Running head: Recovering root system traits using image analysis
Corresponding author: Daniel Leitner, University of Vienna, Computational Science Center, Nordbergstr. 15, A-1090 Vienna, Austria, Phone: +43 1 4277 23705,
Email: daniel.leitner@univie.ac.at.
Research area: Breakthrough Technologies
Recovering root system traits using image analysis

Exemplified by 2-dimensional neutron radiography images of lupine

Daniel Leitner¹, Bernd Felderer², Peter Vontobel³, Andrea Schnepf⁴,⁵

¹ University of Vienna, Computational Science Center, Nordbergstr. 15, A-1090 Vienna, Austria
² ETH Zurich, Institute of Terrestrial Ecosystems, CHN G76.2 Universitätsstr. 22, CH-8092 Zurich, Switzerland
³ Paul Scherrer Institute, ASQ Division, WBBA-107 CH-5232 Willigen, PSI, Switzerland
⁴ Forschungszentrum Jülich GmbH, Agrosphere (IBG-3), D-52425 Jülich, Germany
⁵ BOKU - University of Natural Resources and Life Sciences Vienna, Institute of Hydraulics and Rural Water Management, Muthgasse 18, A-1190 Vienna, Austria

One-sentence summary: Image-based parametrisation of root architectural models is advanced by a new approach for the analysis of image sequences of plant root systems, exemplified by neutron radiographic images of root systems of soil-grown lupine plants.
This research was supported by the Austrian Science Fund FWF (Grant No.: V220-N13). Daniel Leitner is recipient of an APART-fellowship of the Austrian Academy of Sciences at the Computational Science Center, University of Vienna.

Corresponding author: Daniel Leitner, Email: daniel.leitner@univie.ac.at
Abstract

Root system traits are important in view of current challenges such as sustainable crop production with reduced fertilizer input or in resource limited environments. We present a novel approach for recovering root architectural parameters based on image analysis techniques.

It is based on a graph representation of the segmented and skeletonised image of the root system, where individual roots are tracked in a fully-automated way. Using a dynamic root architecture model for deciding whether a specific path in the graph is likely to represent a root helps to distinguish root overlaps from branches and favours the analysis of root development over a sequence of images. After the root tracking step, global traits such as topological characteristics as well as root architectural parameters are computed.

Analysis of neutron radiographic root system images of *Lupinus albus* grown in mesocosms filled with sandy soil results in a set of root architectural parameters. They are used to simulate the dynamic development of the root system and to compute the corresponding root length densities in the mesocosm.

The graph representation of the root system provides global information about connectivity inside the graph. The underlying root growth model helps to decide which path inside the graph is most likely for a given root. This facilitates the systematic investigation of root architectural traits in particular with respect to parametrisation of dynamic root architecture models.
Introduction

Crucial factors for plant development are light quantity and quality as well as water and nutrient availability in soils. Regarding water and nutrient uptake, root architecture is the main aspect of plant productivity (Lynch, 2007; Smith and de Smet, 2012) and needs to be accurately considered when describing root processes. Currently, understanding the impact of roots and rhizosphere traits on plant resource efficiency is of highest relevance (Hinsinger et al., 2011). Development in this area will increase food security, by enabling a more sustainable production with reduced fertilizer input by improving cropping systems and cultivars for resource limited environments (de Dorlodot et al., 2007).

Root architectural development includes architectural, morphological, anatomical as well as physiological traits. For the systematic investigation of such complex biological systems mathematical modelling is inevitable (Roose and Schnepf, 2008). Ideally, experiments and theoretical models are developed mutually supporting each other. In this way models are created which include state of the art knowledge and have significant parameters. There are various root architectural models incorporating a multitude of processes (Dunbabin et al., 2013) which are originally based on Pagès et al. (1989) and Diggle (1988). Generally, the parametrisation of such models is difficult and demands elaborate experimental effort. In this work we present a novel approach for recovering root system parameters based on image analysis techniques. In this way we simplify the systematic investigation of root architectural traits in particular with respect to parametrisation of root system models.

Imaging techniques for the visualisation of soil-grown root systems in two and three dimensions include x-ray computed tomography
(Mooney et al., 2012; Tracy et al., 2010; Heeraman et al., 1997), neutron radiography (NR) (Oswald et al., 2008) and magnetic resonance imaging (Pohlmeier et al., 2008). NR is one of the most suitable techniques to investigate roots grown in soil, because it allows a high throughput, provides a strong contrast between roots and soil and therefore requires little effort for image processing. A major advantage of NR as well as magnetic resonance imaging is the possibility to monitor water distribution and roots simultaneously (Oswald et al., 2008; Moradi et al., 2009; Carminati et al., 2010; Menon et al., 2007; Stingaciu et al., 2013). This is especially useful as water is a crucial factor ruling root allocation in soil (Hodge, 2010).

Images of root architecture comprehend a huge amount of information and image analysis helps to recover parameters describing certain root architectural and morphological traits. The majority of imaging systems for root systems is designed for 2-dimensional images, e.g. RootReader2D (Clark et al., 2013), GiA Roots (Galkovskiy et al., 2012), SmartRoot (Lobet et al., 2011), EZ-Rhizo (Armengaud et al., 2009), Growscreen (Nagel et al., 2012). See also Le Bot et al. (2010) for a review of available software. Starting point for image analysis is commonly a grey scale image of a root system. The first step is to create a binary image by segmentation. Further steps include skeletonisation, root tracking, and data analysis. The most common segmentation method is some form of thresholding, e.g. RootReader2D, GiA Roots, SmartRoot, EZ-Rhizo, or Stingaciu et al. (2013). Other methods include the livewire algorithm (Basu and Pal, 2012) or the levelset-method (RootTrak, Mairhofer et al., 2012) that determine the boarders of each root. The creation of a root system skeleton is either done manually (e.g. DART, Le Bot et al. 2010) or based on morphological operators such as thinning and closing (GiARoots, RootReader2D, EZ-Rhizo), sometimes with options for the
user to correct skeleton points (EZ-Rhizo). The root tracking step can be performed manually (DART), or based on creating a graph representation of the root system combined with Dijkstra’s algorithm, a search algorithm that finds the shortest path between two nodes inside a graph (RootReader2D and Stingaciu et al. 2013). Furthermore, algorithms can operate on the skeleton (EZ-Rhizo) or directly on the image source (SmartRoot). In SmartRoot, the user selects a root in the original image with a mouse click and then a skeletonisation algorithm determines the skeleton of the selected root. Output of all root tracking algorithms is a data structure of a set of roots that stores information such as connectivity between roots and their position in space.

Global traits of the root system are obtained directly from the segmented image or the skeleton (GiA Roots, RooTrak). Global traits include convex hull, network depth, network length distribution, maximum number of roots, maximum width of root system, network length, or specific root length. Furthermore, the data structure from a root tracking procedure is used to obtain individual, local root parameters (DART, RootReader2D, EZ-Rhizo, SmartRoot). The latter are able to obtain root architectural parameters that can be used for model parametrisation. An additional aspect is the dealing with dynamic data, i.e., images of the same root system taken at several times. Analysis of such sequences may lead to better insight on the development of the root system e.g. DART, SmartRoot, or even reveal growth zones and their local growth velocities (Basu and Pal, 2012).

Analysis software for 2-dimensional images of soil-grown root systems currently work in a semi-automated way with respect to tracing individual roots. This requires considerable user input for larger root systems. We present a new, fully-automated approach for recovering root architectural parameters from 2-dimensional images of root systems. The software 'Root System Analyser' is the first
algorithm for 2-dimensional analysis of soil-grown root systems that features a fully-automated root tracking. Only primary roots have to be initiated manually by the user. The user is also free to initiate any laterals, but this is not mandatory. Further growth of primary roots and laterals is then tracked in a fully-automated way. In addition, there is a user interface that allows for manual correction of individual roots if required. In this work, we do not go into the details about the segmentation step, but we focus on the root tracking step and the parametrisation of a root system model (Leitner et al., 2010). The described algorithm starts with a sequence of segmented 2-dimensional images showing dynamic development of a root system. For each image morphological operators are used for skeletonisation. Based on this, a graph representation of the root system is created. A dynamic root architecture model helps to decide which edges of the graph belong to an individual root. The algorithm elongates each root at the root tip and simulates growth confined within the already existing graph representation. The increment of root elongation is calculated assuming constant growth. For each root the algorithm finds all possible paths and elongates the root into the direction of the optimal path. In this way each edge of the graph is assigned to one or more coherent roots. The algorithm considers the fact that new branches can only emerge after the apical zone has developed, which helps in the decision whether the root is branching or two roots are crossing or overlapping. Image sequences of root systems are handled in such a way that the previous image is used as starting point for the current image. This is helpful in the analysis of complex root systems as well as for retrieving dynamic parameters such as elongation rates. The algorithm is implemented in a set of Matlab m-files which makes the code flexible so that it can easily be adjusted to specific experimental set-ups or mathematical models.
We exemplify the approach with 2-dimensional neutron radiography images of *Lupinus albus* root systems grown in mesocosms filled with a sandy soil. Furthermore, we compare our approach with the approaches of SmartRoot and RootReader2D and we demonstrate how our approach can be used to analyse large root systems.

1 Results

1.1 Root tracking in neutron radiography image sequences

Figures Error! Reference source not found.a-d show the segmented images of four lupine root systems that were grown in mesocosms in a sandy soil under the same homogeneous soil moisture conditions. The root systems were imaged by neutron radiography; the different colours indicate three different measurement times. The images of each sequence were registrated using the plugin “stackreg of ImageJ 3. The segmentation algorithm was based on matched filter response (Hoover et al., 2000). The new root tracking algorithm of ’Root System Analyser’ was applied to each of these images. Figures Error! Reference source not found.e-h illustrate the sequential root tracking for the three measurements of the first image (Fig. Error! Reference source not found.a). In order to detect coherent roots the algorithm is based on two assumptions: First, roots are elongated by growing root tips. Second, new root tips emerge in the branching zone and form new lateral roots. In this way, the root tracking algorithm uses a dynamic root architectural model to decide whether a detected root is valid from a root developmental point of view.

The first step in the detection procedure is the skeletonisation of the segmented image by morphological operations. This process reduces each root to its centreline. From the skeleton, a graph is created where the nodes $N$ are the branching points, crossing points and root tips at each measurement time. On the skeleton, neighbouring nodes are connected by an edge stored in an adjacency matrix $A$. For each edge, the corresponding coordinates are stored in a list $E$. Figure Error! Reference source not found.e shows the skeleton of the above image together with the nodes of the graph. The same graph representation of the root system is also used by RootReader2D. The method of Stingaciu et al. (2013) is also based on a graph representation, however in their case the nodes represent voxels of the 3-dimensional image.

In the second step we apply the root tracking algorithm to the first measurement (indicated by the red colour in Figures Error! Reference source not found.a-d). The algorithm can initiate a tap root system automatically by selecting the node with the largest z-coordinate. Alternatively, the user can initiate one or more roots manually, then all other roots are traced in a fully-automated way. The edges of the graph are assigned to roots in the following way: For each existing root tip, further root growth is calculated in small time steps according to the underlying root architectural model. In each time step, all possible growth paths in the graph of a certain length are evaluated. The optimal growth path is chosen in dependence on straightness and average diameter. Furthermore, it is penalized if the edge is already assigned to a root. The increment of growth for each root tip occurs along the optimal path and corresponding edges are assigned to belong to the root. New root tips are created into all other possible directions. These tips start to grow after a time delay and form new lateral roots. The method enables the distinction between crossings and branches. Analysing all possible paths increases the scale on which decisions are
made, and therefore, makes it more likely to find the correct solution. Additionally, we developed a graphical user interface which enables a visual check and manual correction based on Dijkstra’s algorithm if needed. Similar to that, RootReader2D and Stingaciu et al. (2013) apply the Dijkstra’s algorithm for the root tracking procedure.

The second step is repeated for all measurements using the edge assignment of the previous measurement as initialization. Figures Error! Reference source not found.,f-h represent the assignment of the edges to roots for each measurement time. The colour red denotes the tap root, blue first order laterals, green second order laterals, and magenta higher order laterals. In this way, our algorithm can handle images with temporal information like e.g. DART or SmartRoot. This benefits on the one hand the root tracking, and on the other hand it enables the extraction of dynamic root traits like elongation rates of different root types.

The way the roots are tracked is new in the way the underlying dynamic root architecture model is used for deciding which paths in the graph potentially belong to an individual root. The algorithm starts at the tips of user-provided or previously detected roots and simulates further root growth according to the root architectural model with assumed parameters but confined within the already existing graph representation of the root system. The increment of root elongation is calculated for a small time step. Then the algorithm finds all possible paths of this length the root could take in the graph. “Possible” considers the fact that a new branch can only emerge once the length of the parent root is at least equal to the sum of the basal and apical zones. This helps in the decision whether there is a branching or a crossing. The dynamic root assignment is illustrated in the video provided in supplementary material S1.
As in Le Bot et al. (2010) the output information for each root is an identification number, the branching order, the time of emergence, the parent identification number, the distance between branching point to the parent root base, and the root length at the observation time. In addition, we store the area of the root in the image as well as the nodes of the graph that belong to each individual root. Together with the adjacency matrix $A$ and the list of coordinates $E$, this gives us all information about the position of the root in the source images. Additional examples, as well as software and documentation can be found at http://www.csc.univie.ac.at/rootbox/rsa.html or in supplementary material S2.

Certain global parameters can be determined from this result without further analysis. Table Table 1 shows mean and standard deviations of number of roots, and total length over the four root systems for the three measurement times. Further postprocessing is necessary to retrieve parameters for root architecture models.

### 1.2 Parametrising the root architectural model

For each of the four lupine root systems, we derived a data structure as described above (following Le Bot et al. 2010). Since the growth conditions were the same, we view the four images as replicates. Thus we merge the data structures and use this data to parametrise the dynamic root architectural model of Leitner et al. (2010). This model needs 12 input parameters for each root type as shown in Table Table 2. Figure Error! Reference source not found. outlines the workflow for parametrising the root architectural model.
We assume that the root system is not strictly organised in terms of root orders but in terms of different root types that do not necessarily coincide with root orders (following Pagès et al., 2004). For lupine, we assume that there is one tap root and different types of lateral roots that grow from any predecessor with a certain probability. To distinguish these groups of laterals we perform a cluster analysis. We use the Matlab-implemented algorithm k-means. The algorithm requires beforehand knowledge of the amount of clusters. From visual comparison we decided that we need two clusters of laterals, long and short. The algorithm assigns each root to one of the clusters using the observed root length. Outliers and random starting points could result in wrong or unintuitive clustering of roots. We validated the clustering by examining the clusters in the histogram of root lengths.

In a further postprocessing step, root architectural parameters are retrieved from the data structure and averaged over each root type. The estimated parameters are shown in Table Table 2.

The values for \( l_b \), \( l_a \), and \( l_n \) can directly be retrieved from the data structure. The great variation of the parameters \( l_a \), \( l_b \) and \( l_n \) is reflected in large standard deviations. This is not a measurement error but is the true variability that can be observed in the original images. The root radius \( a \) is computed from the stored values of area and length and averaged over each root type. The branching angle \( \Theta \) is measured from the corresponding coordinates of a root and its predecessor.

Root elongation is described in the model by the root growth function \( \lambda \) which is dependent on the maximal root length \( k \) and the initial elongation rate \( r \). These two parameters can not be observed directly but are obtained by fitting the root growth function \( \lambda \) to the data of root age versus root length for each type (see Fig. Error!}
Reference source not found.) Root age is only known exactly for the 0th roots. For higher root orders, root age is estimated based on the growth rate and length of apical zone of the predecessor. Note that errors in the calculation amplify in higher orders. However, the standard deviations of $r$ and $k$, which are given by the diagonal elements of the covariance matrix, are small. The maximal number of branches per root $nob$ is computed from the values of $k$, $l_n$, $l_a$, and $l_b$:

$$ nob = \frac{k - l_a - l_b}{l_n} + 1. $$

Variation from growth direction is described by two parameters, $\sigma$ and $N$. The deflection parameter $\sigma$ describes the expected angular deviation from growth direction per cm root length. It is computed along the individual roots and averaged per root type. The parameter $N$ describes the strength which keeps the root in its initial growth direction. For the tap root, this describes gravitropism, for lateral roots exotropism. This is the only parameter that we obtained by visual comparison. Different tropisms are hard to disentangle without specific experiments, e.g. hydrotropism and gravitropism (Tsutsumi et al., 2004).

1.3 Modelling case study

Models of root architecture and function increase our mechanistic understanding of soil-plant interactions. For example, Schnepf et al. (2012) investigated P uptake by a root system as effected by root growth and exudation. The parametrisation of the root architectural model is of utmost importance for a realistic description of the root surface development in soil. In this work, we use the parameter set of young lupine plants to create simulated root architectures based on the
model of Leitner et al. (2010) and to investigate the dynamic development of global root system properties.

We performed a topological characterisation of 100 simulated root systems according to Fitter (1987). The analysis reveals that, after 25 days, the mean altitude is 73.60 ± 3.28, the mean magnitude is 240.09 ± 36.95 and the mean external path length is 8376.65 ± 1667.71. Table 3 shows the dynamic development of these characteristic values. According to Fitter et al. (1991), the topological index is a number between zero and one and correlates positively with exploitation efficiency of the root system. Thus, these numbers provide an estimate of the exploitation efficiency and enable quick comparisons, e.g. between cultivars.

Simulations were performed in 3 dimensions and constrained within the mesocosm as in the experimental setup. This is illustrated in Fig. Error! Reference source not found.a. Figure Error! Reference source not found.b demonstrates further analysis in terms of root length densities. Such information can be further used in plant-soil interaction models by calculating sink terms for plant water or nutrient uptake, for example. The Matlab implementation of the root architecture model facilitates both post processing and coupling to other models. The root growth inside the mesocosm is visualized in supplementary material S3.

The dynamic development of total root length and number of roots is illustrated in Fig. Error! Reference source not found.. The solid and dashed lines represent the total root length and number of roots of a simulated lupine root system based on the parameter set of Table 2. The asterisks with error bars show the corresponding measured root lengths and number of roots according to Table Table 1. We can
observe that initially the average root length is approximately 1 cm; this increases over time to an average length of approximately 2 cm.

1.4 Comparison with SmartRoot and RootReader2D

Several image analysis tools for 2-dimensional root systems are available. Our algorithm combines features that are partly present in other softwares. It shares the skeletonisation and graph representation of the root system with RootReader2D, while it is similar to SmartRoot with respect to the use of image sequences and also with respect to postprocessing and parameter-retrieval. We chose the first lupine image shown in Fig. Error! Reference source not found.a and analysed it with both software for comparison.

In Table Table 4, we compare the results of Root System Analyser with the results obtained by analysing the same image with SmartRoot. The resulting tracings according to both tools are shown in Fig. Error! Reference source not found.. The main parameters such as overall root length, internodal distance on 0\(^{th}\)-order root and branching angles agree well. Thus, we suggest that both produce satisfactory results. However, there are two main differences in the results: Firstly, our automated algorithm detects more very small roots than SmartRoot, i.e. roots are detected at an earlier stage. This means that the overall root length is still similar, however the number of (in this case 2\(^{nd}\)-order) roots is larger and thus the internodal distance (in this case on the 1\(^{st}\)-order roots) is smaller. This has implications for parametrisation of root architectural models.
Secondly, the image was analysed independently with both imaging tools, and this resulted in the fact that the 0th-order root took a different path in the two cases. The underlying algorithms are very different between the two tools. Our automated procedure is based on skeletonisation and graph representation of the root system coupled with a dynamic root architectural model for automatic root tracking. It does feature manual correction should the user find it necessary. SmartRoot works on the (grey scale) image itself; the user can follow individual roots by mouse-clicks. SmartRoot does feature an automated algorithm for individual roots that is triggered by a mouse-click inside the root, and this algorithm is based on finding the mid-line of the root and progressing forward and backward until the base and tip of the root are reached. The user-input required by our automated tool is considerably less and thus it is more suitable for the complete analysis of large root systems. Although the underlying algorithms for root detection are different, the handling in the graphical user interface is similar, in particular with respect to handling sequences of images of the same root system. Both use the tracings from the previous time step as starting points for the current root tracking. In SmartRoot, the user is required to determine anchor points in each image of a sequence in order to be able to compensate for little shifts between images. The postprocessing in SmartRoot is user friendly, with links to SQL databases or output produced as txt-files and images. When using Root System Analyser, one needs to be familiar with Matlab. However, this offers a lot of postprocessing-options within Matlab, and several Matlab functions for parametrisation of root architectural models are already provided for.

Like Root System Analyser, RootReader2D is based on an algorithm that first creates the centrelines of each root and then a graph representation where the root branching points, crossing points and root
tips are the nodes of the graph and the edges are the links between the nodes. Root System Analyser and RootReader2D produce the same skeleton and graph with only tiny variations of the position of the centreline due to differences in the skeletonisation procedure. Root tracking in RootReader2D is based on Dijkstra’s algorithm. However, the handling for the user is similar to that of SmartRoot regarding that each root is tracked individually by one or more mouse-clicks. Neither RootReader2D nor SmartRoot offers a fully automated procedure for tracking all the roots of the root system like Root System Analyser. Thus, this tool is more suitable for the extensive analysis of larger root systems. Results of analysing a large root system are shown in the next section.

1.5 Application to a large maize root system

“Root System Analyser is so far the only algorithm based on imaging techniques that was applied to root systems as complex as that of a 78 days old maize root system. There is the potential to use this algorithm to extract more information from the large number of existing 2D images of excavated root systems such as those from the “Wurzelatlas” series.

To test the algorithm on a large fibrous root system, we used the software to track roots in the 2-dimensional drawing of a large maize root system (Kutschera et al., 2009, p. 164). Assuming the sowing time to be May 1st, the root system is 78 days old. It goes down to a depth of approximately 60 cm and has a maximum width of approximately 120 cm, thus being much larger than the laboratory root systems typically used for imaging. The original hand drawing (Lichtenegger, 2003) was scanned at high resolution (1200 dpi) yielding an image size of 116 megapixel. The image was turned into a black-and-white image by thresholding and then analysed by Root System Analyser.
In the tracking procedure, we encountered two problems: Firstly, it is hard to distinguish individual roots in the highly dense part of the root system just below the stem. Thus, the node points assigned in this area are questionable. Secondly, the algorithm does not yet detect multiple primary roots automatically. Therefore, we assigned several primary roots by hand and only then started the automated procedure. In this way, we picked 21 initial roots and the algorithm detected 3457 roots automatically.

The tracking results agreed well with data from literature. Roots of cereal plants, such as maize, typically have three or four orders (Roose et al., 2001), and this has been confirmed by our analysis. The resulting root system has a maximum root order of 4, however most of the roots belong to orders 0-3. The original image and the tracked root system are shown in Figure Error! Reference source not found.. Number and length of roots in each order that were detected in this 2-dimensional view of the root system are provided in Table Table 5. Selected parameters of orders 0-3 are presented in Table 6, dynamic parameters r and k could not be estimated from only one image. Average root architectural parameters for the first three orders compare well to those presented in Pagès et al. (1989) with the exception of the the length of the apical zone of 0-order roots.

2 Conclusions and outlook

We presented a novel approach for root tracking from 2-dimensional images of root systems. The combination of the graph representation of the root system together with an underlying root architectural model
results in a reliable method for root assignment. In this way, the decisions are made based on global information about the connectivity within the graph and it is assured that root assignment is valid from a root system developmental point of view. Particularly, this is beneficial for dealing with overlaps and intersections in 2-dimensional images. Comparison with SmartRoot shows similar capabilities for postprocessing, parameter retrieval and handling of image sequences, but differences in the underlying algorithm for root tracing. Our algorithm uses the global information contained in the graph representation of the root system and traces all roots of the root system in a fully-automated way. Comparison with RootReader2D shows similarities with respect to the automated skeletonisation and graph creation procedure. However, also in RootReader2D individual roots have to be traced manually by at least one mouse click. Thus, our new algorithm is more suitable for larger root systems as demonstrated by tracing the roots of a large maize root system. This is promising for retrieving more information from the many 2-dimensional hand-drawings of excavated root systems, e.g. in the “Wurzelatlas book series.

If an image sequence is available that shows root system development, the root assignment becomes more robust since the detected roots of the previous image act as initial roots for the current image. Furthermore, the dynamical development of the root system can be analysed, like elongation rates for individual soil grown roots.

Our approach works on the segmented images of root systems. In contrast, Lobet et al. (2011) and Naeem et al. (2011) work directly on the original image data. This data contains more information and therefore can better seize small scale image features and thus work with high precision on small root systems. However, they lack the ability to use global information about connectivity, which is the strength of our
graph based technique. A combination of these approaches could further increase the quality of root tracking algorithms.

While the literature for automatic root assignment is limited, there is vast literature for biomedical applications, like vessel tracking (e.g. Lesage et al., 2009). Ideas from this area could further benefit image analysis for root architecture.

Plant root experiments are extremely costly regarding labour and time. Therefore, we suggest that the proposed image analysis method can help to extract the most information from root system images. In particular, we propose this approach for image-based parametrisation of root architectural models.

3 Materials and methods

Starting point for the automated root tracking algorithm is a segmented 2-dimensional image. In order to annotate coherent roots from this image, we make use of morphological and graph theoretic methods to perform a full-automated or semi-automated root tracking. The methods are implemented in Matlab. The corresponding m-files are freely available at http://www.csc.univie.ac.at/rootbox/rsa.html and as well in supplementary material S2.

3.1 Binary 2-dimensional neutron radiography images of root systems of Lupinus albus

Neutron radiography is one of the few non-invasive and non-destructive techniques available for studying plant root systems in situ (Moradi et al., 2009). Although this technique is too time consuming for high-throughput screening of plants, it is capable of imaging plant root systems in opaque soil. Providing a soil
environment to the plant roots is necessary for studying plastic traits of root systems such as root length. Furthermore, this technique can be used to recover complete root systems.

We demonstrated the applicability of Root System Analyser to detect root systems in neutron radiography images. Segmented neutron radiography images were provided by B. Felderer who performed an experiment on the effects of P and water inhomogeneity on root system growth of *Lupinus albus* (Felderer and Vontobel, unpublished). Briefly, the experimental setup, filtering and segmentation are outlined below.

Single white lupin (*Lupinus albus*) plantlets were grown in teflon-coated Al-containers of 27x27x1.4 cm size over a period of 26 days. The Al-containers were filled with sandy soil extracted from the forefield of an open cast mine near Cottbus (Welzow Süd, Germany). The soil contained almost no organic matter. Plants were grown in a climate chamber at a humidity of 60 % with a 16 h : 8 h day : night cycle with 21/16°C temperature, respectively. The root system was visualised with neutron radiography at 12, 19 and 26 days after germination (at the Paul Scherrer Institute in Villigen, Switzerland), which is highly suitable to visualize roots and water in soil (see Menon et al., 2007; Moradi et al., 2009). The field of view was 18 x 18 cm with a nominal resolution of 0.176 mm, thus four scans per plant container were needed to record the whole sample. Image sequences were registered using the ImageJ plugin “stackreg.

The neutron radiography images were first corrected for their flat-field, for sample scattering and for variation of beam intensity between the scans using the Quantitative Neutron Imaging Software (Hassanein et al., 2005). After stitching, a matched filter response was applied for root segmentation using a software developed by Anders Kästner (Paul Scherrer Institut, Villigen), which amplifies thin structures and creates one binary image per measurement.
For this study, the images of four containers with initially homogeneous water content were selected (see Figs. Error! Reference source not found.a-d) in order to demonstrate the new root tracking approach.

3.2 Automatic root tracking

3.2.1 From the binary image to a graph

In a first step we derive an approximated centreline from each binary image $im_B$. Therefore, we create a skeleton $im_S$ from $im_B$ by iteratively applying two morphological operations: Thinning, which removes boundary pixels without disconnecting any domains (Lam et al., 1992), and closing, (i.e. dilation followed by erosion).

In the next step $im_S$ is used to create a graph representation of the root system. For each pixel on the skeleton in $im_S$ we count the number of neighbouring skeleton pixels in an 8-neighbourhood. This yields a matrix $im_C$. If a pixel of $im_C$ has exactly one neighbour it represents a leaf in the corresponding graph. If it has more than two neighbours, it represents a branching or crossing point. Therefore all pixels that correspond to nodes in the graph are given by

$$im_N(x) :=$$

The connected components in $im_N$ represent the nodes in the graph. They are labelled with the Matlab function `bwlabel` and the coordinates of the nodes $N$ are exactly the centres of each connected components. Two nodes are connected by an edge if they are connected by the skeleton exclusive of the nodes, i.e. $im_S \setminus im_N$. For each edge the

...
corresponding pixel coordinates from the skeleton $im_S$ are stored in a list $E$. Node connectivity is represented by an adjacency matrix $A$ of the undirected graph, where the entries represent the edge indices of $E$. In a final step, results are improved by applying a 1-dimensional Gauss filter to the edge coordinates, and optionally, by removing small terminal edges from the graph. The above approach is implemented in the file $\text{image2graph.m}$. For the root tracking algorithm we use the following edge weights: First, the length of each edge is calculated from the coordinates $E$ using the Euclidean distance. Second, the approximate radius of the root at each coordinate is obtained by calculating the distance function of the binary image $u_b$. In the resulting distance matrix $D$ each pixel value holds the distance to the closest boundary of the root. Therefore, the values of $D$ at the centreline are exactly the root radii. The distance matrix $D$ is calculated by the Matlab function $\text{bwdist}$ (described in Maurer et al., 2003).

### 3.2.2 Root tracking in the graph

In the root tracking algorithm we assign each edge of the graph to one or more individual roots and retrieve information about the connectivity of these individual roots. We use basic knowledge about root system development to detect coherent roots: Roots emerge from growing root tips. Each root consists of a basal zone succeeded by a branching zone followed by an apical zone. Lateral roots emerge only in the branching zone, after the apical zone has developed. The following algorithm is implemented in the Matlab file $\text{trackRoots.m}$.

We store a list $T$ of growing root tips. Initially, these are set manually, or if there is only one tap root, the algorithm picks the node with largest $z$-coordinate.
The time that has passed at each measurement time is subdivided into smaller time steps typically used for root growth modelling. While there are any tips in the list $T$, the growth increment for each root during each time step is calculated according to the underlying root growth model. For each root we add edges until the growth increment is reached. This is done in the following way:

1. Find all possible growth paths in the graph from the node which represents the growing root tip ($\text{getPath.m}$). This is performed in a recursive way. All paths from the node with a length smaller than a maximal path length $R_s$ are retrieved. Each path is a list of edges with corresponding nodes, and edge weights such as root length and radius.

2. Choose the optimal path by evaluating the quality $q_j$ of each path with index $j$ as described in the following section. The optimal path has index $j=\arg\max_{i \in \mathcal{I}} q_i$ with $q_j = \max_{i \in \mathcal{I}} q_i$. The heuristic is based on the path length, radius, straightness, and information on previous edge assignments.

3. 
   - If the optimal path $j$ fulfils the quality criterion $q_j > 1$, the first edge of the optimal path is added to the growing root, and the root tip moves to the next node. The assigned edge is denoted as visited in order to penalize multiple visits.
   - Otherwise, the root stops growing and is removed from the list $T$.

4. New laterals emerge into all other possible directions. These new laterals are added to the list $T$ and start to grow after a time delay, in order to let the apical zone of the base root develop.
The procedure (1)-(4) is repeated for all time steps until no growing root tips are left.

The algorithm dynamically assigns each edge in the graph to one or more roots. Following Le Bot et al. (2010) the output information for each root is an identification number \textit{number}, the branching order \textit{order}, the time of emergence \textit{ct}, the parent identification number \textit{predecessor}, the distance between branching point to the parent root base \textit{prelength}, and the root length at the observation time \textit{length}. In addition, we store the area of the root in the image \textit{area}, as well as the nodes that belong to each individual root \textit{path}.

### 3.2.3 Path evaluation

A path in the graph is evaluated by the following heuristic quality criterion (implemented in the Matlab file \texttt{quality.m}). The quality criterion is based on the average root radius and the straightness of the root.

The algorithm starts at a growing root, which is typically thicker than its lateral branches. Therefore, we want to follow the path with the largest average root radius \( r \) to find a coherent root. The computation of the average path radius \( r \) is described in section 3.2.1. Additionally, we take into account if the edge is already assigned to another root. In this case we subtract its average root radius from the average radius of the edge.

We assume that roots grow preferably straight. Thus, we want to favour the straightest path in the graph. The straightness \( s \) between two coordinates \( x_1 \) and \( x_2 \) is given by the ratio of the length of the straight line between the points and the length along the edges. The coordinate \( x_1 \) is located at a distance \( R_a \) along the root in front of the current root tip location, \( x_2 \) is located at \( R_a \) along the new path after the current root tip. Therefore, the straightness is given by
\[ s = |x_2 - x_1|/\sqrt{2R}. \]

We define the quality \( q \) of each path as the product of the radius \( r \) and straightness \( s \):

\[ q = rs^e. \]

The exponent \( e \) describes the strength of influence of the straightness (according to experience \( e = 2 \) performs well).

3.2.4 Image sequences

Image sequences enable the calculation of elongation rates of individual roots. Furthermore, our algorithm uses this dynamic information to improve the root tracking. Young root systems are still small and easier to track automatically. After the tracking of the initial measurement, the root assignment of the previous measurement always acts as initial root system for the current image. If the time difference between the measurements is small, the root development is easily manageable automatically by the tracking algorithm. This procedure facilitates correct root tracking even in large and complex root systems.

Manual correction is hardly necessary. However, we provide a graphical user interface (RSA_GUI.m) to enable manual root corrections if required. This includes removal of wrongly assigned roots as well as manual assignment by picking the starting and end points of the root. Figure Error! Reference source not found. shows a screen shot of the graphical user interface after automatic detection of the roots of the lupine root system at the first measurement time. If a path is found to be incorrect, the user can remove the root by selecting “remove and clicking at any point on this root. A new root can be added by selecting “add root and clicking on start and end points of the root. The path of minimal length between those nodes is found by Dijkstra’s algorithm. Dijkstra’s algorithm is a graph search algorithm.
that solves the single source shortest path problem for a graph with positive edge path costs. In our application this costs are defined as edge lengths or edge radii. Thus, for any wrongly determined root, one mouse-click for its removal and two mouse-click for creation of a new path are required.

### 3.2.5 Processing efficiency

On a standard PC, the analysis of young root systems takes only takes a few minutes. The most time consuming step is after the automated analysis if the user wishes to manually correct some of the assigned roots. The automatic detection of roots in a large root system such as a large maize root system (11658 nodes, 14783 edges, 3478 detected roots) may take a few hours on a standard PC. In mature root systems there are limitations with respect to the complexity of root system. The part just below the stem is so densely rooted that individual roots can no longer be recognised. As a result this area appears as a white or black area in the segmented image, and detection fails. This can be overcome if sequences of the same root system at earlier times are available.

Note that Root System Analyser can work with all two dimensional imaging techniques that offer a sufficient resolution such that a segmented binary image can be created, where roots are represented by white pixels and the background by black pixels. Furthermore, all pixels that belong to the roots must be connected. Currently, there is no automatic segmentation provided in the software. Generally, this kind of segmentation is a non-trivial task for most imaging techniques due to different types of artefacts. Depending on the imaging technique that is used there might be specialised software available for the segmentation step.
3.3 Model parametrisation

We use the resulting data structure to parametrise the dynamic root architectural model of Leitner et al. (2010); however, any other root architectural model could be parametrised as well. This model needs 12 input parameters as shown in Table 2. They are computed as averages over root order or root type (Pagès et al., 2004). In order to find root types, we perform a cluster analysis regarding root length using the Matlab function \texttt{kmeans}. Alternatively, root types could be distinguished according to the developmental stage of a plant, e.g. a maize plant. This is not implemented in the current version of "Root System Analyser", but any user is free to replace the \texttt{kmeans} function by any other function that groups the roots into different types. The computation of the root architectural parameters for each root type is implemented in the Matlab file \texttt{analyseRS.m}.

The values for $l_b$, $l_a$, and $l_n$ can be retrieved from the data structure as described in end of section 3.2.2. For each root $j$ with laterals $I$, the length of the basal zone is given by $l_{b,j} = \min_{i \in I} \{\text{prelength}_i\}$. The apical zone is given by $l_{a,j} = \text{length}_j - \max_{i \in I} \{\text{prelength}_i\}$. Apical and basal zones can only be determined if the root has at least one branch. The internodal distances are given by $l_{n,j,i} = \text{prelength}_{i+1} - \text{prelength}_i$, where the distances between branching point to the parent root base are in ascending order. Internodal distances can only be determined if the root has at least two branches.

The root radius $a$ is computed from the stored values of area and length and averaged over each root type. The branching angle $\Theta$ is calculated as the angle between two straight lines approximating the root and its lateral in the branching point.
Root elongation is described in the model by the growth function $\lambda$ which is dependent on the maximal root length $k$ and the initial elongation rate $r$, i.e., $\lambda=k\left(1-e^{-\frac{rt}{k}}\right)$. These two parameters are obtained by fitting the root growth function $\lambda$ to the data of root age versus root length for each type. The standard deviations of $r$ and $k$ are the diagonal elements of the covariance matrix.

Variation from growth direction is described by two parameters, $\sigma$ and $N$. The deflection parameter $\sigma$ describes the expected angular deviation from growth direction per cm root length. This value is obtained from by calculating the angular change along the roots. All parameters except for $N$ are computed along the individual roots and then averaged per root type. The parameter $N$ describes the strength with which the root will keep its initial growth direction (i.e. gravitropism for tap roots, exotropism for later roots). This is the only parameter that we obtained by visual comparison.
References


List of Figures

1  Top row (a-d): Four segmented images of Lupinus albus from NR. The colours indicate measurement time (red=12 days, yellow=19 days, blue=26 days). Bottom row: (e) Skeleton representation of the root system shown in 1a together with the nodes of the corresponding graph. (f-h) Assignment of edges to the individual roots at the three measurement times.

2  Workflow for the parametrisation.

3  Fitting the parameters $k$ and $r$ of the root growth function $\_\_\_$ to the data of rootage versus length.

4  (a): 3-dimensional visualisation of the simulated root architecture grown in a mesocosm. (b): Corresponding root length densities.

5  Measured (markers) and simulated (lines) development of root length and number of roots with time.

6  The root system as tracked by the different root image analysis tools.

7  Tracking a 2-dimensional drawing of a large Maize root system.

8  Graphical user interface of “Root System Analyser”.
Table 1: Average number of roots and root lengths of four lupine plants at three sampling times

<table>
<thead>
<tr>
<th>Sampling time</th>
<th>Number of roots</th>
<th>Overall root length</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 days</td>
<td>78.5 ± 16.4</td>
<td>135.4 ± 13.3 cm</td>
</tr>
<tr>
<td>19 days</td>
<td>177.3 ± 28.9</td>
<td>341.4 ± 27.2 cm</td>
</tr>
<tr>
<td>26 days</td>
<td>266.0 ± 29.7</td>
<td>449.2 ± 43.3 cm</td>
</tr>
</tbody>
</table>
Table 2: Root system parameters of *Lupinus albus* as recovered from four 2-dimensional neutron radiography images

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Tap root [mean, std]</th>
<th>Long laterals [mean, std]</th>
<th>Short laterals [mean, std]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basal zone $l_b$ (cm)</td>
<td>[0.38, 0.33]</td>
<td>[2.20, 1.78]</td>
<td>[0.34, 0.46]</td>
</tr>
<tr>
<td>Apical zone $l_a$ (cm)</td>
<td>[3.07, 2.94]</td>
<td>[4.38, 2.49]</td>
<td>[0.64, 0.83]</td>
</tr>
<tr>
<td>Internodal distance $l_n$ (cm)</td>
<td>[0.33, 0.31]</td>
<td>[0.81, 0.96]</td>
<td>[0.78, 0.79]</td>
</tr>
<tr>
<td>Root radius $a$ (cm)</td>
<td>[0.09, 0.01]</td>
<td>[0.07, 0.01]</td>
<td>[0.07, 0.01]</td>
</tr>
<tr>
<td>Branching angle $\Theta$ (-)</td>
<td>-</td>
<td>[1.45, 0.47]</td>
<td>[1.47, 0.67]</td>
</tr>
<tr>
<td>Initial growth rate $r$ (cm day$^{-1}$)</td>
<td>[4.4, 0.84]</td>
<td>[0.88, 0.07]</td>
<td>[0.12, 0.01]</td>
</tr>
<tr>
<td>Maximal root length $k$ (cm)</td>
<td>[28.55, 1.39]</td>
<td>[17.41, 2.56]</td>
<td>[1.55, 0.19]</td>
</tr>
<tr>
<td>Probability Type 1 successor (-)</td>
<td>0.31</td>
<td>0.009</td>
<td>0</td>
</tr>
<tr>
<td>Probability Type 2 successor (-)</td>
<td>0.69</td>
<td>0.991</td>
<td>1</td>
</tr>
<tr>
<td>Number of branches $nob$ (-)</td>
<td>77</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>Tropism $N$ (-)</td>
<td>2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Tropism $\sigma$ (cm$^{-1}$)</td>
<td>0.39</td>
<td>0.47</td>
<td>0.61</td>
</tr>
</tbody>
</table>
Table 3: Topological parameters over time

<table>
<thead>
<tr>
<th>Time (days)</th>
<th>Magnitude [mean, std]</th>
<th>Altitude [mean, std]</th>
<th>External path length [mean, std]</th>
<th>Topological index [mean, std]</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>[33.01, 10.22]</td>
<td>[30.35, 8.66]</td>
<td>[578.66, 305.43]</td>
<td>[0.98, 0.02]</td>
</tr>
<tr>
<td>10</td>
<td>[68.20, 13.29]</td>
<td>[52.68, 7.69]</td>
<td>[1814.09, 529.92]</td>
<td>[0.94, 0.02]</td>
</tr>
<tr>
<td>15</td>
<td>[114.11, 19.21]</td>
<td>[64.43, 5.95]</td>
<td>[3484.78, 818.65]</td>
<td>[0.88, 0.02]</td>
</tr>
<tr>
<td>20</td>
<td>[171.98, 26.34]</td>
<td>[70.31, 4.35]</td>
<td>[5650.54, 1166.15]</td>
<td>[0.83, 0.02]</td>
</tr>
<tr>
<td>25</td>
<td>[240.09, 36.95]</td>
<td>[73.60, 3.28]</td>
<td>[8376.65, 1667.71]</td>
<td>[0.79, 0.02]</td>
</tr>
</tbody>
</table>
Table 4: Comparison of outputs of Root System Analyser with those of SmartRoot

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Root System Analyser</th>
<th>SmartRoot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall number of roots</td>
<td>273</td>
<td>212</td>
</tr>
<tr>
<td>Number of 1&lt;sup&gt;st&lt;/sup&gt; order roots</td>
<td>58</td>
<td>57</td>
</tr>
<tr>
<td>Number of 2&lt;sup&gt;nd&lt;/sup&gt; order roots</td>
<td>204</td>
<td>144</td>
</tr>
<tr>
<td>Number of 3&lt;sup&gt;rd&lt;/sup&gt; order roots</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Overall length of roots (cm)</td>
<td>391.60</td>
<td>357.73</td>
</tr>
<tr>
<td>Length of 0&lt;sup&gt;th&lt;/sup&gt; order root (cm)</td>
<td>27.30</td>
<td>18.44</td>
</tr>
<tr>
<td>Length of 1&lt;sup&gt;st&lt;/sup&gt; order roots (cm)</td>
<td>279.25</td>
<td>269.47</td>
</tr>
<tr>
<td>Length of 2&lt;sup&gt;nd&lt;/sup&gt; order roots (cm)</td>
<td>83.85</td>
<td>68.02</td>
</tr>
<tr>
<td>Length of 3&lt;sup&gt;rd&lt;/sup&gt; order roots (cm)</td>
<td>1.19</td>
<td>1.81</td>
</tr>
<tr>
<td>Average root radius (cm)</td>
<td>0.07</td>
<td>0.06</td>
</tr>
<tr>
<td>Average radius of 0&lt;sup&gt;th&lt;/sup&gt; order root (cm)</td>
<td>0.09</td>
<td>0.09</td>
</tr>
<tr>
<td>Average radius of 1&lt;sup&gt;st&lt;/sup&gt; order roots (cm)</td>
<td>0.07</td>
<td>0.06</td>
</tr>
<tr>
<td>Average radius of 2&lt;sup&gt;nd&lt;/sup&gt; order roots (cm)</td>
<td>0.07</td>
<td>0.06</td>
</tr>
<tr>
<td>Average radius of 3&lt;sup&gt;rd&lt;/sup&gt; order roots (cm)</td>
<td>0.07</td>
<td>0.06</td>
</tr>
<tr>
<td>Internodal distance on 0&lt;sup&gt;th&lt;/sup&gt; order root (cm)</td>
<td>0.34</td>
<td>0.32</td>
</tr>
<tr>
<td>Branching angle of 1&lt;sup&gt;st&lt;/sup&gt; order roots (°)</td>
<td>79.6</td>
<td>82.08</td>
</tr>
<tr>
<td>Internodal distance on 1&lt;sup&gt;st&lt;/sup&gt; order roots (cm)</td>
<td>0.7</td>
<td>0.98</td>
</tr>
<tr>
<td>Branching angle of 2&lt;sup&gt;nd&lt;/sup&gt; order roots (°)</td>
<td>81.4</td>
<td>87.43</td>
</tr>
</tbody>
</table>
Table 5: Number and lengths of roots in the different orders of a maize root system (Lichtenegger, 2003)

<table>
<thead>
<tr>
<th>Order</th>
<th>Number of roots</th>
<th>Root Length (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>21</td>
<td>12.55</td>
</tr>
<tr>
<td>1</td>
<td>1380</td>
<td>37.08</td>
</tr>
<tr>
<td>2</td>
<td>1781</td>
<td>17.53</td>
</tr>
<tr>
<td>3</td>
<td>278</td>
<td>2.16</td>
</tr>
<tr>
<td>4</td>
<td>18</td>
<td>0.10</td>
</tr>
<tr>
<td>sum</td>
<td>3478</td>
<td>69.43</td>
</tr>
</tbody>
</table>
Table 6: Selected root system parameters of *Zea mays* as recovered from the image of Lichtenegger (2003)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>0\textsuperscript{th} order [mean, std]</th>
<th>1\textsuperscript{st} order [mean, std]</th>
<th>2\textsuperscript{nd} order [mean, std]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basal zone $l_b$ (cm)</td>
<td>[0.15 0.35]</td>
<td>[0.76 1.03]</td>
<td>[0.35 0.49]</td>
</tr>
<tr>
<td>Apical zone $l_a$ (cm)</td>
<td>[3.00 1.74]</td>
<td>[1.05 1.09]</td>
<td>[0.60 0.55]</td>
</tr>
<tr>
<td>Internodal distance $l_n$ (cm)</td>
<td>[0.88 2.12]</td>
<td>[0.89 1.13]</td>
<td>[0.61 0.55]</td>
</tr>
<tr>
<td>Branching angle $\Theta$ (-)</td>
<td>-</td>
<td>[1.51 0.91]</td>
<td>[1.44 0.86]</td>
</tr>
<tr>
<td>Number of branches $nob$ (-)</td>
<td>[65.71 38.83]</td>
<td>[1.29 3.47]</td>
<td>[0.16 0.76]</td>
</tr>
</tbody>
</table>
Figure 1: Top row (a-d): Four segmented images of *Lupinus albus* from NR. The colours indicate measurement time (red=12 days, yellow=19 days, blue=26 days). Bottom row: (e) Skeleton representation of the root system shown in 1a together with the nodes of the corresponding graph. (f-h) Assignment of edges to the individual roots at the three measurement times.
Figure 2: Workflow for the parametrisation
Figure 3: Fitting the parameters $k$ and $r$ of the root growth function $\lambda$ to the data of root age versus length.
Figure 4: (a): 3-dimensional visualisation of the simulated root architecture grown in a mesocosm. (b): Corresponding root length densities.
Figure 5: Measured (markers) and simulated (lines) development of root length and number of roots with time.
Figure 6: The root system as tracked by the different root image analysis tools

(a) Root System Analyser

(b) SmartRoot
Figure 7: Tracking a 2-dimensional drawing of a large Maize root system
Figure 8: Graphical user interface of “Root System Analyser”.