and the geodesic distance (the source pixel is located in the center of the top foreground pixel line). Here both Morse functions generate identical critical point.

State of the Art

This thesis addresses two main areas of research: image based plant phenotyping and graph based shape representations.

3.1 Image Based Plant Phenotyping

In the research area of plant phenotyping the (semi-)automatic extraction of traits from image data allows for an analysis of large numbers of images / individuals. Table 3.1 shows an overview of available applications¹ that allow for a description of root systems or measurements on the roots. A very recent approach to describe root structures based on a skeleton representation of 2D image data and a 3D root reconstruction that allows for comparisons of root systems are discussed in the following sections.

Root System Analyser

The 2014 in [24] published tool “Root System Analyser” recovers root architectural parameters based on 2D images of the roots. The root system is represented as a graph in this approach. The approach proceeds based on the following operations:

1. Skeleton representation:
The root image is transformed into a medial axis skeleton in a first step.
2. Graph representation:
The branching points and end points are detected in order to form the nodes of the graph. As neighboring nodes are connected in the graph, an adjacency matrix is created, that stores this neighborhood information. The coordinates of edges are stored in an edge-list. Figure 3.1 shows four different root input images, the color indicates the age of the root,

¹<http://www.plant-image-analysis.org/>

Table 3.1: Available root system analysis and measurement applications.

Root system analysis tools		
Name	Analysis / measurements	Published in
DART	length, insertion, topology	[22]
EZ-Rhizo	length, insertion, topology, insert.-angle, #branches	[1]
GiA Roots	convex hull, length, perimeter, surface, volume, #branches, depth	[12]
GrowScreen-Root	length, insert.-angle, #branches	[28]
IJ-Rhizo	length, diameter	[30]
RootFly	length, diameter, color	[36]
RootNav	length, count, convex-hull, insertion, insert.-angle	[31]
RooTrak	3D reconstruction	[26]
RootReader 2D	#branches, length, topology, depth	[7]
RootReader 3D	length, width, depth, volume, surface, convex-hull, #branches, orientation, insert.-angle	[8]
RootScope	shape	[32]
Root System Analyser	length, insertion, insert.-angle, diameter, count	[24]
RootTrace	length, curvature, #branches	[11]
SmartRoot	diameter, insertion, insert.-angle, length, orientation, #branches, topology	[25]
WinRhizo / WinRhizo TRON	color, diameter, length, topology, volume, surface	[2]

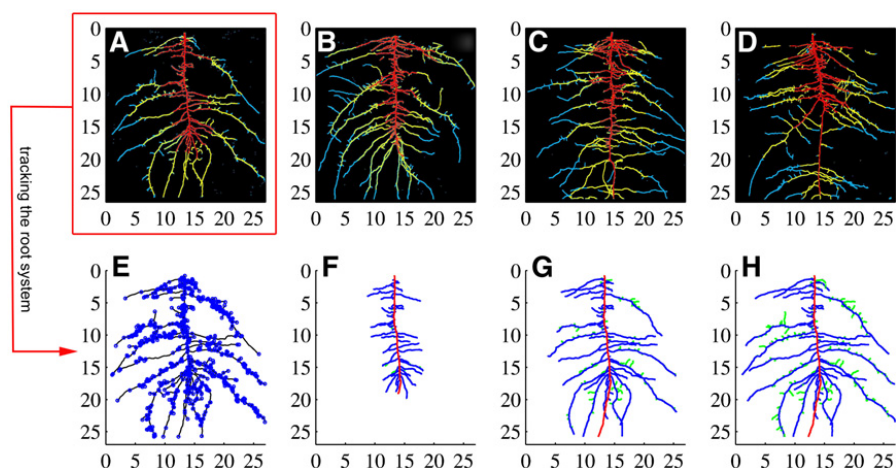


Figure 3.1: Images A-D show root input images, colors indicate the age of the root, with red labeling the oldest parts of the root. The corresponding graph representation of image A is shown in image E, individual branches of the root system are marked in the images F-H. Image taken from [24].

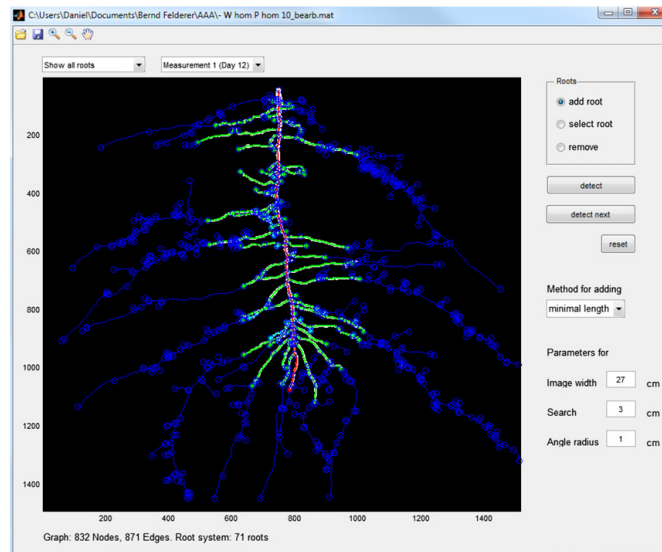


Figure 3.2: User interface of the “Root System Analyser”. Image taken from [24].

with red parts being the oldest parts of the root. For image A the nodes forming the graph are labeled in image E.

3. Root tracking:

To determine individual branches an underlying dynamic root architecture model is applied. The algorithm starts at predefined tips of roots and estimates the growth of the root for a small time step. All possible paths for this time step are found in the graph. The optimal growth path is determined dependent on the following characteristics: straightness and average diameter and it is penalized if an edge is already assigned to a root. Based on this method, overlaps and branching points can be distinguished. Figure 3.1 shows the marked individual branches of input images B-D in the images F-H.

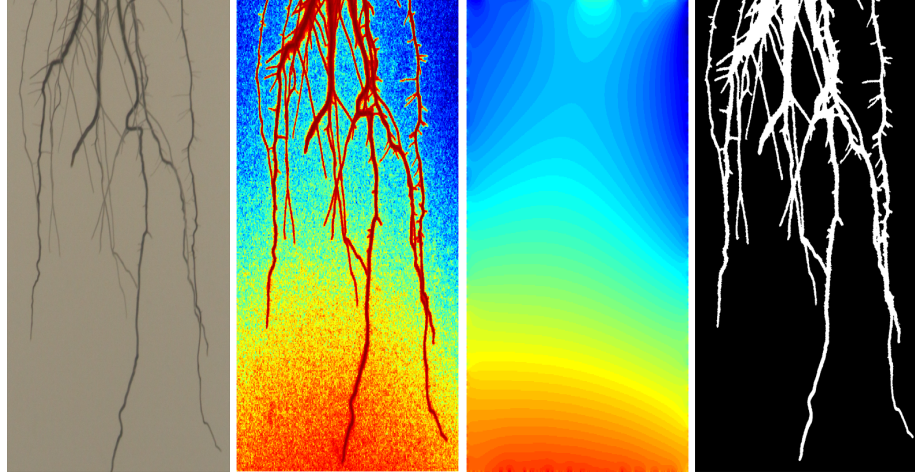
A graphical user interface allows the user to select individual branches, characteristics of these branches as for example length or width of the branch are then displayed. Moreover the user is able to correct the root system to a certain degree, by deleting roots or adding roots. Figure 3.2 shows an example for this graphical user interface.

Reconstruction of 3D Plant Root Shape

A related work on the analysis of branching patterns of roots based on a 3D reconstruction of the root architecture of rice plants is provided in [37]. The aim of this work is to build a 3D model of the branching pattern of roots based on 2D images. This 3D model allows for a comparison of the root systems of plants with different genotypes. The approach progresses in three steps:

1. Harmonic Background Substraction:

For the separation of the foreground (the root) from the background (the rest of the im-



(a) grayscale input root image (b) normalized intensities (c) harmonic background model (d) segmented root image

Figure 3.3: Segmentation process to separate foreground (root) from background. Image taken from [37].

age) the background is modeled according to a harmonic function. The foreground is constructed using the difference between an image and the background model. Figure 3.3 illustrates this segmentation process. The authors showed, that this approach more reliably preserves fine root structures than a single threshold or hysteresis thresholding.

2. Regularized Visual Hull:

3D shapes are often reconstructed from 2D images by the visual hull method. In this approach the visual hull is extended by a regularization term to a regularized visual hull. The visual hull can be defined as the set of voxels, that maximizes the total consistency. When $v \in \pi_k^{-1}(F_k)$ is the maximal set of voxels with projection F_k : “Define the consistency of a voxel v with the k -th image as:

$$cons_k(v) = \begin{cases} 1 & \text{if } v \in \pi_k^{-1}(F_k), \\ -N & \text{otherwise.} \end{cases} \quad (3.1)$$

and its total consistency as $cons(v) = \sum_{k=1}^N cons_k(v)$. Then the visual hull is the set of voxels that maximizes the total consistency:

$$V = arg \max_S \sum_{v \in S} cons(v).” [37] \quad (3.2)$$

Additionally one of the root images is used to improve the 3D reconstruction based on a regularization parameter $\lambda \geq 0$. This one image guides the 3D reconstruction dependent from λ . If λ is small, the influence of this image is low. The standard visual hull V is

computed and taken as a first regularized visual hull V_λ . Each pixel in the foreground of the “regularization image” F_i is visited next. If pixel u is not covered, a voxel with maximal consistency measure $cons(v)$ in the set of all voxels with projection F_i is found. $cons(v)$ is negative here as u is not covered yet. If $cons(v) + \lambda$ is positive, v is added to V_λ , otherwise v is discarded.

3. Repairing Connectivity:

As the regularized visual hull can consist of more than one connected component, the connectivity is repaired in a final step of the reconstruction. This is done by solving an optimization problem: as a solution a connected set of voxels U is required that minimizes the maximum distance to V_λ and the minimum inconsistency with the 2D images. This is done by building a graph that describes the connectivity of the voxels in the 3D reconstruction. The weights of the edges are set as the larger inconsistency value of the two nodes of the edge. Based on this, a minimum spanning tree is computed. The minimum cost path describes the path between two voxels that minimizes the weight of the edges in between. A voxel that lies on such a path is called a separating voxel. The solution to the optimization problem consists of all voxels in V_λ and all separating voxels of the minimum spanning tree.

3.2 Graph Based Shape Description

Skeletons and graph structures are widely used for shape representation and comparison.

The medial axis was introduced as a shape descriptor by Harry Blum in 1967 [5]. It has since been used in a multitude of various applications and developed into a standard approach for a simple shape representation. While the medial axis provides a skeleton representation that can serve as a basis for a graph representation, Reeb graphs derive a graph directly on the image data (as shown in [13]). For the representation of 3D data Reeb graphs can be efficiently derived from meshes (as shown in [35]) or from point clouds (presented in [29]). The following sections discuss a Reeb graph based approach to derive a skeleton representation based on a point set as input data and an approach for skeletonization of 3D scan data. Both approaches use the geodesic distance as a basic function to compute the Reeb graph. Besides the height function, the geodesic distance is as well used for the Reeb graph computation in the approach presented in this thesis.

Data skeletonization via Reeb graphs

The basic idea of the approach presented in [13] is to find a hidden space in discrete samples that has a graph-like geometric structure. A skeleton graph is therefore computed based on a point set as input data. An example of a skeletonization of a 2D image obtained through this approach is shown in Figure 3.4. The proposed method proceeds as follows:

1. Simplicial complex K :

The hidden domain, the input points are sampled from, needs to be approximated. This is done using a simplicial complex K . First a proximity graph is constructed. All input points

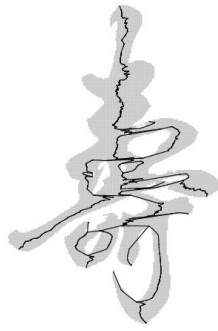


Figure 3.4: Reeb graph based skeletonization result of the input data in gray. Image taken from [13].

are therefore connected to their neighbors based on either the k -nearest neighborhood (the k neighbors that are closest to a point) or the r -neighborhood (the neighbors within distance r from a point) definition. All points and edges from this proximity graph are added to the simplicial complex. For any three points that are pairwise connected in the proximity graph, a triangle is added to the simplicial complex.

2. Reeb graph computation:

The Reeb graph is computed on the simplicial complex K using the geodesic distance according to a base point $b \in K$. This base point b is obtained by taking an arbitrary point $v \in K$ and defining b as the point furthest away from v . The geodesic distance function is here approximated by the shortest distance in the proximity graph. The (augmented) Reeb graph is computed according to the approach described in [15]: for a point q and the connected component C_q that holds q all triangles of the simplicial complex that intersect C_q are collapsed onto q and its adjacent edges. This procedure is repeated for all vertices in the simplicial complex K .

3. Post-processing:

In case the represented data was not already embedded in 2D or 3D, the input points are projected to \mathbb{R}^3 . The projected points are connected according to the Reeb graph

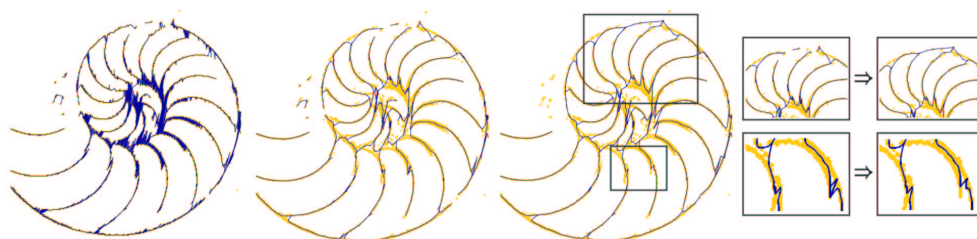


Figure 3.5: Reeb graph based skeletonization (left to right): input points in yellow - augmented Reeb graph - iterative smoothing - post-processing: adding missing links (top) and removing spurious branches and loops (bottom). Image taken from [13].

connectivity and are iteratively smoothed by substituting the position of a point by the average of its neighbors. Missing links are added by connecting pairs of degree-1 nodes with a distance smaller than a certain threshold. To remove spurious branches or loops in the Reeb graph features and their persistence are computed on the graph. The graph is simplified by merging features with a persistence smaller than a predefined threshold.

Figure 3.5 shows an example images for the methodological sequence described above.

Human body scan segmentation based on Reeb graphs

The overall goal of the approach presented in [35] is to segment a 3D human body scan into subsets which correspond to functional body parts. As this segmentation is done pose-independent, a graph representation is computed on the 3D data first. The approach is divided into the following operations:

1. Computation of Morse functions:

For the computation of the Reeb graph two Morse functions are used: the geodesic distance between a point and a source point and the sum of geodesic distances for one point as the sum of geodesic distances to all other points of the surface. The geodesic distance to a source point is computed on the voxels using a wavefront approach: starting from a source point in each iteration all neighboring voxels are taken as a level set with the same distance to the source. In this way the level-sets are already extracted while computing the Morse function. The sum of geodesic distances requires the computation and summation of the geodesic distance for all voxels with the current voxel as source point.

2. Decomposition into level-set curves:

Based on the extracted level-sets the Reeb graph is constructed by representing each level-set curve, a iso-valued connective point set, by a node in the graph. The edges in the Reeb graph are formed according to the connection of level-set curves with each other and connect nodes of two adjacent level-set curves.

3. Extraction of branches:

In order to extract branches corresponding to body parts, critical nodes in the Reeb graph need to be detected. The main problem here is that noise in the data leads to false positives, critical nodes that only describe noise. Critical nodes are divided into three types of nodes: O-type (loop), λ -type (split) and Y-type (merge). While the O-type consist of two saddle nodes connected by two edges, the types λ and Y are both consistent of one saddle node connected to two leaf nodes. While these two types are topologically equivalent, they can be distinguished when taking the direction of the Morse function into account. The human body topology can not produce O-type saddle nodes. Based on the direction of the Morse function one of the two saddle nodes λ -type and Y-type can not occur for a human body as well. These nodes are therefore caused by data corruption and can be discarded. Furthermore, it is assumed that a branch associated to a true either λ -type or Y-type saddle node may not be smaller than the size of the smallest body part. In this way false critical nodes are discarded and the true critical nodes are detected. This nodes represent the

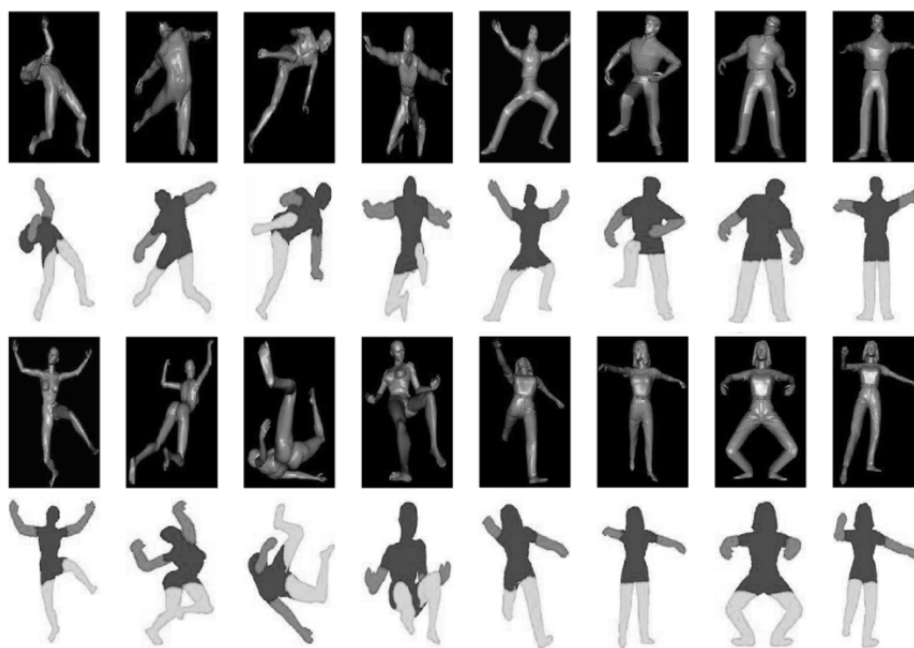


Figure 3.6: Segmentation results of simulated scans. Image taken from [35].

bottom of the feet, the groin, hand tips, armpits and top of the head. Body parts are identified based on the branches between the nodes and the 3D data is segmented into legs (branches between feet and groin) and arms (branches between hands and armpits). The rest of the data corresponds to the torso and head segment.

Figure 3.6 shows the final segmentation for synthetic 3D data.

Dataset and Pre-Processing

The images of the root dataset were acquired by the Gregor Mendel Institute of Molecular Plant Biology (GMI). For this dataset images of the plant *Arabidopsis thaliana* were taken. This plant is a model organism, which is widely used in plant sciences, due to the small size of its genome, the small size of the plant itself and its rapid life-cycle [16].

The whole set of plant images used here consists of 9 sets of time series. Each set holds 6 images of one plant taken over time (day 1, day 4, day 8, day 12, day 16 and day 20 of the growth period). Of these 54 images, 34 images (day 4 -day 20) are analyzed, the other images are too early in the growth process and therefore too small in structure to be represented by a non-trivial Reeb graph.

4.1 Image Acquisition

At the GMI the seeds are planted on a nutrient containing agar gel surface in plastic petri dishes / plates. One of these petri dishes holds 24 plants (2 rows of 12 plants each). Figure 4.1 shows one of these petri dishes, the plants in this image are already 20 days old. These plates are stored vertically in growth chambers that allow for controlled conditions as constant temperature, illumination or humidity. The vertical orientation of the plates is based on the fact that roots grow primarily in the direction of gravity. The amount of water, or for example certain nutrients, further affect the speed of the growth process of the roots.

The images are taken using an image scanner. A special fixture allows for two datasets / plates to be placed in an exact known position inside the scanner. A ruler at this fixture is imaged together with the plate and serves as a reference marker. This allows for later measurements on the image to be converted from pixel measurements to real world dimensions as for example millimeter. The images are acquired with a scan at 1200 dpi resolution with 8bit color depth, therefore one image is of approximately 6000x6000 pixels in size. The images are stored as bmp

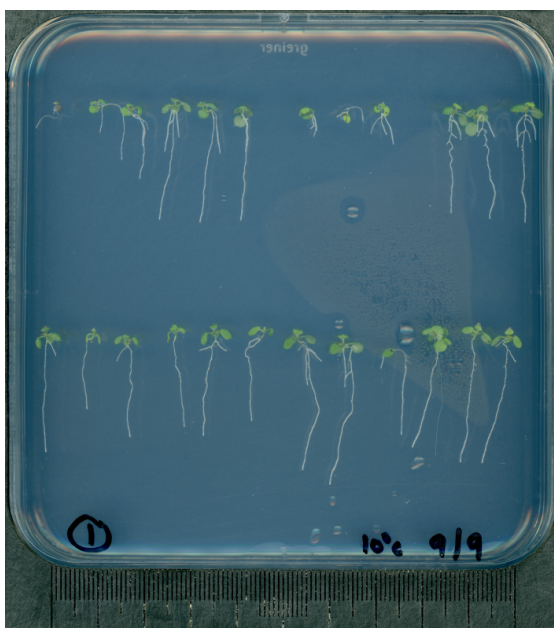
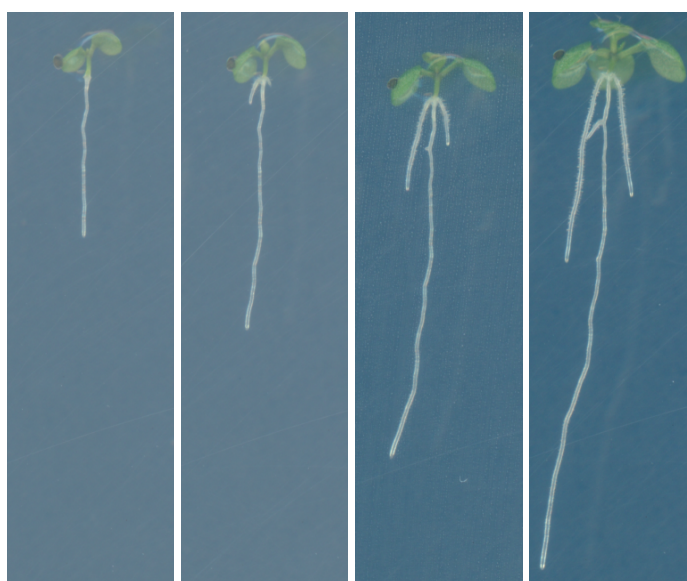


Figure 4.1: Plants grown inside a petri dish, picture on day 20 of their growth cycle. The ruler at the bottom of the image is used as a reference marker for measurements on the image.



(a) day 8 (b) day 12 (c) day 16 (d) day 20

Figure 4.2: Example images of the root dataset: root004 on several days during the growth cycle.

files of about 150MB.

Along time several successive images are acquired this way, as each plate is scanned at several successive days of the growth process. A 3D stack of 2D images over time is thus created for each root. An example for such successive images of one plant is shown in Figure 4.2

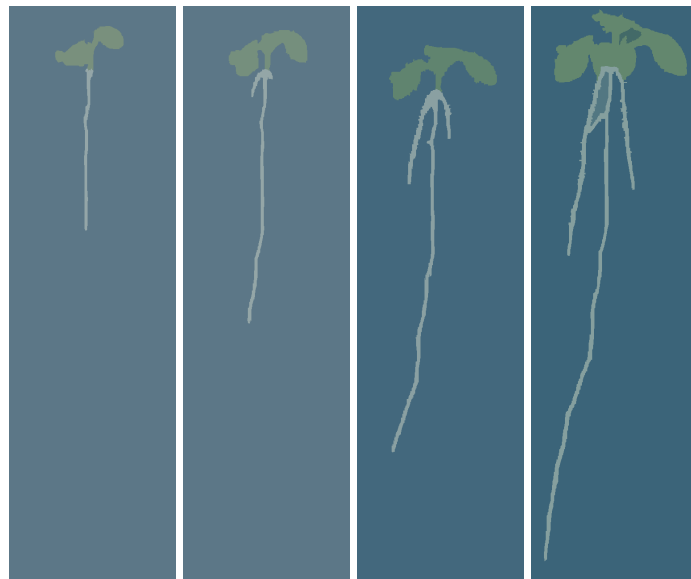
4.2 Pre-Processing

As a first step in the process of a graph based root representation pre-processing methods need to be applied.

The plants are pictured in a set of 24 plants in one petri dish. Therefore each of these images needs to be cut to single plant images. The size of these single plant images is determined based on the image of the last day of the growth cycle. All younger plants are extracted accordingly. For the images in Figure 4.2 this pre-processing step has already been carried out.

As the approach presented in this thesis requires a pre-segmented image as in an input, the single plant images are segmented in the next pre-processing step. For this segmentation procedure a pyramid based approach is used. Image pyramids describe an image at multiple levels of resolution. The bottom level of the pyramid corresponds to the high resolution of the original image. For higher levels the resolution decreases based on a reduction function applied to the pixels within the reduction window [20]. The segmentation approach is presented in [14]. It takes an arbitrary image as input and returns a hierarchy of segmented images. In a next step the user can modify the segmentation: The user has the possibility to merge regions or prevent them from being merged in higher levels of the hierarchical segmentation. For changes made by the user the pyramid is recomputed to adapt the underlying combinatorial maps [14]. All images of the root dataset used in this thesis were segmented using this interactive, hierarchical segmentation tool.

The segmented images of the root dataset consist of 2 foreground regions (leaves and roots, only the roots are analyzed for this approach) and up to 2 holes in the foreground structure. For reasons of this needed preceded segmentation, the dataset is restricted in its size, as this semi-automatic segmentation approach is costly in terms of time. For the quite thin root structures a high amount of user interaction is needed to generate a suitable segmentation result. Figure 4.3 shows the segmentation results for the root images shown in Figure 4.2.



(a) day 8 (b) day 12 (c) day 16 (d) day 20

Figure 4.3: Segmented images of root004, day 4 to day 20.

Graphs on Root Images

Within the scope of this thesis, root images are represented by three graph based representations: graphs based on a medial axis skeleton, Reeb graphs based on the height function as Morse function and Reeb graphs based on the geodesic distance as Morse function.

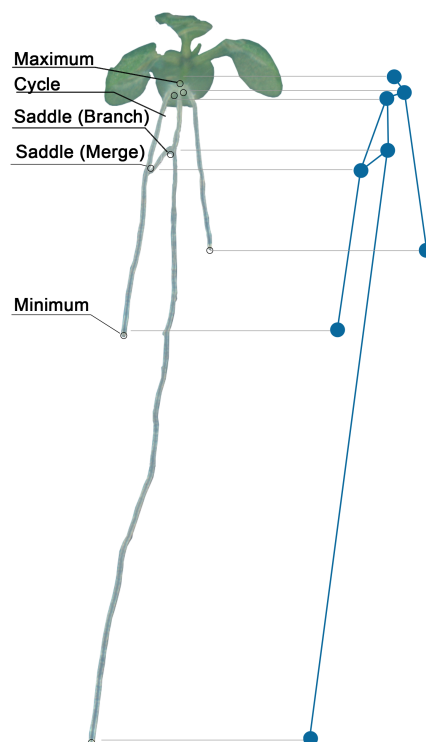


Figure 5.1: Example root with characteristic locations labeled and a possible graph representation.

This graph representations capture characteristics of the roots such as branching points and end points of branches. Figure 5.1 shows an example image of a root, with the characteristic points labeled.

An overview of the implementation is given as pipeline for the three approaches presented in this thesis in Figure 5.2. The approaches are discussed in detail in the following sections.

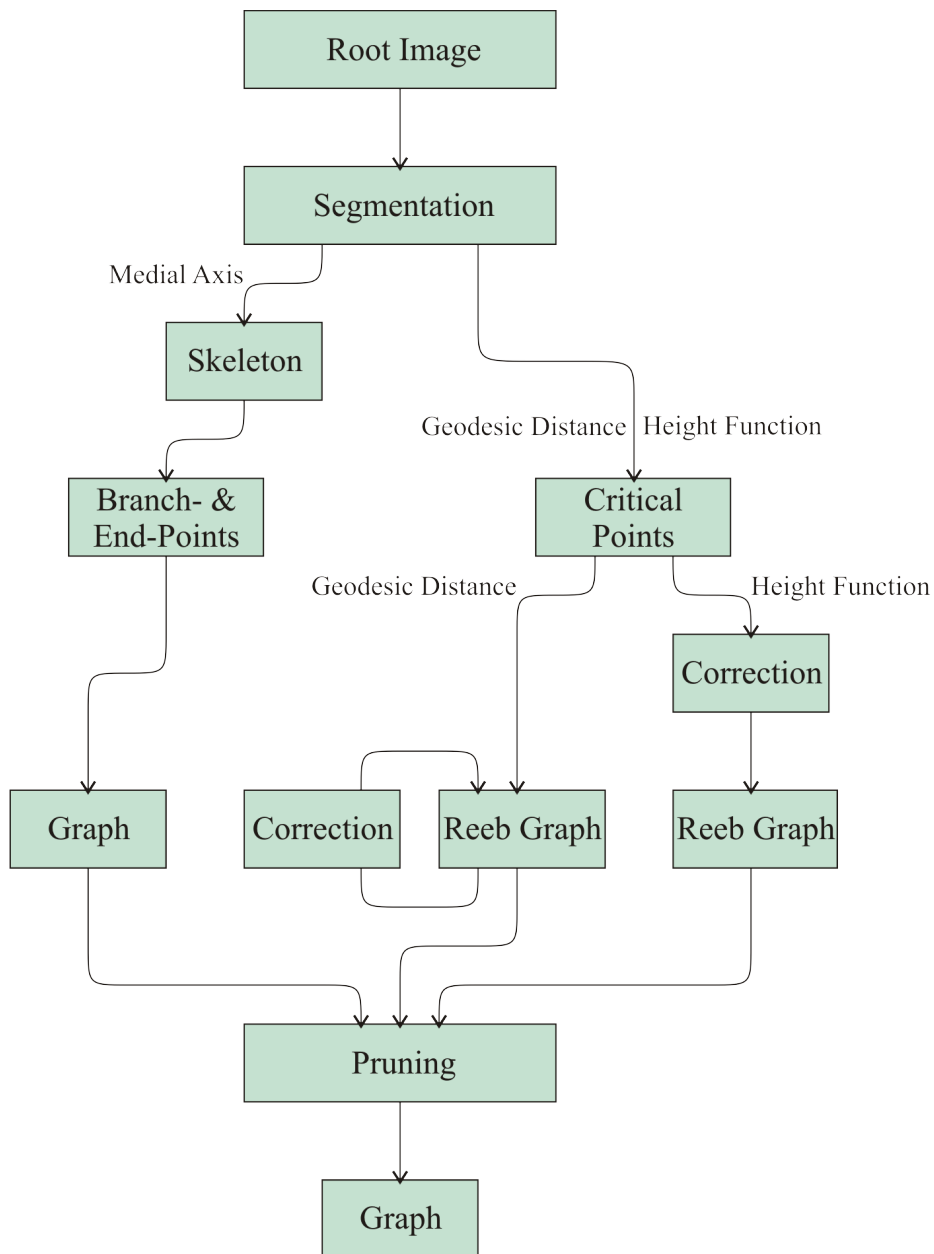


Figure 5.2: Implementation pipeline for the three approaches.

5.1 Computation of Medial Axis Graph

For the computation of the medial axis an iterative thinning approach is applied to the foreground region of the segmented image. For each pixel three conditions are checked in two sub-iterations, if all three of them hold, the pixel is removed. If one or more of the three conditions do not hold, the pixel is identified as a skeleton pixel and kept. Moreover for each skeleton pixel the shortest distance to the background is stored as a width.

The conditions are checked on the neighborhood of a pixel p , the east-neighbor is labeled x_1 , the others are labeled with successive numbers counter-clockwise as illustrated by the following Figure 5.3:

x_4	x_3	x_2
x_5	p	x_1
x_6	x_7	x_8

Figure 5.3: Pixel neighborhood.

According to [21] the conditions are:

1. sub-iteration: the pixel is deleted if the conditions G_1 , G_2 and G_3 hold:

- G_1 : $\sum_{i=1}^4 b_i = 1$, with:

$$b_i = \begin{cases} 1 & \text{if } x_{2i-1} = 0 \text{ and } (x_{2i} = 1 \text{ or } x_{2i+1} = 1), \\ 0 & \text{otherwise.} \end{cases}$$
- G_2 : $2 \leq \min(n_1(p), n_2(p)) \leq 3$, with:

$$n_1(p) = \sum_{k=1}^4 x_{2k-1} \vee x_{2k}$$

$$n_2(p) = \sum_{k=1}^4 x_{2k} \vee x_{2k+1}$$
- G_3 : $(x_2 \vee x_3 \vee \neg x_8) \wedge x_1 = 0$

2. sub-iteration: the pixel is deleted if the conditions G_1 , G_2 and G'_3 hold:

- G_1 : same as for sub-iteration 1
- G_2 : same as for sub-iteration 1
- G'_3 : $(x_6 \vee x_7 \vee \neg x_4) \wedge x_5 = 0$

For each computed skeleton branching points and endpoints of branches are detected according to their neighborhood-structure. These are used as nodes of the graph. An endpoint is detected as a pixel with only one neighboring pixel (see Figure 5.4b), a branching point has more than two neighboring pixels (see Figure 5.4a).

The actual graph is then computed on the skeleton using line tracing and stored as an adjacency matrix. For each branching point all skeleton connections are followed until another branching point or an endpoint is reached. If a branching point or endpoint is reached, the according entry in the adjacency matrix is set to one.

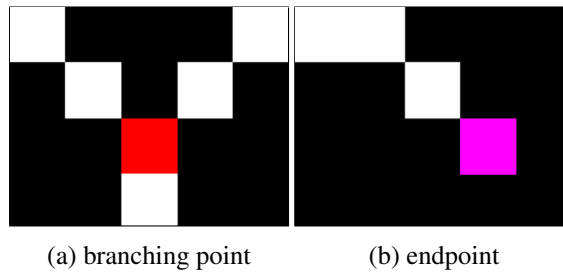


Figure 5.4: Branching point in a medial axis skeleton marked red, endpoint marked pink.

5.2 Computation of Reeb Graph (Height Function)

For the computation of the critical points according to the height function the basic idea is, to analyze the border of flat-regions in the image. The foreground region borders are therefore analyzed with regard to horizontal borders as these might describe a change in the number of components. Critical points indicate topology changes and these changes might only occur at the border of a region but not within a region.

For each horizontal border its endpoints are checked for connected vertical borders:

- no change in topology - no critical point: one endpoint connected to a vertical border going up, the other connected to a vertical border going down
- split or birth: both endpoints connected to a vertical border going down
- merge or death: both endpoints connected to a vertical border going up

The so found critical points are located at the center of such a horizontal border. Figure 5.5 shows examples for computed critical points.

Correction of critical points

Due to the resolution of the image, the discretization of the root and further distortions during the segmentation process, two critical points may be computed at the same height. Critical

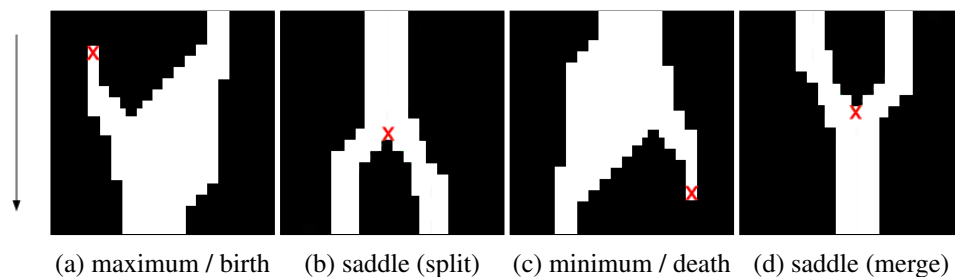


Figure 5.5: Four different types of critical points, computed according to the height function [18].

points with the same Morse function value contradict the third criteria of Morse theory (defined in Section 2.3). Thus, a Reeb graph can not be built in such a case. Figure 5.6a illustrates the problem: solid lines indicate well-defined connections in the graph, the dashed lines indicate possible connections. From these possible connections, one connection from the black center node to a red node is needed and the green node needs to be connected to one of the red nodes as well. In order to determine a correct set of edges between the nodes, the following technique is used to solve the ambiguous connections for critical points on the same height:

For two critical points on the same height, a rotation is applied by adding a factor f , $0 \leq f < 1$. A critical point $p = (x, y)$ is rotated to $p' = (x, y + f)$ with f computed as $f = \frac{1}{w}(x - 1)$. The order of heights is preserved by this controlled rotation as only critical points that were primarily at the same height are changed. The critical points are mathematically discriminative after applying this correction, while staying in the actual pixel line, when rounding down the y -coordinate of such a critical point to an integer value [18].

The configuration of the nodes in Figure 5.6a is rotated to the configuration in 5.6b. As the critical points are now on different heights, a unique Reeb graph can be built. The Reeb graph for Figure 5.6a after application of this correction is drawn in Figure 5.6b.

Construction of graph

The critical points function as nodes in the Reeb graph. As a final step the edges of the graph need to be computed. The edges are stored as adjacency matrix.

In order to obtain the connections between vertices, all critical points are sorted according to their height. Starting with the topmost critical point an approach similar to the flood fill algorithm is deployed for all nodes:

1. The pixel line of the current critical point is followed to the right until a critical point in the same pixel line or a background pixel is reached.

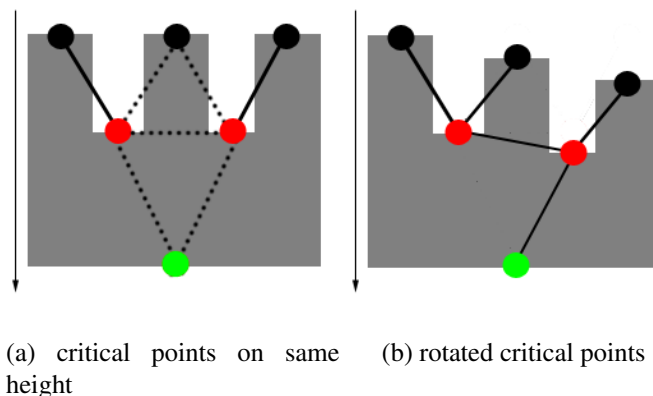
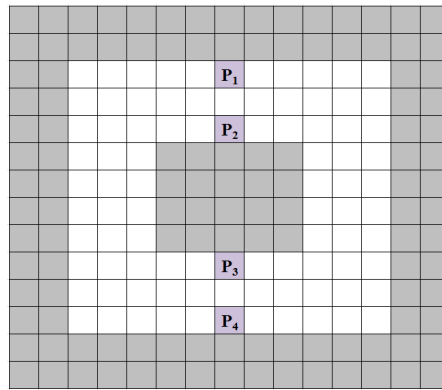
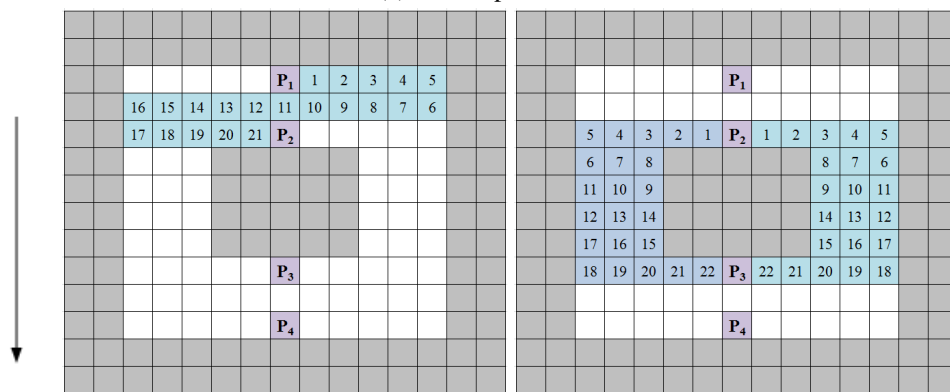


Figure 5.6: For critical points on the same height a unique Reeb graph can not be built, as the connections of the nodes in the graph are ambiguous [18].



(a) critical points



(b) edge P_1P_2

(c) cycle P_2P_3

Figure 5.7: Computation of edges in Reeb graph

2. In case a background pixel was reached, the searched is moved one row down and this pixel line is followed to the left, until a critical point or a background pixel is reached.
3. In case a background pixel was reached, the search again moves one row down and progresses to the right until a critical point or background pixel is reached.
4. Step 2 and 3 are repeated until a critical point is found.

Once a critical point is found the adjacency matrix is set to one for the according edge. In case the searched started from a critical point of type saddle (split) the search for critical points is done twice, once starting to the right of the critical point and once to the left. Figure 5.7 shows how the connections are computed for the critical points P_1 (Figure 5.7b) and P_2 (Figure 5.7c) in the white foreground region of Figure 5.7a.

5.3 Computation of Reeb Graph (Geodesic Distance)

For each foreground pixel the geodesic distance to one predefined source pixel is computed. For the root images the distance is computed from the center pixel of the topmost pixel row. As a distance measurement for computing the critical points the Chebyshev (chessboard) distance is used.

There is only one node of type maximum in a Reeb graph based on the geodesic distance, which is the seed point (distance = 0). Minimum nodes are found as the position of a maximum distance in a branch (local maxima). Saddle points are determined as locations at which foreground parts with the same distance to the source are split in two connected components or are merged from two into one connected component. Figure 5.8 shows an example for a critical point of type split. The source pixel is located in the top right corner (distance 0), at distance 8 the structure is split into two connected components, this would result in a critical point of type split at distance 7. The split in Figure 5.8 would not be determined as a critical point when analyzing the structure according to the height function, as the number of connected components stays one for all heights in this structure. The geodesic distance is more flexible in the identification of critical points, a alignment of the analyzed structure to the perpendicular axis is not needed.

Correction of critical points

As for the Reeb graphs based on the height function, critical points with the same function value (in this case distance to a source pixel) may occur. However, for this Reeb graph there is no direct correction applied to the critical points. The critical points are rather connected as described in the next section, to the next critical point that is encountered while tracing back the foreground structure from one critical point towards the source pixel. However, if two critical points at exactly the same distance are encountered - this might for example happen when computing

14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
14	13	12	11	10	9	8	7	6	5	4	3	2	1	1
14	13	12	11	10	9	8	7	6	5	4	3	2	2	2
14	13	12	11	10	9	8	7	6	5	4	3	3	3	3
				10	9	8	7	6	5	4	4	4	4	4
				10	9	8	7	6	5	5	5	5	5	5
							7	6	6	6	6	6	6	6
							7	7	7	7	7	7	7	7
									8	8	8	8	8	8
									9	9	9	9	9	9
									10	10	10	10	10	10
										11	11	11	11	
											12	12		
												13	13	

Figure 5.8: Geodesic distances are computed using the chessboard distance, the source pixel is located in the top right corner. At distance 8 the foreground is split into two connected components, this would result in a critical point of type split at distance 7.

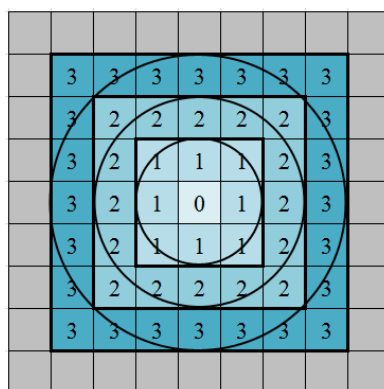


Figure 5.9: Geodesic distance computed as chessboard distance from the center pixel leads to concentric squares. The euclidean distance is described by the concentric circles.

to closest point to the maximum node -a second distance measurement is used for the decision. Starting from the source pixel the geodesic distance is computed based on the euclidean distance. The risk of encountering two critical points in the same branch having the same euclidean geodesic distance to a source pixel is negligibly small.

For two critical points at the same Chebyshev distance from the source pixel, we can use the euclidean distance. Both critical points are in any case at same chessboard distance, the decision is based on the differing euclidean distances. The values obtained by the euclidean distance are always larger than the chessboard distance as the euclidean distance is described by the incircle of the square formed by the chessboard distance (see Figure 5.9).

However, these decisions based on the euclidean distance may result in wrong decisions, one saddle node may have four adjacent nodes while another one only has two adjacent nodes. As illustrated in the “geodesic distance” branch in Figure 5.2 a correction procedure is applied to repair the graph in case of incorrect decisions.

Construction of graph

The connections in the Reeb graph are obtained in two steps:

1. First all critical points of type saddle or merge are connected. Therefore, for each saddle node the foreground structure is traced backwards. This means starting at a saddle node P with distance x to the source node, the algorithm moves from the region with distance $x - 1$ that is reachable from P to $x - 2$ etc. until a critical point is reached. Figure 5.10 demonstrates this procedure for the critical point P_2 at distance 19. For saddle nodes of type merge this is done for both branches and two connections are found.
2. As a second step, the maximum node (source node) in the geodesic distance Reeb graph is connected to the node closest to it. All minimum nodes are connected to the closest split or merge node. Again the foreground structure is traced backwards (in the direction of decreasing distance) until a critical point is found.

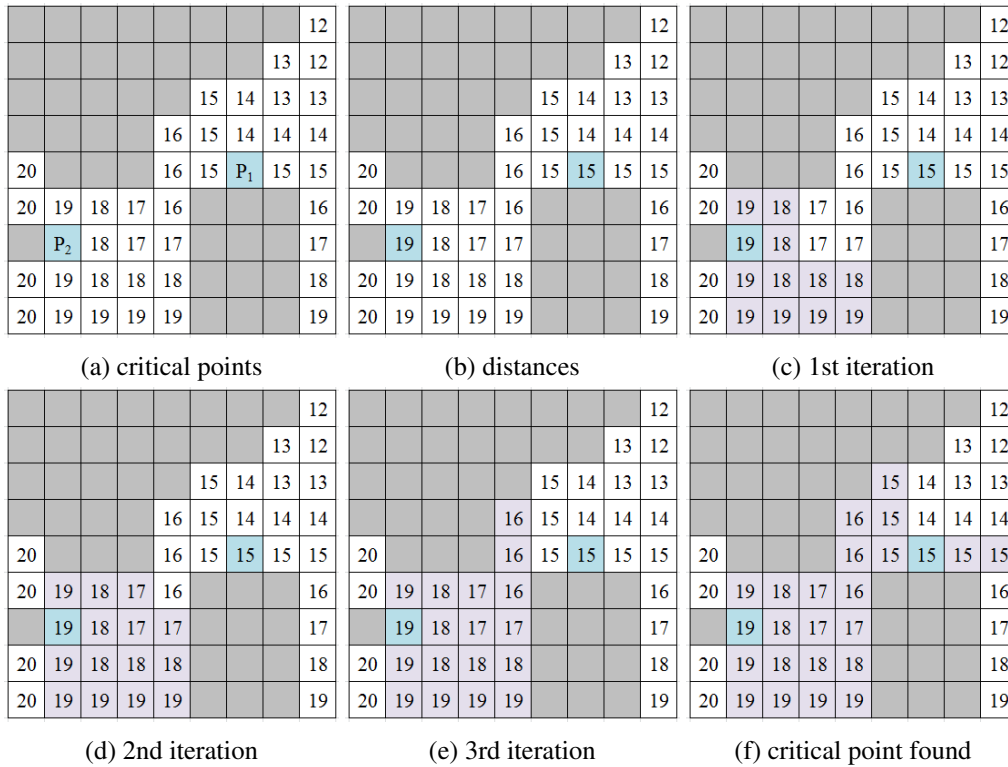


Figure 5.10: Example for the computation of edges in the geodesic distance Reeb graph. The connection between critical point P_2 and P_1 is built.

3. For two critical points at the same distance, wrong decisions may be taken during the second step of this process. Therefore the correctness of the graph is checked after both steps are finished. In case a saddle node is adjacent to four other nodes, there needs to be a second saddle node with only two adjacent nodes. For a saddle node with four adjacent nodes, the minimum or maximum node, that is at the largest distance from the saddle node is deleted from the adjacency of the saddle node and connected to the closest saddle node that has only two adjacent nodes.

All edges that are computed in that way are stored in an adjacency matrix.

5.4 Improvements on the Graphs

During preprocessing the image is segmented. This segmentation procedure may generate frayed borders in the foreground region. These artefacts introduce additional, spurious nodes in the graphs. In order to use the graphs as a representation of the root structure, these additional branches need to be removed. A simple graph pruning approach is applied for this purpose. Branches and endpoints are discarded according to certain criteria. If needed, nodes are relinked

after pruning. Saddle nodes that are reduced to a degree-2 node by the pruning process are smoothed out and therefore removed from the graph.

Pruning based on length

This approach dismisses branches with an euclidean distance between the endpoints of less than 1.5% of the image height. The images are cut in a preprocessing step to single plant images. The size of the image is for each plant based on the latest day of its growth cycle. All earlier images of this plant are cut to the same size. The image height therefore is directly linked to the length of the longest branch as this determines the image size. The threshold of 1.5% of the image height was determined empirically.

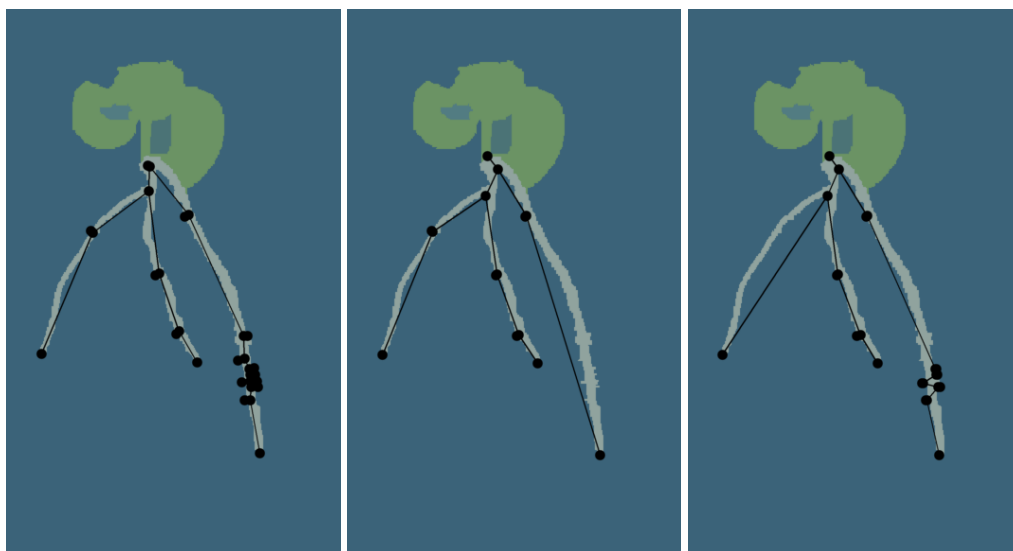
Pruning based on width

For the medial axis graph a second pruning approach based on the width of branches was tested. The basic idea here is that branches due to segmentation artefacts are in most cases only 1 or 2 pixels wide. True branches that resemble the spurious branches in length are mostly broader than that. However, the pruning process can not be based on the width of branches only, as especially longer branches tend to be thin in the end part and may be discarded during pruning. Therefore pruning candidates are detected using the length based pruning approach described before. These pruning candidates are verified for pruning by checking the average width of the branch between the endpoints of the edge forming the branch. This can be easily done, as the width of the foreground structure is stored with the medial axis skeleton. In case the average width is below 7 pixels the pruning candidate is verified and the according branch discarded. The width threshold again was determined empirically.

Results

Figure 6.1 shows results for root 09, day 20. Here the Reeb graph based on the height function produces the best result, as the root is well aligned with the perpendicular axis and the response to the artefacts introduced by the segmentation approach is lower for this Reeb graph. This is based on the restriction of the critical points, as these only occur with horizontal borders/gaps in the foreground structure. The medial axis responds very sensitive to noise in the foreground borders, the high number of spurious branches shows this.

By application of graph pruning most the spurious branches can be detected and discarded.



(a) medial axis graph

(b) Reeb graph (height func-

(c) Reeb graph (geodesic dis-

Figure 6.1: Resulting graphs for root 09, day 20.

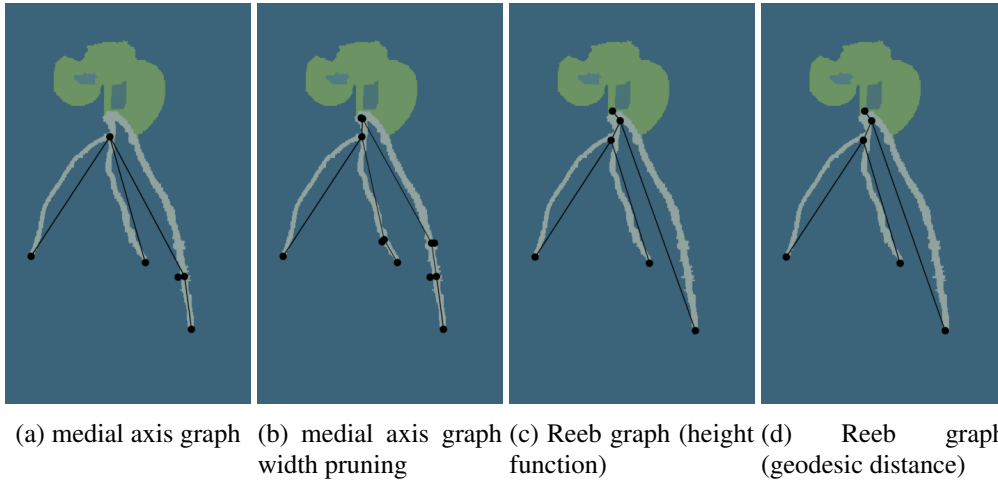


Figure 6.2: Resulting graphs for root 09, day 20 - length based graph pruning was applied to the results shown in Figure 6.1. For the medial axis based representation the width pruning approach was applied as well.

Figure 6.2 shows again the resulting images for root 09, day 20 with length based graph pruning applied. The resulting graphs based on the Reeb graph approaches after pruning capture the characteristics that were intended to be obtained, well: start and endpoints of the roots are found, as well as the two branching points. For the medial axis approach under pruning according to length, the actual startpoint (for the medial axis it is just one of the endpoints and therefore not secure under pruning) was discarded in the pruning approach. This created a regular node (degree 2) for the higher branching points which thus was discarded as well. While the startpoint and one branching point are missing, one spurious branch was kept. By application of width based pruning described in 5.4, the missing startpoint and branching point are kept, but two more spurious branches (in total three spurious branches) are kept just as well (see Figure 6.2b.)

Table 6.1: Branches wrongly discarded (false negative) and wrongly accepted (false positive) in the graph pruning approach.

Wrong decisions on graph pruning			
representation	# images	# false negatives	# false positives
RG height function	8	10	0
RG geodesic distance	7	7	2
medial axis	17	11	16
medial axis width pruning	18	4	29

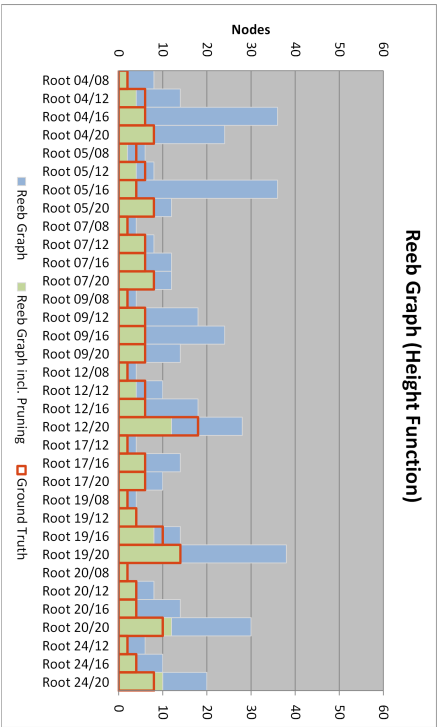
6.1 Pruning Effect

For the comparison of the three different graph representations used in this thesis, the numbers of nodes computed for the graphs are compared. As illustrated by the Figures 6.1 and 6.2 spurious branches may be introduced due to segmentation artefacts. Due to these segmentation artefacts the number of nodes is in general high for all three representations, but can be reduced by up to 70% under the application of graph pruning (see Table 6.2). The applied graph pruning is able to remove most of the branches. However, spurious branches may still be accepted under application of graph pruning (false positives), while true branches that resemble the spurious branches (in length or width) are therefore discarded during the pruning process (false negatives). Table 6.1 shows the number of false positives or negatives that occur for each of the discussed approaches. The medial axis approach introduces the most false positives, especially the width pruning approach is very likely to accept spurious branches. However, the number of false negatives is lowest under this pruning approach. The best results are obtained using the Reeb graphs based on the geodesic distance in combination with the length based graph pruning. While almost all false positives are correctly discarded during graph pruning, only few true branches (false negatives) are missing. The graphs in Figure 6.3 show that the Reeb graph according to the geodesic distance approximates the ground truth best.

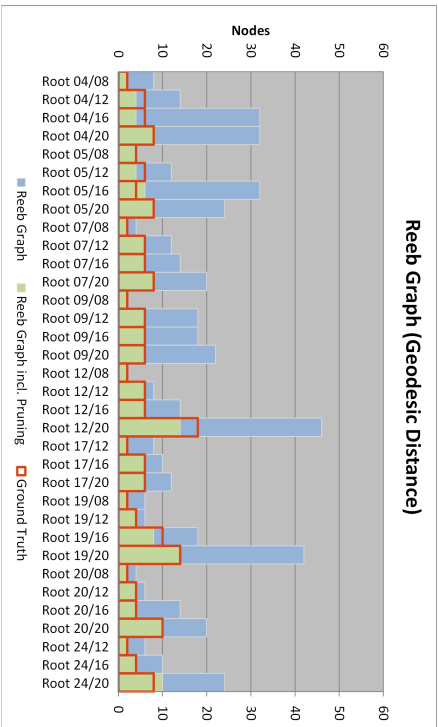
Table 6.2 shows the numbers of nodes for all root images in the dataset and the reduction of the node numbers when using graph pruning. Figure 6.3 shows the according graphs to the results presented in Table 6.2.

The abbreviations used in Table 6.2:

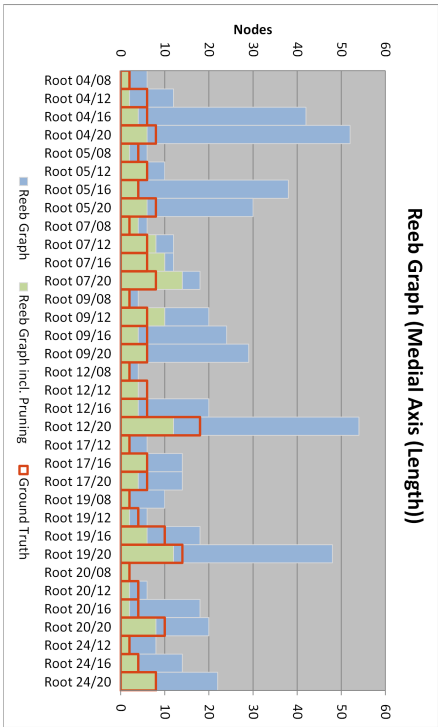
- RG1: Reeb graph based on height function
- RG1 + P: Reeb graph based on height function including pruning
- RG2: Reeb graph based on geodesic distance
- RG2 + P: Reeb graph based on geodesic distance including pruning
- MAG: Medial axis graph
- MAG + P: Medial axis graph including pruning
- MAG + WP: Medial axis graph including width based pruning
- RSA: Result from the Root System Analyser tool [24]
- GT: Human generated ground truth



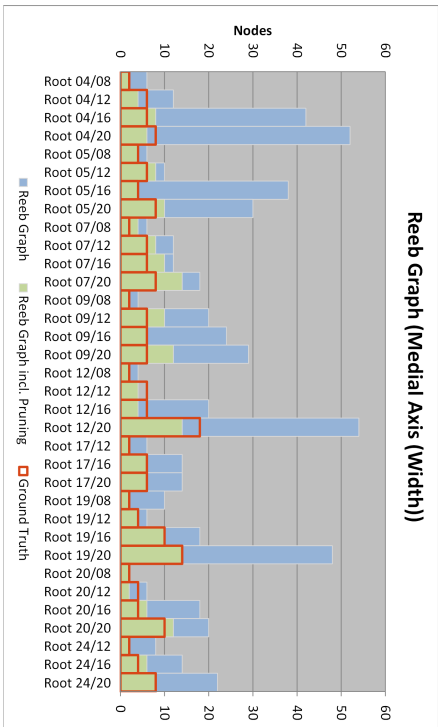
(a) Reeb graphs (height function)



(b) Reeb graph (geodesic distance)



(c) Medial axis graphs



(d) Medial axis graphs (width pruning)

Figure 6.3: Numbers of nodes used in the graphs compared to human ground truth.

Table 6.2: Overview: numbers of nodes for the different graphs representations.

Numbers of nodes in the graph									
Image	RG1	RG1 + P	RG2	RG2 + P	MAG	MAG + P	MAG + WP	RSA	GT
root04, day 8	8	2	8	2	6	2	2	5	2
root04, day 12	14	4	14	4	12	2	4	6	6
root04, day 16	36	6	32	6	42	4	8	25	6
root04, day 20	24	8	32	8	52	6	6	32	8
root05, day 8	6	2	4	4	6	2	4	2	4
root05, day 12	8	4	12	4	10	6	8	7	6
root05, day 16	36	4	32	6	38	4	4	15	4
root05, day 20	12	8	24	8	30	6	10	21	8
root07, day 8	4	2	4	2	6	4	4	6	2
root07, day 12	8	6	12	6	12	8	8	11	6
root07, day 16	12	6	14	6	12	10	10	9	6
root07, day 20	12	8	20	8	18	14	14	18	8
root09, day 8	4	2	2	2	4	2	2	2	2
root09, day 12	18	6	18	6	20	10	10	17	6
root09, day 16	24	6	18	6	24	4	6	16	6
root09, day 20	14	6	22	6	29	6	12	24	6
root12, day 8	4	2	2	2	4	2	2	3	2
root12, day 12	10	4	8	6	4	4	4	4	6
root12, day 16	18	6	14	6	20	4	4	14	6
root12, day 20	28	12	46	14	54	12	14	40	18
root17, day 12	4	2	8	2	6	2	2	4	2
root17, day 16	14	6	10	6	14	6	6	10	6
root17, day 20	10	6	12	6	14	4	6	12	6
root19, day 8	4	6	2	2	10	2	2	4	2
root19, day 12	4	4	6	4	6	2	4	4	4
root19, day 16	14	8	18	8	18	6	10	12	10
root19, day 20	38	14	42	14	48	12	14	28	14
root20, day 8	2	2	4	2	2	2	2	2	2
root20, day 12	8	4	6	4	6	2	2	3	4
root20, day 16	14	4	14	4	18	2	6	13	4
root20, day 20	30	12	20	10	20	8	12	12	10
root24, day 12	6	2	6	2	8	2	2	5	2
root24, day 16	10	4	10	4	14	4	6	8	4
root24, day 20	20	10	24	10	22	8	8	13	8
Sum	478	184	524	188	609	174	218	407	196
Reduction		61.5%		64.1%		71.4%	64.2%		

6.2 Overlapping Branches

The roots are imaged as a projection of a 3D structure (the root) to the 2D image space. Branches may therefore overlap in the 2D image. One major advantage when analyzing roots based on Reeb graphs is posed by the ability to immediately distinguish between branching points and overlaps in the root structure [18]. An overlap introduces a cycle in the graph and therefore a saddle node of type merge in the set of critical points. For a medial axis based graph there are only two types of nodes: endpoints and branching points. Therefore a merge of the foreground structure can not be immediately distinguished from a split of the structure based on the type of

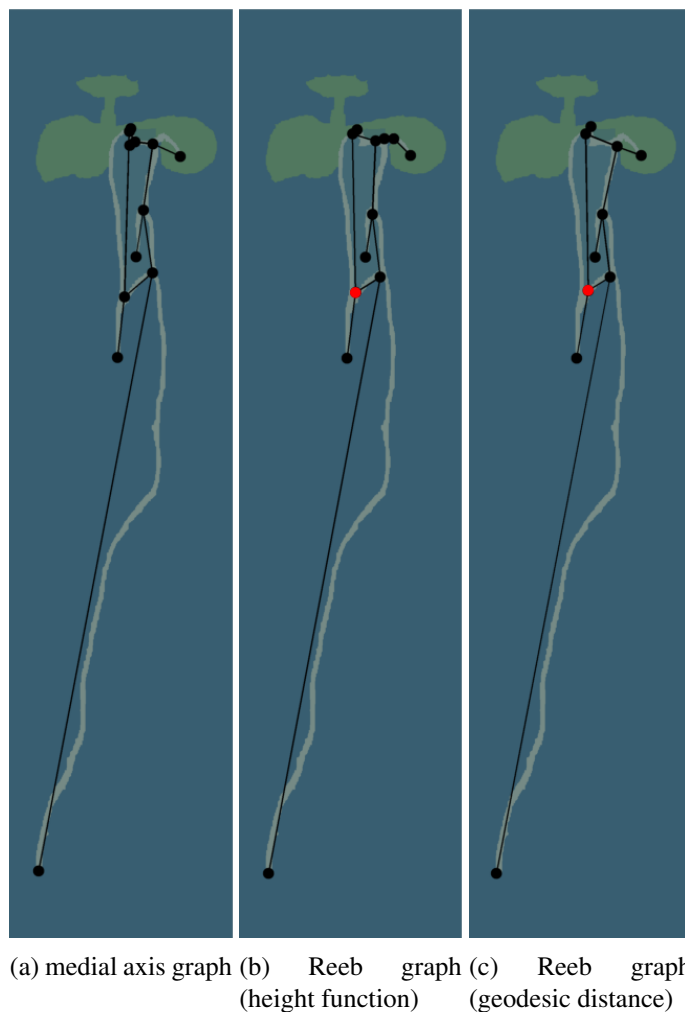


Figure 6.4: Resulting graphs for root 12, day 20: branches overlap in the image and form a cycle in the graphs. The saddle node of type merge is marked in red in the two Reeb graphs.

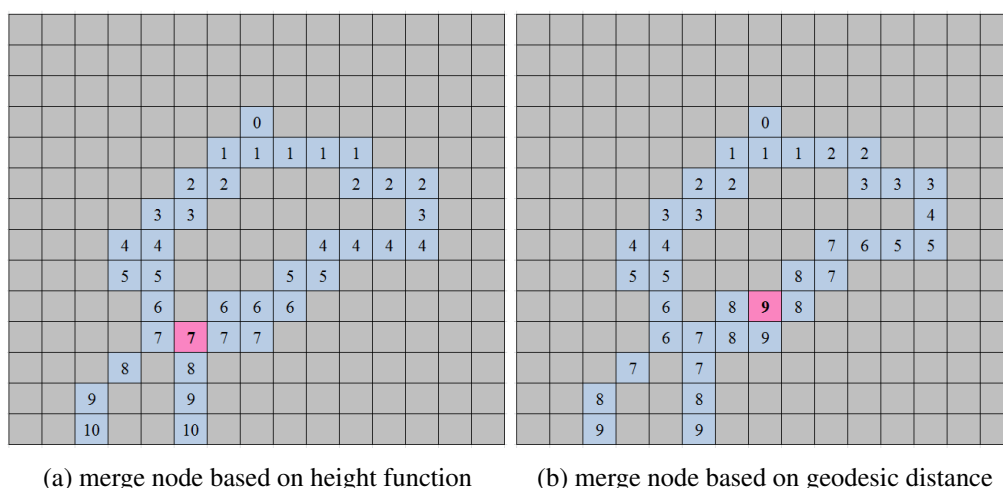


Figure 6.5: Comparison of position of merge nodes in the two Reeb graphs.

node in the graph.

Figure 6.4 shows an example of such an overlap of branches and the computed graphs for this image. There is a cycle in all three images, however only in the Reeb graph representations the overlap is explicitly marked by the merge node (in Figure 6.4 marked in red).

For Reeb graph representations based on the geodesic distance, the saddle node of type merge may not always be at/near the location of the actual overlap of branches as it is computed at the location of a merge of connected components with the same distance. As illustrated in Figure 6.5b this position can be inside on of the branches forming the cycle. For the Reeb graph based on the height function, the merge node is locate as shown in Figure 6.5a. However, for the height function saddle nodes of type merge may be introduced without any cycle in the graph. For the height function more than one maximum node may appear in the graph, these maximum nodes are connected to the foreground structure by a saddle node of type merge , without any cycle in the graph (respectively hole in the foreground structure).

6.3 Comparison: Root System Analyser

The root dataset was evaluated using the tool “Root System Analyser” (RSA) [24] as well. The number of nodes computed by RSA is listed in Table 6.2 in column “RSA”. Although a smoothing is applied, a large number of nodes that represent noise is kept by this approach. This is indicated by the high number of nodes needed for the root dataset (compare: 407 nodes RSA to 188 nodes geodesic distance Reeb graph with pruning to 196 nodes human ground truth). Nodes of degree 2 that have no influence on the topology of the foreground structure are kept in the representation as well.

The representation shown in Figure 6.6 was computed using RSA. The main branches of the root are well represented, some small branches due to noise are kept in the final representation. While both Reeb graph representations presented in this thesis use 6 nodes and 3 branches (see

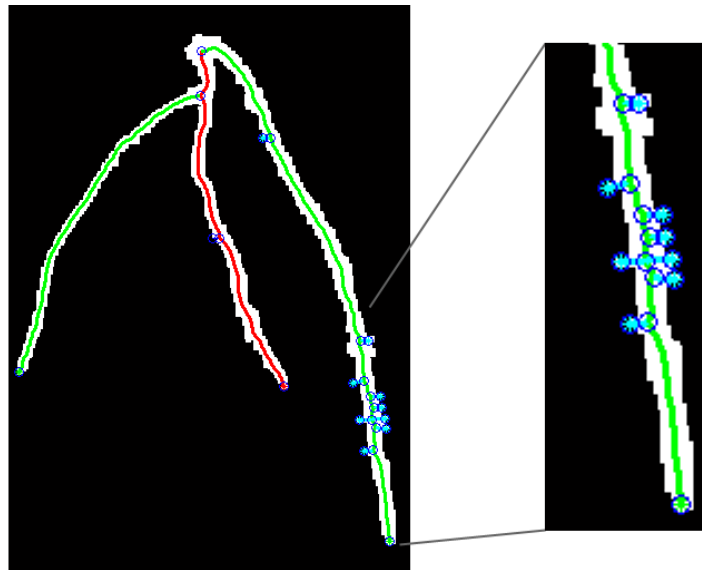


Figure 6.6: Result for root09 day 20, computed using the “Root System Analyser”. Some small branches due to noise are kept in the final representation.

Figure 6.2) which conforms to the human ground truth, the RSA uses 23 nodes and detects 12 branches.

Figure 6.7 shows a comparison of the representation computed by RSA and a representation computed using the geodesic distance based Reeb graph approach as presented in this thesis. The RSA representation (Figure 6.7a) detects 16 branches, using 40 nodes for the representation. The Reeb graph representation based on the geodesic distance with pruning (Figure 6.7b), needs 14 nodes for the representation and computes a total of 6 branches. However, two small branches are wrongly discarded during the graph pruning as is the end part of one of the overlapping branches. Just as the Reeb graph based representations, the Root System Analyser Tool detects overlaps of branches in the image and tries to reestablish the correct connectivity of the branches. However, in this process incorrect decisions are likely to be taken. For Figure 6.7a this is shown as a detail. The overlap of two branches is split into four branches here, none of them overlap, but two of them share an endpoint.

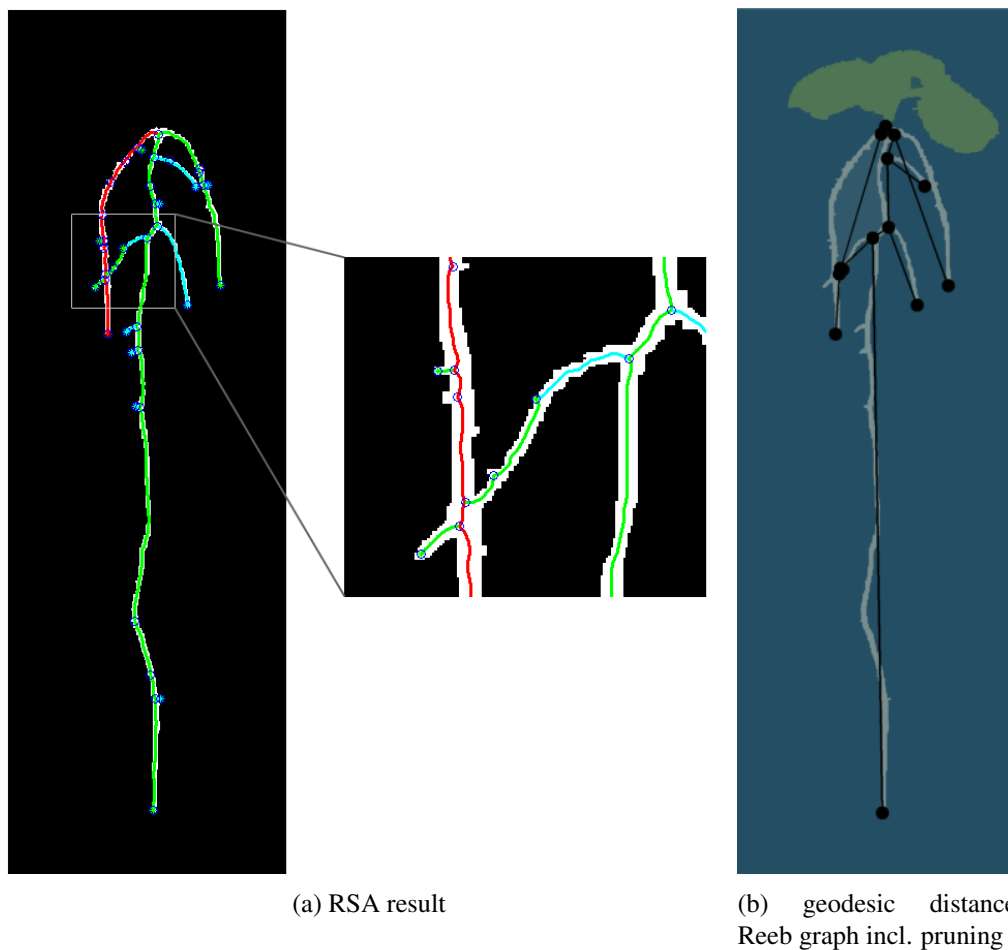


Figure 6.7: Results for root12, day 20. While some minor branches due to noise are wrongly kept in the final representation of the RSA tool, three small true branches are wrongly discarded in the pruning of the Reeb graph approach. The overlap of branches is wrongly dissolved by the RSA tool.

A Normalized Root Representation

The three graph representations provide different characteristics of the root. The length of a root in the Reeb graph based on the height function is determined by the height of the endpoint of a root in the image. Additional length of the root due to curvature is not taken into account. This is however done for the geodesic distance, as the length is measured as the intrinsic length of the root structure. Compared to the Reeb graphs used in this thesis, the medial axis representation is the only representation, that captures the width of the roots. In order to combine the advantages of several representations, a measurement of equality of graph representations is needed. Based on this graph equality, the characteristics of various representations can be collected for a combined description of the root.

7.1 Equality of Graphs

Within the scope of this thesis two graph representations of the same image are considered equal, if the graphs are isomorphic. Two representations that are not equal (their graphs are not isomorphic) can be compared using the distance between these graphs. This distance can be measured based on the maximal common subgraph, as presented in [6]. For two graphs G_1 and G_2 and their maximal common subgraph G_{mcs} the number of nodes in these graphs is denoted as $|G_1|$ respectively $|G_2|$ and $|G_{mcs}|$. The distance between the graphs G_1 and G_2 is defined as:

$$d(G_1, G_2) = 1 - \frac{|G_{mcs}|}{\max(|G_1|, |G_2|)}. \quad (7.1)$$

The definition of subgraph and subgraph isomorphism needs to be further specified respectively altered for the scope of this thesis:

A general subgraph is a graph built from the subset of the vertices and corresponding edges of a graph. Within this thesis nodes of degree 2 are smoothed out of the graph. A subgraph is therefore derived as a general subgraph which is altered by smoothing of nodes. Figure 7.1 shows the process of deriving such a subgraph.

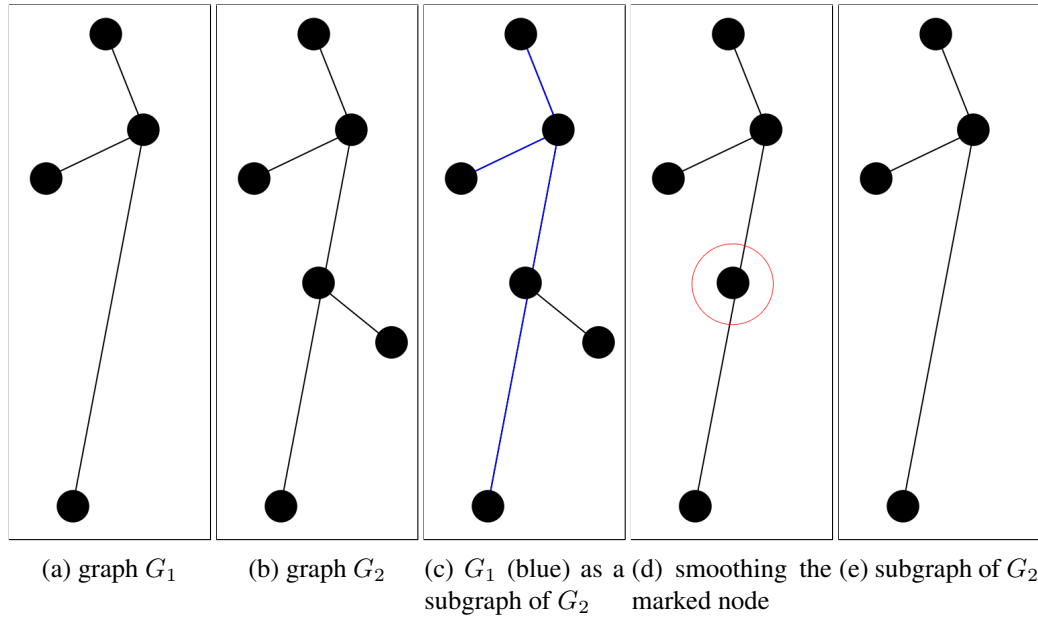


Figure 7.1: Based on the definition in this thesis graph G_1 is a subgraph of G_2 . G_2 is subgraph isomorphic to G_1

Based on this definition of a subgraph and the distance measurement given in equation 7.1, the distance between the graphs G_1 and G_2 in Figure 7.1 is given as:

$$d(G_1, G_2) = 1 - \frac{|G_1|}{|G_2|} = 1 - \frac{4}{6} = 0.33 .$$

Graph isomorphism respectively subgraph isomorphism (using the subgraph definition including smoothing) were used to compare and evaluate the graph representations of the root dataset. Table 7.1 shows this comparison. For all images in the root dataset two representations are in each case compared. Graph isomorphism is indicated by the sign \checkmark in Table 7.1. All pairs of graphs that are not isomorphic, share a subgraph. If one of the graphs already forms such a subgraph the acronym of this graph is given in the table cell. If both graphs contain isomorphic subgraphs, the according cell of the table is labeled “both”.

The abbreviations used in Table 7.1 and Table 7.2:

- RG1 + P: Reeb graph based on height function including pruning
- RG2 + P: Reeb graph based on geodesic distance including pruning
- MA + P: Medial axis graph including pruning
- MA + WP: Medial axis graph including width based pruning

Table 7.1: Graph isomorphism and isomorphic subgraphs in the root dataset.

Equality of graphs						
Image	RG1+P / RG2+P	RG1+P / MA+P	RG1+P / MA+WP	RG2+P / MA+P	RG2+P / MA+WP	MA+P / MA+WP
root04, day 8	✓	✓	✓	✓	✓	✓
root04, day 12	✓	MA+P	both	MA+P	both	MA+P
root04, day 16	✓	MA+P	both	MA+P	both	MA+P
root04, day 20	✓	MA+P	MA+WP	MA+P	MA+WP	✓
root05, day 8	R2+P	✓	R1+P	R2+P	✓	MA+P
root05, day 12	✓	R1+P	R1+P	R2+P	R2+P	MA+P
root05, day 16	R1+P	✓	✓	MA+P	MA+WP	✓
root05, day 20	✓	MA+P	R1+P	MA+P	R2+P	MA+P
root07, day 8	✓	R1+P	R1+P	R2+P	R2+P	✓
root07, day 12	✓	R1+P	R1+P	R2+P	R2+P	✓
root07, day 16	✓	both	both	both	both	✓
root07, day 20	✓	R1+P	R1+P	R2+P	R2+P	✓
root09, day 8	✓	✓	✓	✓	✓	✓
root09, day 12	both	both	both	R1+P	R2+P	✓
root09, day 16	✓	MA+P	✓	MA+P	✓	MA+P
root09, day 20	✓	MA+P	R1+P	MA+P	R2+P	MA+P
root12, day 8	✓	✓	✓	✓	✓	✓
root12, day 12	R1+P	both	both	MA+P	MA+WP	✓
root12, day 16	✓	MA+P	MA+WP	MA+P	MA+WP	✓
root12, day 20	R1+P	both	both	both	both	MA+P
root17, day 12	✓	✓	✓	✓	✓	✓
root17, day 16	✓	✓	✓	✓	✓	✓
root17, day 20	✓	MA+P	both	MA+P	both	MA+P
root19, day 8	✓	✓	✓	✓	✓	✓
root19, day 12	✓	MA+P	✓	MA+P	✓	MA+P
root19, day 16	✓	MA+P	both	MA+P	both	MA+P
root19, day 20	both	MA+P	✓	MA+P	both	MA+P
root20, day 8	✓	✓	✓	✓	✓	✓
root20, day 12	✓	MA+P	MA+WP	MA+P	MA+WP	✓
root20, day 16	✓	MA+P	R1+P	MA+P	R2+P	MA+P
root20, day 20	R2+P	R1+P	both	R2+P	R2+P	MA+P
root24, day 12	✓	✓	✓	✓	✓	✓
root24, day 16	✓	✓	R1+P	✓	R2+P	MA+P
root24, day 20	✓	MA+P	MA+WP	MA+P	MA+WP	✓

Table 7.2: Distances between the graph representations

Distances between the graphs						
Image	RG1+P / RG2+P	RG1+P / MA+P	RG1+P / MA+WP	RG2+P / MA+P	RG2+P / MA+WP	MA+P / MA+WP
root04, day 8	0.00	0.00	0.00	0.00	0.00	0.00
root04, day 12	0.00	0.50	0.50	0.50	0.50	0.50
root04, day 16	0.00	0.50	0.63	0.50	0.63	0.63
root04, day 20	0.00	0.25	0.25	0.25	0.25	0.00
root05, day 8	0.50	0.00	0.50	0.50	0.00	0.50
root05, day 12	0.00	0.33	0.50	0.33	0.50	0.25
root05, day 16	0.33	0.00	0.00	0.33	0.33	0.00
root05, day 20	0.00	0.25	0.20	0.25	0.20	0.40
root07, day 8	0.00	0.50	0.50	0.50	0.50	0.00
root07, day 12	0.00	0.25	0.25	0.25	0.25	0.00
root07, day 16	0.00	0.50	0.50	0.50	0.50	0.00
root07, day 20	0.00	0.43	0.43	0.43	0.43	0.00
root09, day 8	0.00	0.00	0.00	0.00	0.00	0.00
root09, day 12	0.50	0.70	0.70	0.40	0.40	0.00
root09, day 16	0.00	0.33	0.00	0.33	0.00	0.33
root09, day 20	0.00	0.33	0.50	0.33	0.50	0.50
root12, day 8	0.00	0.00	0.00	0.00	0.00	0.00
root12, day 12	0.33	0.50	0.50	0.33	0.33	0.00
root12, day 16	0.00	0.33	0.33	0.33	0.33	0.00
root12, day 20	0.14	0.33	0.43	0.50	0.50	0.14
root17, day 12	0.00	0.00	0.00	0.00	0.00	0.00
root17, day 16	0.00	0.00	0.00	0.00	0.00	0.00
root17, day 20	0.00	0.33	0.33	0.33	0.33	0.33
root19, day 8	0.00	0.00	0.00	0.00	0.00	0.00
root19, day 12	0.00	0.50	0.00	0.50	0.00	0.50
root19, day 16	0.00	0.25	0.40	0.25	0.40	0.40
root19, day 20	0.14	0.14	0.00	0.14	0.14	0.14
root20, day 8	0.00	0.00	0.00	0.00	0.00	0.00
root20, day 12	0.00	0.50	0.50	0.50	0.50	0.00
root20, day 16	0.00	0.50	0.33	0.50	0.33	0.66
root20, day 20	0.17	0.33	0.17	0.17	0.17	0.33
root24, day 12	0.00	0.00	0.00	0.00	0.00	0.00
root24, day 16	0.00	0.00	0.33	0.00	0.33	0.33
root24, day 20	0.00	0.20	0.20	0.20	0.20	0.00
Average	0.06	0.25	0.26	0.25	0.25	0.17

For all pairs of graphs in Table 7.1 the distance between the graphs was computed as well. These distances are given in Table 7.2. The distances of all subgraph isomorphic graphs are computed based on the distance definition given before. A distance of 0 indicates graph isomorphism.

7.2 Combined Root Representation

As the root images are taken as time series, several images of a root on different days of the growth cycle can be used for a normalized representation. Therefore the images will be aligned based on the starting point of the latest image in a time series and orientated according to the longest edge from the starting point of the latest image.

In order to combine the characteristics of the three graph based representations, corresponding nodes in the graphs have to be obtained by matching of the graphs as discussed in Section 7.1. The graph is built based on one representation, here the geodesic distance based Reeb graph is a good choice, as this representation approximates the ground truth best (see Chapter 6.1). As additional information the number of nodes, branches, and the width and length of branches can be provided. Figure 7.2 shows an example for such a representation.

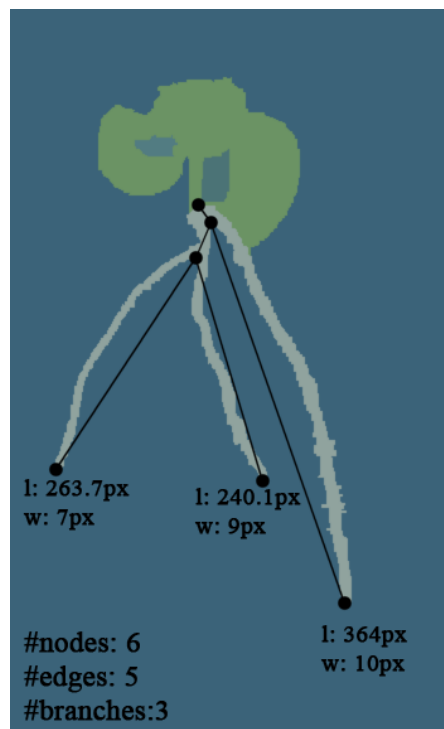


Figure 7.2: Combined Characteristics for root09, day 20.

Conclusion

In this thesis the possibility of a Reeb graph based representation of roots was inquired. Intermediate results on the representation of root structures by Reeb graphs have been published in the Computer Vision Winter Workshop 2014 [18].

For phenotyping of plants the length of branches, numbers of branches in a root structure and the width of branches are analyzed characteristics. All these traits are captured by the presented graph representations.

Two Reeb graph based approaches as well as an approach based on the medial axis were evaluated and compared to human generated ground truth. For all three approaches the graphs are based on a set of nodes and the adjacency between the nodes. The nodes are obtained by either analyzing the shape according to a (Morse) function to obtain the critical points that form the nodes in the Reeb graphs or by computing a medial axis skeleton and detecting branching points and endpoints, that form the nodes of the medial axis graph. For the Reeb graph based approaches two Morse functions were used: the height function and the geodesic distance. According to the nodes, the connections in the graphs are obtained by analyzing the structure. The root is traced from one critical point to the next in order to compute the adjacency of the nodes. Reeb graphs according to a height function are able to describe a root correctly, if its general direction does not strongly deviate from the vertical direction. The length of branches in such a Reeb graph is measured considering the height function. Additional length in a branch due to curvature of the branch is not taken into account. Therefore Reeb graphs according to the geodesic distance proved to be better suited for the task of length measurement, as in this case the intrinsic length of the root structure is measured. Reeb graphs according to the geodesic distance captured the branching pattern of the root as well as the length of individual branches. The width of a branch can be measured using a medial axis transformation. Although a graph can be built based on the medial axis as well, a combination of geodesic distance Reeb graph and medial axis should be used to correctly describe the root, as the medial axis graph (especially in combination with graph pruning) does not capture all branching points correctly.

All three representations are very sensitive to noise in the segmented image and introduce spurious branches in the graphs. These spurious branches are discarded in a graph pruning process.

However when choosing a parameter for graph pruning that removes the spurious branches, some true branches may be discarded as well.

As the roots are imaged through the projection of the 3D structure (the root) to the 2D image space, branches may overlap in the image. These overlaps introduce a cycle in all graph representations. One major advantage when using a Reeb graph based representation compared to a medial axis representation is, that due to the different types of nodes in a Reeb graph, an overlap generates a merge node in the graph. Therefore an overlap can be immediately detected based on the type of the corresponding node in the graph.

The results show that the Reeb graph based on the geodesic distance approximates the human generated ground truth best, regarding the number of nodes in the representation as well as the structure of the graph generated.

To include parameters from several representations, as for example the width of branches based on the medial axis, the Reeb graph can be combined with the medial axis representation. In order to allow for this combination of different graphs based representations, a distance based equality measurement was introduced. For isomorphic graphs the characteristics of corresponding nodes can be combined immediately, for isomorphic subgraphs this is only possible for the nodes of the subgraphs.

For future work the segmentation that is done as a preprocessing step proved to be a major drawback in the attempt to automatically analyze large datasets. The semi-automatic segmentation method used for the creation of the root dataset presented in this thesis, needs to be changed to either a fully automatic approach or a semi-automatic one, that is optimized with respect to computation time and needed user interaction. This would reduce the time and human input needed for the preprocessing step. Moreover the current segmentation approach introduced artefacts in the form of frayed borders, that generate spurious branches in the graphs. Although these spurious branches can be discarded using graph pruning, true branches may be removed in the graph pruning process as well. A reduction of this noise by either using a better suited segmentation method or by application of smoothing on the segmentation result may reduce this problem.

For future work the temporal correlation of the root images should be taken into account. The roots are imaged on several days of their growth cycle, therefore the development of a route in a certain time frame is captured by such an image sequence. To distinguish a small true branch from a spurious branch may for example not be possible based on one image. A later image of the same root may help with this decision: in case there is a branch at this position in a later image, the branch is detected as a true branch, otherwise it is discarded as a spurious branch. Thus, the information and knowledge about a root gained on the images of later days in the growth cycle can be used for decisions on earlier images.

List of Figures

2.1	Shape representation: skeleton	5
2.2	Graphs on image pixels.	6
2.3	Two examples for graph based image representation.	6
2.4	Operations on graphs. The top half shows the original graphs with changing parts labeled blue, the bottom half of the images show the altered graphs.	7
2.5	Definition of subgraphs and graph isomorphism	8
2.6	Computation of a medial axis by inscription of maximal circles.	8
2.7	Distance transformation computed for two shapes. Red indicates high distance values, blue indicate low distance values.	9
2.8	Graph representation of a shape derived based on the medial axis skeleton.	9
2.9	Critical points computed based on the height function (downwards) and corresponding Reeb graph. The white image region shows the foreground region described by the Reeb graph, black parts are background.	11
2.10	Example Morse functions	12
2.11	Reeb graph, computed for the white foreground region, according to height function (going downwards) and the geodesic distance (the source pixel is located in the center of the top foreground pixel line). Here both Morse functions generate identical critical point.	13
3.1	Images A-D show root input images, colors indicate the age of the root, with red labeling the oldest parts of the root. The corresponding graph representation of image A is shown in image E, individual branches of the root system are marked in the images F-H. Image taken from [24].	16
3.2	User interface of the “Root System Analyser”. Image taken from [24].	17
3.3	Segmentation process to separate foreground (root) from background. Image taken from [37].	18
3.4	Reeb graph based skeletonization result of the input data in gray. Image taken from [13].	20
3.5	Reeb graph based skeletonization (left to right): input points in yellow - augmented Reeb graph - iterative smoothing - post-processing: adding missing links (top) and removing spurious branches and loops (bottom). Image taken from [13].	20
3.6	Segmentation results of simulated scans. Image taken from [35].	22
		55

4.1	Plants grown inside a petri dish, picture on day 20 of their growth cycle. The ruler at the bottom of the image is used as a reference marker for measurements on the image.	24
4.2	Example images of root dataset	24
4.3	Segemented images of root004, day 4 to day 20.	26
5.1	Example root with characteristic locations labeled and a possible graph representation.	27
5.2	Implementation pipeline for the three approaches.	28
5.3	Pixel neighborhood.	29
5.4	Branching point in a medial axis skeleton marked red, endpoint marked pink. . . .	30
5.5	Four different types of critical points, computed according to the height function [18].	30
5.6	For critical points on the same height a unique Reeb graph can not be built, as the connections of the nodes in the graph are ambiguous [18].	31
5.7	Computation of edges in Reeb graph	32
5.8	Geodesic distances are computed using the chessboard distance, the source pixel is located in the top right corner. At distance 8 the foreground is split into two connected components, this would result in a critical point of type split at distance 7.	33
5.9	Geodesic distance computed as chessboard distance from the center pixel leads to concentric squares. The euclidean distance is described by the concentric circles. .	34
5.10	Example for the computation of edges in the geodesic distance Reeb graph. The connection between critical point P_2 and P_1 is built.	35
6.1	Resulting graphs for root 09, day 20.	37
6.2	Resulting graphs for root 09, day 20 - length based graph pruning was applied to the results shown in Figure 6.1. For the medial axis based representation the width pruning approach was applied as well.	38
6.3	Numbers of nodes used in the graphs compared to human ground truth.	40
6.4	Resulting graphs for root 12, day 20: branches overlap in the image and form a cycle in the graphs. The saddle node of type merge is marked in red in the two Reeb graphs.	42
6.5	Comparison of position of merge nodes in the two Reeb graphs.	43
6.6	Result for root09 day 20, computed using the “Root System Analyser”. Some small branches due to noise are kept in the final representation.	44
6.7	Results for root12, day 20. While some minor branches due to noise are wrongly kept in the final representation of the RSA tool, three small true branches are wrongly discarded in the pruning of the Reeb graph approach. The overlap of branches is wrongly dissolved by the RSA tool.	45
7.1	Based on the definition in this thesis graph G_1 is a subgraph of G_2 . G_2 is subgraph isomorphic to G_1	48
7.2	Combined Characteristics for root09, day 20.	51

List of Tables

3.1	Available root system analysis and measurement tools.	16
6.1	Wrong decisions on graph pruning	38
6.2	Numbers of nodes in the graphs	41
7.1	Equality of graphs	49
7.2	Distances between graphs	50

Bibliography

- [1] Patrick Armengaud, Kevin Zambaux, Adrian Hills, Ronan Sulpice, Richard J. Pattison, Michael R. Blatt, and Anna Amtmann. Ez-rhizo: integrated software for the fast and accurate measurement of root system architecture. *The Plant Journal*, 57(5):945–956, 2009.
- [2] J.-L. Arsenault, C. Messier S. Pouleur, and R. Guay. WinrhizoTM, a root-measuring system with a unique overlap correction method. *HortScience*, 30(906), 1995.
- [3] Stefano Berretti, Alberto Del Bimbo, and Pietro Pala. 3D mesh decomposition using Reeb graphs. *Image and Vision Computing*, 27(10):1540–1554, September 2009.
- [4] S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno. Reeb graphs for shape analysis and applications. *Theoretical Computer Science*, 392(1–3):5–22, February 2008.
- [5] Harry Blum. A Transformation for Extracting New Descriptors of Shape. In Weiant Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967.
- [6] Horst Bunke and Kim Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3–4):255–259, March 1998.
- [7] Randy T. Clark, Adam N. Famoso, Keyan Zhao, Jon E. Shaff, Eric J. Craft, Carlos D. Bustamante, Susan R. McCouch, Daniel J. Aneshansley, and Leon V. Kochian. High-throughput two-dimensional root system phenotyping platform facilitates genetic analysis of root growth and development. *Plant, Cell & Environment*, 36(2):454–466, 2013.
- [8] Rany T. Clark, Robert B. MacCurdy, Janelle K. Jung, Jon E. Shaff, Susan R. McCouch, Daniel J. Aneshansley, and Leon V. Kochian. Three-dimensional root phenotyping with a novel imaging and software platform. *Plant Physiology*, 156(2):455–465, 2011.
- [9] Harish Doraiswamy and Vijay Natarajan. Efficient algorithms for computing Reeb graphs. *Computational Geometry*, 42(6–7):606–616, August 2009.
- [10] Rachid EL Khoury, Jean-Philippe Vandeborre, and Mohamed Daoudi. 3D mesh Reeb graph computation using commute-time and diffusion distances. In *Proceedings SPIE: Three-Dimensional Image Processing (3DIP) and Applications II*, volume 8290, pages 82900H–82900H–10, 2012.

- [11] Andrew French, Susana Ubeda-Tomás, Tara J. Holman, Malcolm J. Bennett, and Tony Pridmore. High-throughput quantification of root growth using a novel image-analysis tool. *Plant Physiology*, 150(4):1784–1795, 2009.
- [12] Taras Galkovskyi, Yuriy Mileyko, Alexander Bucksch, Brad Moore, Olga Symonova, Charles Price, Christopher Topp, Anjali Iyer-Pascuzzi, Paul Zurek, Suqin Fang, John Harer, Philip Benfey, and Joshua Weitz. GiA roots: software for the high throughput analysis of plant root system architecture. *BMC Plant Biology*, 12(1):116, 2012.
- [13] Xiaoyin Ge, Issam I. Safa, Mikhail Belkin, and Yusu Wang. Data skeletonization via Reeb graphs. In J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 837–845, 2011.
- [14] Michael Gerstmayer, Yll Haxhimusa, and Walter Kropatsch. Hierarchical interactive image segmentation using irregular pyramids. In Xiaoyi Jiang, Miquel Ferrer, and Andrea Torsello, editors, *Graph-Based Representations in Pattern Recognition*, volume 6658 of *Lecture Notes in Computer Science*, pages 245–254. Springer Berlin / Heidelberg, 2011.
- [15] William Harvey, Yusu Wang, and Rephael Wenger. A randomized $O(m \log m)$ time algorithm for computing reeb graphs of arbitrary simplicial complexes. In *Proceedings of the twenty-sixth annual symposium on Computational geometry*, pages 267–276. ACM, 2010.
- [16] Makoto Hayashi and Mikio Nishimura. Arabidopsis thaliana - a model organism to study plant peroxisomes. *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research*, 1763(12):1382–1391, December 2006.
- [17] David Houle, Diddahally R. Govindaraju, and Stig Omholt. Phenomics: the next challenge. *Nat Rev Genet*, 11(12):855–866, December 2010.
- [18] I. Janusch, W. G. Kropatsch, and W. Busch. Reeb graph based examination of root development. In *Proceedings of the 19th Computer Vision Winter Workshop*, pages 43–50, Feb 2014.
- [19] W. Johannsen. The genotype conception of heredity. *The American Naturalist*, 45(531):pp. 129–159, 1911.
- [20] Walter Kropatsch, Yll Haxhimusa, and Adrian Ion. Multiresolution image segmentations in graph pyramids. In Abraham Kandel, Horst Bunke, and Mark Last, editors, *Applied Graph Theory in Computer Vision and Pattern Recognition*, volume 52 of *Studies in Computational Intelligence*, pages 3–41. Springer Berlin / Heidelberg, 2007.
- [21] Louisa Lam, Seong-Whan Lee, and Ching Y Suen. Thinning methodologies-a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence*, 14(9):869–885, 1992.
- [22] Jacques Le Bot, Valérie Serra, José Fabre, Xavier Draye, Stéphane Adamowicz, and Loïc Pagès. Dart: a software to analyse root system architecture and development from captured images. *Plant and Soil*, 326(1-2):261–273, 2010.

- [23] D. T. Lee. Medial axis transformation of a planar shape. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-4(4):363–369, July 1982.
- [24] Daniel Leitner, Bernd Felderer, Peter Vontobel, and Andrea Schnepf. Recovering root system traits using image analysis exemplified by two-dimensional neutron radiography images of lupine. *Plant Physiology*, 164(1):24–35, 2014.
- [25] Guillaume Lobet, Loïc Pagès, and Xavier Draye. A novel image-analysis toolbox enabling quantitative analysis of root system architecture. *Plant Physiology*, 157(1):29–39, 2011.
- [26] Stefan Mairhofer, Susan Zappala, Saoirse R. Tracy, Craig Sturrock, Malcolm Bennett, Sacha J. Mooney, and Tony Pridmore. Roottrak: Automated recovery of three-dimensional plant root architecture in soil from x-ray microcomputed tomography images using visual tracking. *Plant Physiology*, 158(2):561–569, 2012.
- [27] Ilona Miko. Genetic dominance: genotype-phenotype relationships. *Nature Education*, 1(1):140, 2008.
- [28] Kerstin A. Nagel, Alexander Putz, Frank Gilmer, Kathrin Heinz, Andreas Fischbach, Johannes Pfeifer, Marc Faget, Stephan Blossfeld, Michaela Ernst, Chryssa Dimaki, Bernd Kastenholz, Ann-Katrin Kleinert, Anna Galinski, Hanno Scharr, Fabio Fiorani, and Ulrich Schurr. Growscreen-rhizo is a novel phenotyping robot enabling simultaneous measurements of root and shoot growth for plants grown in soil-filled rhizotrons. *Functional Plant Biology*, 2012.
- [29] Mattia Natali, Silvia Biasotti, Giuseppe Patanè, and Bianca Falcidieno. Graph-based representations of point clouds. *Graph. Models*, 73(5):151–164, September 2011.
- [30] Alain Pierret, Santimaitree Gonkhamdee, Christophe Jourdan, and Jean-Luc Maeght. Ij_rhizo: an open-source software to measure scanned images of root samples. *Plant and Soil*, 373(1-2):531–539, 2013.
- [31] Michael P. Pound, Andrew P. French, Jonathan A. Atkinson, Darren M. Wells, Malcolm J. Bennett, and Tony Pridmore. Rootnav: Navigating images of complex root architectures. *Plant Physiology*, 162(4):1802–1814, 2013.
- [32] Daniela Ristova, Ulises Rosas, Gabriel Krouk, Sandrine Ruffel, Kenneth D. Birnbaum, and Gloria M. Coruzzi Coruzzi. Rootscape: A landmark-based system for rapid screening of root architecture in arabidopsis. *Plant Physiology*, 161(3):1086–1096, 2013.
- [33] R. Sozzani and Benfey PN. High-throughput phenotyping of multicellular organisms: finding the link between genotype and phenotype. *Genome Biology*, 12(3):219, 2011.
- [34] James Stewart. *Calculus*. Cengage Learning Emea, 6th edition. international met edition, February 2008.

- [35] N. Werghi, Yijun Xiao, and J.P. Siebert. A functional-based segmentation of human body scans in arbitrary postures. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(1):153–165, 2006.
- [36] Guang Zeng, Stanley T. Birchfield, and Christina E. Wells. Automatic discrimination of fine roots in minirhizotron images. *New Phytologist*, 177(2):549–557, 2008.
- [37] Ying Zheng, S. Gu, H. Edelsbrunner, C. Tomasi, and P. Benfey. Detailed reconstruction of 3D plant root shape. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 2026–2033, 2011.