Reeb graph based examination of root development

Ines Janusch[1], Walter G. Kropatsch[1], and Wolfgang Busch[2]

[1] Vienna University of Technology
Institute of Computer Graphics and Algorithms
Pattern Recognition and Image Processing Group
Vienna, Austria

[2] Gregor Mendel Institute of Molecular Plant Biology
Austrian Academy of Sciences
Vienna, Austria

ines@prip.tuwien.ac.at, krw@prip.tuwien.ac.at, wolfgang.busch@gmi.oeaw.ac.at

Abstract *This paper presents an approach to analyze plant root development by means of topological image analysis. 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. 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. 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 representation of the underlying image data in this way. 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. This is not as easily possible by an analysis solely based on contours.*

1 Introduction

Reeb graphs are widely used as shape descriptors for 3D structures. [2] gives a general overview on the use of Reeb graphs for shape analysis. [10] uses Reeb graphs for a pose independent segmentation of 3D data of human body scans, while [8] provides a skeletal representation of point clouds based on Reeb graphs. As a representation of 2D data, Reeb graphs are for example used in [5] to provide a data skeletonization of the image content. However, Reeb graphs have not been applied to branched structures like roots or blood vessels although they pose a well suited representation. An analysis of branching patterns of roots based on a 3D reconstruction of the root architecture of rice plants is provided in [11].

One of the ultimate challenges of biology is posed by the question how genotypes translate into phenotypes. There, the major bottleneck lies in the ability to phenotype a large number of individuals and genotypes with high accuracy. This is particularly lagging in complex multicellular organisms such as plants, in which specific biological processes often occur only temporarily and are restricted to specific organs, tissues or even individual cells. Efficient and unsupervised image segmentation and the extraction of certain characteristics are a key in approaching this goal. 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.

When analyzing roots (for e.g. phenotyping), characteristics such as the number of branches or the position and number of branch-endings, are studied. These characteristics can be efficiently described by (Reeb) graphs. Reeb graphs describe changes in topology in the represented structure. Reeb graphs are based on Morse theory and analyze the (here) image content according to a function (Morse function).

When growing, roots change their shape, branches are formed - their topology changes. Moreover the projection of the 3D root structure to the 2D image data might cause overlaps of branches in the image. In a Reeb graph a distinction between a branch and an overlap is immediately possible as these changes in topology are captured by the graph.

The Reeb graph is used as a simplified, skeletal representation of the image data that captures the intrinsic topological structure of the data and allows for a comparison of the image content. Especially for the root dataset these comparisons allow for a description of the growth process: the roots are imaged on consecutive days through their growth period. In comparison with a simple standard skeletonization approach as for example the Medial axis transform, the skeleton derived by a Reeb graph not only describes characteristics of the image content (here branches of the roots) but captures the actual positions of these characteristics as well.

The paper is structured as follows: Section 2 gives an introduction to Reeb graphs, Section 3 describes the dataset used and Section 4 shows the computation of a Reeb graph on the root dataset. The need for modifications of the Reeb graphs and the types of modifications are discussed in Section 5, Section 6 shows evaluation results on the root dataset while a conclusion and a perspective to future work are given in Section 7.

2 Reeb Graphs and Morse Theory

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 [4]. 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 0 or if one of these partial derivatives does not exist [9].

Such a critical point can ~~either be~~ ~~a~~ degenerate or ~~a~~ non-degenerate ~~critical point~~. 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 $f : M_d \to \mathbb{R}$ 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 [3].

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 [2]. 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.

Reeb graphs are compact shape descriptors that preserve the topological characteristics of the described shape [2]. Vertices of the Reeb graph correspond to critical points of the function (points where the topology of M changes), edges describe topological persistence [2]. 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.

Reeb graphs are originally defined for the continuous space,

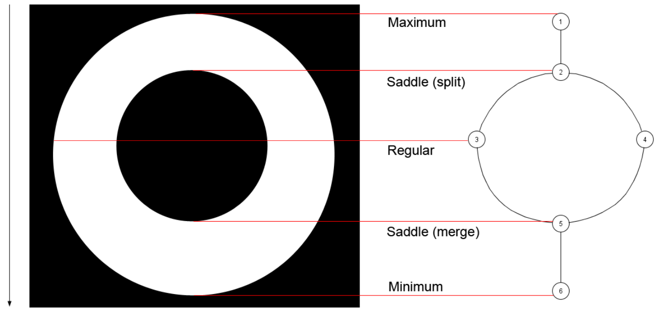


Figure 1: Critical points computed based on the height function and corresponding Reeb graph. The white image region shows the foreground region described by the Reeb graph, black parts are background.

but have been extended to the discrete domain: Here the Reeb graph is defined on a piecewise linear Morse function [4]. As the approach presented in this paper 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 ~~there~~ 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.

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 [10].

In 2D critical points and corresponding nodes in the Reeb graph are minima, maxima or saddles [3]. 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 ~~at~~ 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 [1]).

The approach described in the following sections uses the height function as Morse function μ. In 2D the height function is the function f that associates for each point $P = (x, y)$ the value y as the height of this point: $f(x, y) \mapsto y$.
Figure 1 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.

3 Root dataset

For the root 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 [6]. The plants are grown on a nutrient containing agar gel surface in plastic petri dishes that are vertically oriented. All plants in one plate belong to one dataset. One dataset/plate consists of 2 rows of 12 plants. The plates are placed in a growth chamber that allows for controlled conditions as constant temperature or humidity.
The images are taken using an image scanner. A special fixture allows for two datasets to be placed in an exact known position inside the scanner. 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 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.
In a preprocessing step the 24 plants per plate are cropped to single images: one image per plant with an image size in the range of 500x1300 to 800x1300 pixels resolution and a file size of 1,5-2,5Mbyte. Example images of this dataset are shown in Figure 2.

(a) (b) (c) (d)

Figure 2: Example images of the root dataset: root004 - (a) day 8; (b) day 12; (c) day 16; (d) day 20.

(a) (b) (c) (d)

Figure 3: Four different types of critical points computed according to the height function: (a) maximum / birth; (b) saddle (split); (c) minimum / death; (d) saddle (merge).

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 are analyzed, the other images are too early in the growth process and therefore to small in structure to be represented by a non-trivial Reeb graph.
All images analyzed are segmented in a preprocessing step and 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 the needed preceded segmentation, the dataset is restricted in its size, as the segmentation approach was done semi-automatically and required a lot of time (up to 1.5h for one image).

4 Computation of Reeb graphs

As the roots are imaged in their natural direction of growth (leaves in the top part of the image, roots growing downwards in a vertical direction) and branches occur mostly in this direction of growth, the height function is a suitable measuring function. Critical points indicate a change in topology, therefore they might only appear on the border of a region but not within the region. The borders of flat-regions in the image are analyzed to locate these critical points.
Figure 3 shows the four different types of critical points that are computed for the image content using a height function.

To compute the critical points a segmentation of the image needs to be done during a preprocessing step. As the height function is used to compute the critical points, the foreground region borders are analyzed with regard to horizontal borders as these might describe a change in the number of components. The so found critical points are located at the center of such a horizontal border.

There are two main problems encountered using this approach:

4.1 Critical points at same height

Due to the resolution of the image, the discretization of the root and further distortions during the segmentation process, it is possible that several critical points at different horizontal positions in the image are at the same vertical position (same height) in the image (see Figure 5(a) for an example). In this case the second criteria of Morse theory (see Section

2) is not met. A Reeb graph cannot be built, as a decision on how to connect the nodes in order to build the graph cannot be taken. Figure 4 shows an example: The solid lines illustrate the only two fixed connections in this example, the dashed lines indicate all possible connections. In this Reeb graph four edges are needed: one from each black (maximum) node to a red (saddle) node and one edge from a red (saddle) node to the green (minimum) node. A decision concerning these connections needs to be taken for the black center node as well as for the two red nodes. A solution to build a Reeb graph, despite several critical points at the same height, is discussed in Section 5.1

4.2 Additional critical points

Because of the segmentation prior to the computation of the critical points, segmentation artefacts appear in the images. The most common problem are frayed borders of image regions (see Figure 5(b) for an example). Especially for images of day 16 the segmentation creates noise and distorted region borders. When analyzing the images of day 16 one notices a high humidity between the plates in the form of water drops, which creates a highly texturized background that complicates the segmentation.

These frayed borders in the segmented images result in additional critical points that describe no actual split or merge of the root structure. These artefacts alter the Reeb graph and complicate a comparison or matching of graphs. One possibility on how to deal with these additional critical points is described in Section 5.2.

5 Modifications on the graphs

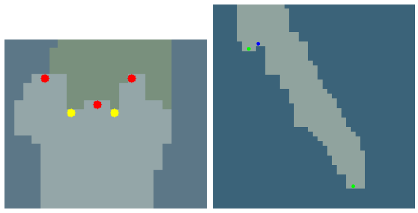
To overcome the problems discussed in Section 4.1 and Section 4.2 the following techniques were used:

5.1 Controlled shift of critical point coordinates

Due to the discrete pixel-space, the coordinates x and y of a pixel $p = (x, y)$ are integers. Critical points at the same height (same y-coordinate) occur for 35% of all images in the root dataset and are shifted. The height of such critical points is changed by an added factor f, $0 \leq f < 1$. A critical point $p = (x, y)$ is shifted to $p' = (x, y + f)$, f is computed using the following formula: $f = \frac{1}{w} \cdot (x-1)$, with w giving the width of the image. The y-coordinate is thereby changed from an integer to a floating-point number. Critical points at the same height are moved downwards in a left-to-right order, thus for two critical points $p_1 = (x_1, y)$ and $p_2 = (x_2, y)$ with $x_1 < x_2$, it is valid that, after shifting the points to $p'_1 = (x_1, y_1)$ and $p'_2 = (x_2, y_2)$, $y_1 < y_2$ holds. The actual order of heights is preserved by this correction procedure as only critical points that were primarily at the same height are changed. All critical points are at different heights, although when rounding down the y-coordinate of the critical points to an integer, they stay in the actual pixel line. A Reeb graph can therefore be built.

It is important to shift the heights in a fixed approach. A random decision choosing one of two critical points at the same height when building the Reeb graph cannot be used, as the results may vary with repeated tests. Reeb graphs built on such random decisions are not unique and therefore useless for e.g. comparison of two images.

Figure 6 shows a Reeb graph built on the marked critical points / nodes. Compared to Figure 4 where there are several critical points at the same height, Figure 6 shows critical points on different heights. The connections in this graph are unique. By shifting the critical points in Figure 4 according to the approach described in this section, the critical points are shifted to a configuration similar to the one shown in Figure 6.



(a) (b)

Figure 5: Problems encountered on the root dataset: (a) several critical points on same height; (b) frayed borders due to segmentation artifacts.

5.2 Graph pruning

Due to the segmentation done as a preprocessing step, segmentation artefacts falsify the number of critical points and therefore the number of nodes and edges in the Reeb graph.

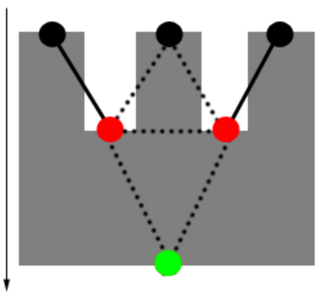


Figure 4: Critical points at same height: the solid lines show connections that are fixed, dashed lines indicate all possible connections - a decision needs to be taken for these.

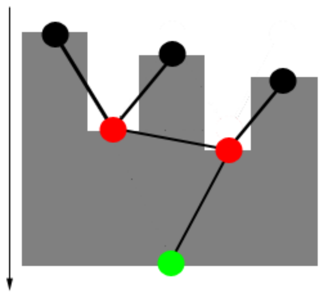
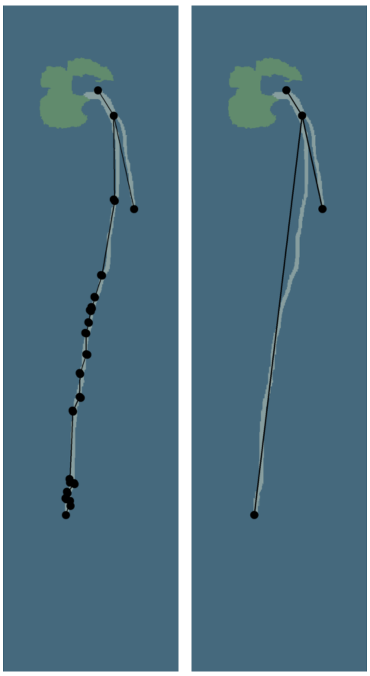


Figure 6: Critical points at different heights, the connections in this Reeb graph are unique.

Number of nodes in graph					
Type of node	birth	split	merge	death	sum
no graph pruning	111	129	84	156	480
graph pruning	38	54	13	79	184

Table 1: Total number of each type of nodes in the Reeb graphs of the root dataset with and without graph pruning.

In order to use the extracted graphs as a skeletal representation, branches that arise with artefacts need to be removed from the Reeb graph.
For each pair of nodes adjacent nodes in the graph the Euclidean distance between these two nodes is computed. If this distance is less than 1,5% of the image height such connections are discarded and nodes are relinked if needed. Regular nodes that may be introduced by this approach. As these regular nodes do not contain any needed information, they are removed after relinking. This threshold proved to be the best choice in the experiments. This graph pruning results in a reduction of the overall number of nodes in the Reeb graphs of the root dataset by 62%. Table 1 shows the numbers of nodes for all Reeb graphs in the root dataset with and without graph pruning and Figure 7 shows an example of the Reeb graph and the modified Reeb graph for root 05, day 16. All the nodes in the lower part of the root for the Reeb graph without graph pruning indicate spurious branches detected due to noise in the segmented image. These spurious branches are correctly discarded by the graph pruning approach.



(a) (b)

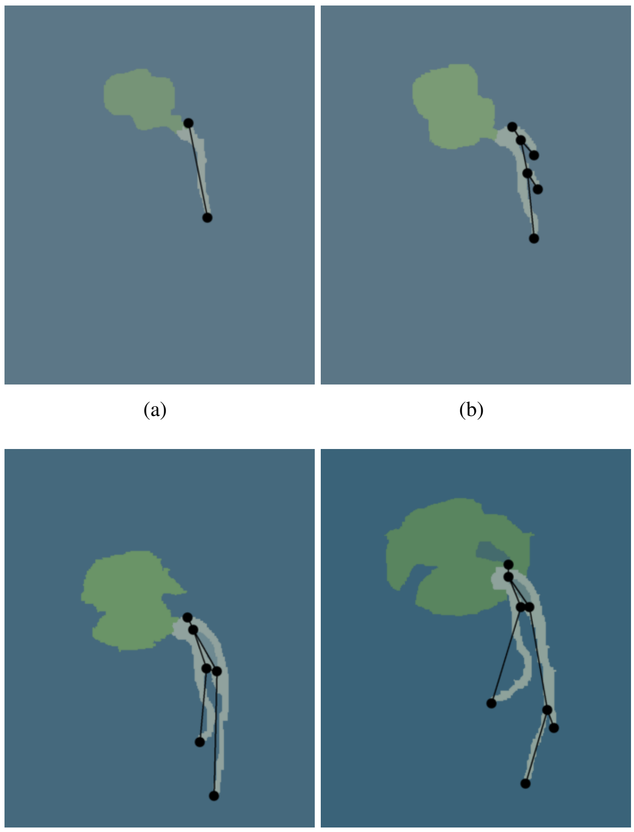
Figure 7: Reeb graph for root 05 day 16. (a) without graph pruning; (b) with graph pruning.

6 Results and evaluation on the root dataset

Figure 8 shows the resulting Reeb graph for root 07 of the dataset with both modifications implemented, drawn as an overlay. There is a cycle in the Reeb graph for day 16 and day 20 (Figure 8(c) and 8(d)). In the image of day 12 (Figure 8(b)) there are three branches: The first and the second branch overlap at sometime during the growth process between day 12 and day 16. Because of this overlap in the 3D space these two branches appear merged in the 2D projection of the image, therefore a cycle is formed in the Reeb graph.

Figure 9 shows the Reeb graphs for root 12 of the dataset (both modifications are used). Some small branches are not represented in the Reeb graphs of day 12, 16 and 20 as they resembled branches due to noise and were discarded during the graph pruning process (see Section 6.1). Again a cycle appears in the Reeb graph for day 20 as two branches overlap.

For the 34 single images of the root dataset the following criteria have been evaluated:



(a) (b) (c) (d)

Figure 8: Resulting Reeb graph for root 07 (a) day 8; (b) day 12; (c) day 16; (d) day 20.

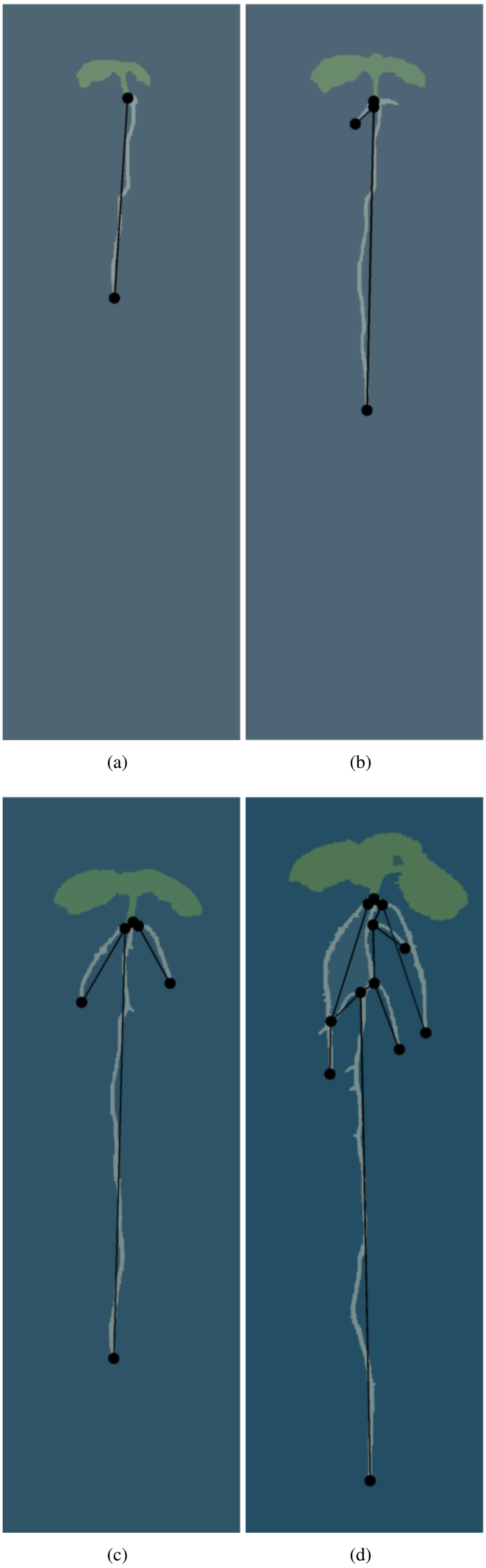


Figure 9: Resulting Reeb graph for root 12 (a) day 8; (b) day 12; (c) day 16; (d) day 20.

wrong decisions on graph pruning			
	images	false negatives	false positives
graph pruning	8	10	0
extension 1	18	4	36
extension 2	13	10	9

Table 2: Branches wrongly discarded (false negative) and wrongly accepted (false positive) in the graph pruning approach with two different corrections based on pixel-color.

6.1 Are all branches correctly detected and represented by the Reeb graph?

All major branches were correctly detected. Table 2 shows the number of images for which branches were wrongly discarded (false negatives) or wrongly accepted (false positive). For 23,5% of the images smaller branches were discarded due to the graph pruning as they resembled the frayed border artefacts caused by the segmentation. To keep these small branches that describe actual root structures, their angle could be taken into account, as true branches seem to inscribe a larger angle than branches due to noise. However, this assumption is based on the dataset presented and may not be true for other datasets. Therefore another approach was tested: for a small branch with a critical point of type split, the color values at three pixels: at the critical point (a), one row below the critical point (b) and two rows below the critical point (c) were compared:

1. the color of (a) and (c) were taken from the segmented image, while the color value of (b) was taken from the unsegmented image

2. all three color values were taken from the unsegmented image

Branches are kept if the color value of (b) is closer to (a) than to (c). Table 2 shows the results for these two tests. While the first option discards less true branches (false negatives) it keeps spurious branches for more than 50% of all images. The second option keeps less spurious branches, but does not reduce the number of false negatives compared to the graph pruning approach without these color comparisons. Taking into account not only the color values of these three pixels but of several neighbors, as it is done with Local Binary Patterns, might present an option for future work.

6.2 Are additional branches (due to e.g. noise) detected?

As shown in Table 2 all additional branches (due to segmentation artefacts) are correctly discarded by the implemented graph pruning approach.

For a series of images of one plant during the growth process the following factors have been analyzed:

Number of nodes / edges / cycles				
	day 8	day 12	day 16	day 20
root 04	2 / 1 / 0	4 / 3 / 0	6 / 5 / 0	8 / 8 / 1
root 05	2 / 1 / 0	4 / 3 / 0	4 / 3 / 0	8 / 8 / 1
root 07	2 / 1 / 0	6 / 5 / 0	6 / 6 / 1	8 / 8 / 1
root 09	2 / 1 / 0	6 / 5 / 0	6 / 5 / 0	6 / 5 / 0
root 12	2 / 1 / 0	4 / 3 / 0	6 / 5 / 0	12 / 12 / 1
root 17	-	2 / 1 / 0	6 / 5 / 0	6 / 5 / 0
root 19	2 / 1 / 0	4 / 3 / 0	8 / 8 / 1	14 / 15 / 2
root 20	2 / 1 / 0	4 / 3 / 0	4 / 3 / 0	12 / 12 / 1
root 24	-	2 / 1 / 0	4 / 3 / 0	10 / 9 / 0

Table 3: Total number of nodes, edges and cycles in the modified graph (graph pruning without corrections) of each root image in the defined dataset.

6.3 Is an automatic grouping of images of one plant from different days possible?

As the roots grow downwards in a vertical direction, there are only minor changes in the position of the starting point of the actual root (transition between leaves an roots) - not accounting for actual movement of the plant (e.g. sliding down the plate). Therefore the starting point was used for this comparison. The average minimal Euclidean distance between all starting points is 14,4 pixels. Using this distance measurement to group one image of a root with earlier or later images of the same root, the grouping is correct for 71% of all images. However, images of day 16 falsify these numbers, as the plate of day 16 appears slightly enlarged in the image compared to the images of other days. As the images were automatically cut into single plant images in a preprocessing step, this scaling is not corrected. Excluding the images of day 16, the average minimal Euclidean distance decreases to 11,6 pixels and one image is grouped correctly with earlier or later images of the same root in 88%.

6.4 General assumption: „Parts of a plant that appear in an early image of a plant do not disappear for a later image of the same plant.“

This assumption proved to be correct for the images in the root dataset. The topology of a root only changes with the creation of new components (e.g. branches) over time. Table 3 shows the number of all nodes, edges and cycles in each (modified) graph of the root images.

However, there is one exception to this assumption, which is based on the projection of a 3D structure to a 2D space. A branch in an early image of a plant might stay in the image of a later day, it may branch again but its ending may also disappear in the 2D image as it is merged with another branch due to an overlap of these two branches in the 3D space.

7 Conclusion and future work

Reeb graphs proved to be suitable descriptors for root structures as they capture the main characteristics of roots, namely branches and branch endings that are used in the phenotyping of plants, well. A Reeb graph provides a skeletal representation of a root that allows for fast analysis of root characteristics and efficient comparison of images and the contained root structure. Overlaps in 3D that appear as a merge of two branches in a 2D image are hard to distinguish from a branching point when analyzing only contours of image regions. Exploiting the topology of the root, actual branching points and overlaps in 3D can be immediately distinguished, as an overlap forms a cycle in the corresponding Reeb graph.

A future application in plant phenotyping is possible. However, for future work the segmentation approach needs to be changed to a less time-consuming (or even automatic) approach in order to allow for a larger dataset to be analyzed. Moreover different functions will be used as Morse functions. Functions that should be taken into consideration are for example a medial axis as in [7] or distance functions: for example the distance to a fixed point in a structure, the sum of geodesic distance (both are used in [10]) or the distance to an existing graph (as for example a medial axis).

Open questions for future work (on the root dataset) are: How does the chosen Morse function influence the correct detection of branches in the root structure? Is the detection of all branches, respectively the detection of additional branches due to noise, dependent on the Morse function used? Which Morse functions are able to correctly represent roots with a complex pattern of growth (e.g.: change in the main direction of growth)?

Furthermore this approach proofed to be suitable to extract plant characteristics used in the phenotyping of plants,

References

[1] Stefano Berretti, Alberto Del Bimbo, and Pietro Pala. 3D mesh decomposition using Reeb graphs. *Image and Vision Computing*, 27(10):1540–1554, Sept. 2009.

[2] S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno. Reeb graphs for shape analysis and applications. *Theoretical Computer Science*, 392(13):5–22, Feb. 2008.

[3] Harish Doraiswamy and Vijay Natarajan. Efficient algorithms for computing Reeb graphs. *Computational Geometry*, 42(67):606–616, Aug. 2009.

[4] Rachid EL Khoury, Jean-Philippe Vandeborre, and Mohamed Daoudi. 3D mesh Reeb graph computation using commute-time and diffusion distances. volume 8290, pages 82900H–82900H–10, 2012.

[5] 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.

[6] 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, Dec. 2006.

[7] Francis Lazarus and Anne Verroust. Level set diagrams of polyhedral objects. In *Proceedings of the fifth*

ACM symposium on Solid modeling and applications, SMA '99, page 130140, New York, NY, USA, 1999. ACM.

[8] Mattia Natali, Silvia Biasotti, Giuseppe Patan, and Bianca Falcidieno. Graph-based representations of point clouds. *Graph. Models*, 73(5):151164, Sept. 2011.

[9] James Stewart. *Calculus*. Cengage Learning Emea, 6th edition. international met edition, Feb. 2008.

[10] 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.

[11] 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.