Robust Skeletonization for Plant Root Structure Reconstruction from MRI

Jannis Horn[1], Yi Zhao[1], Nils Wandel[1], Magdalena Landl[2], Andrea Schnepf[2], and Sven Behnke[1]

[1]Autonomous Intelligence Systems Group, University of Bonn
[2]Forschungszentrum Jülich IBG-3

***Abstract*—Structural reconstruction of plant roots from MRI is challenging, because of low resolution and low signal-to-noise ratio of the 3D measurements which may lead to disconnectivities and wrongly connected roots. We propose a two-stage approach for this task. The first stage is based on semantic root vs. soil segmentation and finds lowest-cost paths from any root voxel to the shoot. The second stage takes the largest fully connected component generated in the first stage and uses 3D skeletonization to extract a graph structure. We evaluate our method on 22 MRI scans and compare to human expert reconstructions.**

I. Introduction

Plant root system architecture is important for plant performance, particularly under challenging environmental conditions such as droughts. 3D volumetric imaging methods such as Magnet Resonance Imaging (MRI) [?], Computer Tomography (CT) [?] or Neutron Radiography (NR) [?] enable in-situ observations of root system development in opaque soil.

In this work, MRI images are used which can suffer from low resolution, compared to the diameter of thin roots, and low signal-to-noise ratio (SNR), caused e.g. by ferromagnetic particles in the soil. Fig. 2a shows a raw MRI image after thresholding. In Fig. 2b, the image has been segmented using a 3D U-Net [?] reducing the noise considerably, but some patches of noise are still present and some roots have gaps.

For plant root system analysis, the structure of a root has to be extracted, i.e. the 3D image has to be transformed into a tree graph structure. Currently, this is done by human experts in a 3D work bench [?], which is a time consuming process that limits plant root analysis to few samples. Automating root structure extraction allows for fast and reproducible processing of larger MRI measurement sets.

For finding the medial axis of roots, skeletonization algorithms [?] can be used, but gaps must be closed before they can be applied. For closing gaps in structures, often morphological operations are used. These are sufficient to fill holes and close smaller gaps, but when roots touch each other, closing can lead to wrong connections of root parts.

In this work, we present a novel robust curve skeletonization algorithm for 3D images. We take a two-stage approach: Stage 1 extracts a largest connected component (LCC), see Fig. 2c, starting from a given starting point, i.e. the shoot. To this end, a modified Dijkstra shortest path algorithm [?] is employed. The shortest path algorithm is modified to connect disconnected areas by finding a single smallest gap. In Stage 2, a curve skeletonization algorithm is applied to the LCC. The resulting algorithm works on imperfect data containing clusters of noise, disconnectivities and intricate object structures. Fig. 2d shows the resulting root structure graph.

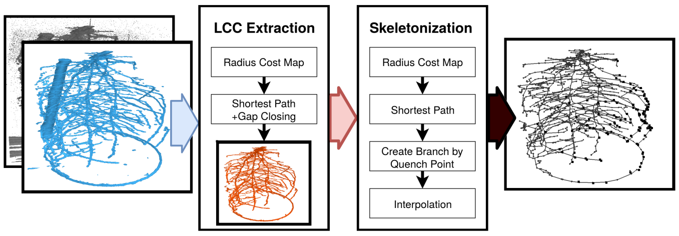
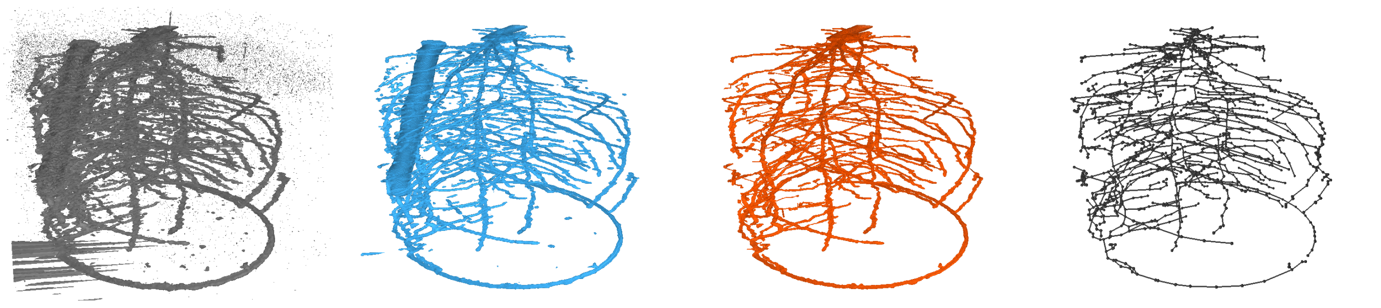


Fig. 1: Given a (segmented) plant root MRI scan, first a largest connected component is extracted, followed by skeletonization of the LCC to create a root graph.

The presented curve skeletonization algorithm has low computational demands. It is capable of reconstructing the root structure from large 3D images (396×512×512) on modest hardware in short time. We compare the reconstructions to human expert reconstructions.

II. Related Work

An early root structure extraction algorithm has been developed by Schulz et al. [?]. They used a four-step approach. First, tubular structures at multiple scales are extracted followed by automated plant shoot extraction. Then the Dijkstra shortest path method [?] is used to determine connectivity to a set of possible leaf candidates, followed by graph pruning. Pruning is done by deleting branches crossing gaps which are determined as too long and by deleting multiple parallel branches corresponding to the same thicker root. The automated root tracking algorithm of Leitner et al. [?] used a mechanistic root growth model to track roots in a graph representing 2D root systems obtained from NR images. Manual corrections were possible; in that case the Dijkstra algorithm was used to find the shortest path between two manually selected points. Building on Schulz et al., NMRooting [?] was developed. This tool is implemented using the Python programming language [?] and Mayavi for visualization [?]. NMRooting extracts a root skeleton by thresholding the input image by a noise dependent threshold, followed by dilation to



(a) Thresholded plant root MRI (b) U-Net segmentation (c) Extracted LCC (d) Extracted root structure graph

Fig. 2: Overview of root structure extraction. MRI (a) is segmented using a 3D U-Net [?] (b). The largest component connected to the shoot is identified (test-tube removed manually) (c). Gaps are closed and the root structure graph is extracted (d).

bridge smaller gaps. All other voxels above a given threshold are connected using the Dijkstra shortest path algorithm.

These methods are designed to work with high-quality 3D images. To improve MRI scans with insufficient resolution and SNR, deep learning based methods using a 2D RefineNet have been employed [?]. Zhao et al. [?] developed a 3D U-Net incorporating further input channels and loss modifications to increase resolution and SNR. The resulting super resolution segmentation still contains noise and disconnectivities. Therefore a root structure extraction method capable of dealing with noisy data is needed.

Graph shortest path based extraction methods are widely used in medical imaging as well, e.g. [?] or [?]. Lung airway extraction is similar to the root structure extraction problem in computing a 3D tree graph from a given 3D image. Jin et al. [?] use curve skeletonization based on radius quench points. To reduce the number of false subbranches, the extracted skeletal branches are dilated based on a radius estimate. Quench points inside this dilated branch are ignored. This ensures that for each airway branch only one graph branch is extracted. We follow this basic approach in our work.

For structural analysis of plant roots, voxel perfect localization of the graph nodes is not necessary. To reduce the number of nodes along elongated roots without deviating much from the original structure, the Douglas-Peucker-Algorithm [?] has proven to work efficiently.

III. Root Data and Root Graph

The proposed algorithm seeks to extract a structural tree graph G given a 3D image $I \in \mathbb{R}^{x_{max} \cdot y_{max} \cdot z_{max}}$, where x_{max}, y_{max} and z_{max} denote the size of the input volume. Further volume images like cost maps are named with capital letters and have the same size as I.

The voxels in the MRI images may have non-cubic aspect ratios, e.g. higher horizontal than vertical resolution. To take this into account for shortest path cost and radius computation a dimension multiplier $m_x \in \mathbb{R}$ is used. Another parameter needed is $int_{min} \in [0, 1]$, the minimum intensity in I believed to be root signal.

G is extracted on the voxel grid. Therefore, each position $I(x, y, z)$ is a potential graph vertex. Roots consist of tubular structures. These have a measure of connection and radius. For root analysis the measures are evaluated per vertex $v = (p, r, id) \in G$. $p \in \mathbb{N}$ is the vertex position, $r \in \mathbb{R}$ the local root radius and $id \in \mathbb{N}$ a unique branch id. To shorten equations $p, p_0, \ldots$ and 3 dimensional positions e.g. (x, y, z) are used interchangeably. The cost traversing from vertex v_1 to v_2 is the cost evaluated at the position of v_2. Each vertex uses its voxel-position 26-neighborhood as local connectivity.

IV. Extraction Pipeline

The root structure is extracted in two major steps:

Stage 1 extracts a largest connected component E given I. This process should reduce noise while keeping the root topology. For low noise, high connectivity images this step can be omitted.

Stage 2 takes E and extracts a corresponding root graph G. This graph should have a unique chain of nodes per root branch—independent of local radii.

V. Largest Connected Component Extraction

The largest connected component is extracted in five steps:

1) Generate radius transformation R given $occ_{min} \in [0, 1]$.
2) Generate cost map C_{gap} given $w_{rad} \in [0, 1]$ and offset $o \in [0, 0.1]$.
3) Extract shortest path C_{path} given $cost_{max} \in \mathbb{R}$.
4) For gap closing $l_{max} \in \mathbb{N}$ is given.
5) Extract LCC given $cost_{max}$ and int_{min}.

Algorithm 1 gives an overview of LCC extraction.

Algorithm 1 Stage I - LCC Extraction

```
Input: $I$, $config_I$, $config_E$
Output: $E$
  function EXTRACTVOLUME($I$, $p_0$, $m_x$, $int_{min}$, $config_E$)
      $R$ = radiusTransformation($m_x, int_{min}, occ_{min}$)
      $C_{gap}$ = costMap($p_0, w_{rad}, o$)
      $C_{path}$ = djikstraSPGapCl($C, p_0, m_x, cost_{max}, l_{max}$)
      $E$ = pathToExtraction($I, p_0, int_{min}$)
  return $E$
  end function
```

A. Radius Cost Map

Root signal should have low cost while any low intensity and noise signal should have high traversal cost. Using I as the basis solves the first requirement. To increase cost on high intensity noise patches, a weighted radius estimation at each position is added:

$$C_{proto}(p) = I(p) + w_{rad}\frac{R(p)}{\max(R)},$$
$$C(p) = 1 - \frac{C_{proto}(p)}{\max(C_{proto})} + o.$$

B. Shortest Path with Gap Closing

The Dijkstra shortest path algorithm is used to find connections. Starting from position p_0, the sum of the traversal cost for each voxel C_{path} is computed. If $C_{path}(p) > cost_{max}$, p is not further expanded and $C_{path}(p) = \infty$.

Disconnected areas in I should be joined by a unique connection. Practice showed that simply using Dijkstra and extraction could lead to merged areas similar to topological closing as the Dijkstra algorithm only minimizes path cost per voxel and not global graph cost.

To address noise and misalignment in I, small areas with non-root information should be bridged. At the same time areas connected by high intensity voxels should be explored before seeking gap closing. Therefore, the cost map is updated:

$$C_{gap}(p) = \begin{cases} C(p) \cdot 10 & \text{if } C(p) > cut_{gap} \\ C(p) & \text{otherwise} \end{cases} \tag{1}$$
$$cut_{gap} = (1 - int_{min}) \cdot \max(C) \tag{2}$$

Increasing the cost for voxels with intensity above cut_{gap} increases the stability of the gap closing.

The Dijkstra algorithm first connects all p connected with $C_{gap}(p) \leq cut_{gap}$ to p_0. To find disconnected areas $cost_{max}$ cutoff is ignored for up to l_{max} steps through $C_{gap}(p) \geq cut_{gap}$ positions. Should this reach an area in $l \leq l_{max}$ steps not explored by the shortest path algorithm before, the assumption is made a gap of length l was found. The point from which this gap bridging path started is p_s while the position found is p_e. The cost of p_e is now updated:

$$C_{path}(p_e) = C_{path}(p_s) + l \cdot C(p_e) \tag{3}$$

This is equal to a assuming the $edge(p_s, p_e)$ is part of the initial graph.

The low cost of p_e leads to the area connected to p_e being explored before more gap closing is done. From this area further gap closing is possible. Should another gap be found connecting to the area containing p_e, the area is already explored and therefore no more gap connections are established to it.

C. LCC extraction

Following the Dijkstra algorithm, all voxels $I(p) \geq int_{min}$ and $C_{path} < \infty$ are extracted by connecting them to p_0. This may include voxels $I(p) < int_{min}$ needed to connect all voxels of interest to p_0. Therefore:

$$path_p \equiv \{p_i \in I : p_i \in \text{ shortest path}(p_0, p)\}$$
$$ext(p) = I(p) \geq int_{min} \wedge C_{path}(p) < \infty$$
$$con(p) = \exists (p_i \in I : ext(p_i) \wedge p \in path_{p_i})$$

Using these criteria, the largest connected component volume E is:

$$E(p) = \begin{cases} 1 & \text{if } ext(p) \vee con(p) \\ 0 & \text{otherwise} \end{cases}$$

E contains a single fully connected area with value 1. Most noise areas are not part of this area while all major root areas should be. Therefore, morphological skeletonization can be applied to recover the root structure.

VI. Skeletonization

To extract the skeleton, a modified version of Jin et al. [?] is used:

Algorithm 2 Stage II - Skeletonization

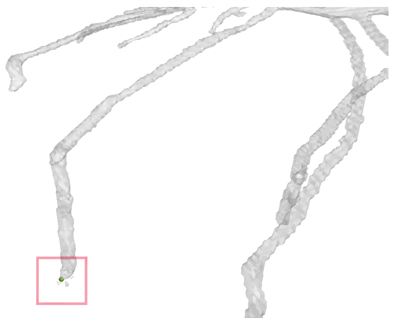
```
Input: $E$, $configI_I$, $configs_S$
Output: $G_{int}$
  function EXTRACTSKELETON($E$, $p_0$, $m_x$, $configs_S$)
    $R_E$ = radiusTransformation($m_x$, 0.5, 0, 9)
    $C_E = 1 - R_E/\max(R_{max})$
    $C_{path}$ = djikstraSP($C_E$, $p_0$, $m_x$, 100000)
    $G_{full}$ = curveSkeletonization($C_{path}$, $p_0$, $m_x$, $d_v$, $d_p$, $cut_z$)
    $G_{int}$ = interpolate($G_{int}$, $m_{max}$)
  return $G_{int}$
  end function
```

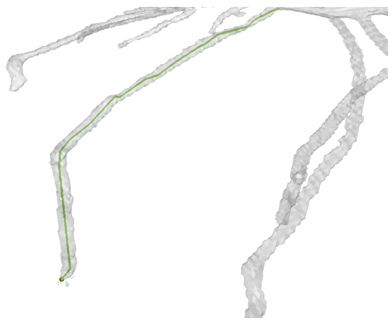
Here, a new cost map on E is generated. The extraction should follow the root centerline. Local radius information is used again. For this, the radius transformation R_E of E is generated using $occ_{min} = 0.9$. From this, the cost for skeletonization is:

$$C_E(p) = \begin{cases} 1 - \frac{R_E(p)}{\max(R_E)} & \text{if } R_E(p) > 0 \\ \infty & \text{otherwise} \end{cases}$$

R_E is also used to get a list of radius quench points. A quench point in this context is a point p so that $R_E(p)$ is larger than at least 21 of its 26 neighbours. The list of quench points $list_{quench}$ is sorted by the largest squared Euclidean distance to p_0.

As can be seen in Fig. 5a, some scans have a large flat area near the plant shoot. To exclude this from extraction cut_z is introduced. All quench points with a height between cut_z and the image border are ignored.


(a) The marked area contains a found quench point (green) for extraction


(b) Shortest Path quench point to start point (green, interpolated)

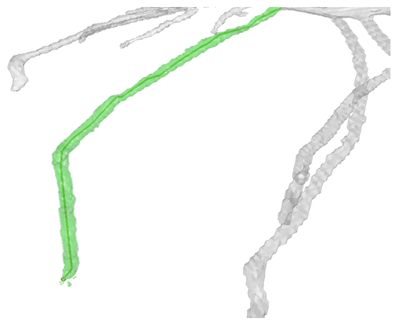

(c) Filled area around extracted path.

Fig. 3: Example skeletal branch extraction. From a found quench point (a) a path is connected (b) to the start point and the surrounding area is filled (c).

A. Build Skeleton

Algorithm 3 Curve Skeleton

```
Input: $list_{quench}$, $C_E$
Output: $G_{full}$
  function CURVESKELETON($C_E$, $list_{quench}$, $d_v$, $d_p$)
  Initialization:
      Volume $V_{fill}$, $V_{node}$ as zero Volumes
      Graph $G_{full}$ with $p_0$ as root
      $V_{node}(p_0) = G_{full}.root$
      int $br = 0$
  Main Loop:
      for $p_q \in list_{quench}$ do
          if $V_{node}(p_q) == 0$ then
              $path_{p_q}$ = shortest path $p_q$ to $p_0$
              addBranchtoG($G_{full}$, $path_{p_i}$)
              for $p_i \in path_{p_q}$ starting do
                  fillVolume($p_i$, $d_v$)
                  fillNodes($p_i$, $d_p$)
              end for
              $br$++
          end if
      end for
  return $G_{full}$
  end function
```

The shortest path on C_E is computed using the Dijkstra shortest path algorithm. By following the inverse radius map, the $path_{p_q}$ follows the center line of the branch. V_{fill} is 1 for all positions inside extracted branches. V_{node} contains the references to $n \in G_{full}$ in already extracted volume.

B. Adding a Branch

For all valid $p_q \in list_{quench}$ a branch is added to G_{full}. Like in [**?**] valid quench points are points outside already extracted branches. A branch is the shortest path $path_{p_q}$ connecting p_q to p_0.

Therefore, starting from p_q a graph node is generated for each $p_i \in path_{p_q}$, see Fig. 3b. If $V_{node}(p_i) \neq 0$, a node for position p_i already exists. Otherwise a new node $node_{p_i}$ is created and filled. Then the area around p_i is filled. Regardless $node_{p_{i-1}}$ is connected as a child node to $V_{node}(p_i)$.

C. Fill Volume

Whenever a new node is created, the area it corresponds to is filled, see Fig. 3c. Therefore, two Volumes V_{fill} and V_{node} are employed. Given position p_i, the area to be filled depends on $R_E(p_i)$. Flattened sphere masks S_r are employed.

To reduce the effect of noise, the radius is modified by d_v and d_p. Usually $d_v = 2d_p$ is used. The masks used at p_i have radius $r_v = R_E(p_i) \cdot d_v$ and $r_d = R_E(p_i) \cdot d_p$. Centered around p_i each voxel in V_{fill} for which $S_{r_v} = 1$ is filled with 1, for V_{node} each voxel $S_{r_d} = 1$ is filled with a reference to $node_{p_i}$.

D. Fill Node

For root structure analysis, each node must have its position in millimeter (mm), the corresponding radius in mm and a branch id unique for each root tip. Therefore to create $node_p$ the parameters p, radius $r = R_E(p)$, and $id = br$ is used. Once the full graph is extracted, the radii and positions are transformed to mm centered around p_0. The branch id is generated incrementally per root tip, therefore satisfying the uniqueness constraint.

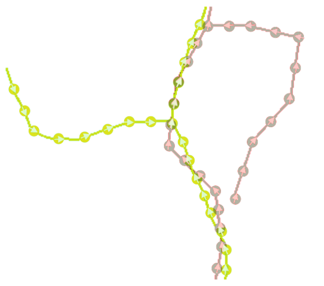
E. Interpolation

For root structure analysis, a per-voxel graph is not needed. To reduce the number of nodes, the graph is pruned. For this all root tips and nodes with more than one child in G_{full} are fixed. Then the Douglas Peucker Algorithm [**?**] for line simplification is applied on all paths connecting fixed points. The parameter used for the interpolation is $m_{max} \in \mathbb{R}$. The resulting graph G_{int} still has the same topology as G_{full} while the number of nodes per path is reduced significantly.

VII. Results

A. Evaluation Method

1) Evaluation Dataset: To evaluate the performance of the extraction algorithm 22 real plant root MRI scans are used.


(a) Extracted structure and manual reconstruction

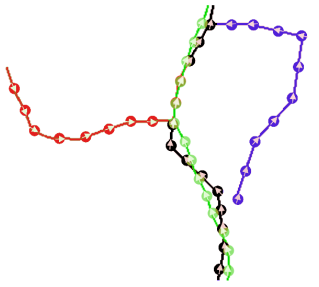

(b) Extracted structure and manual reconstruction after evaluation is applied

Fig. 4: Yellow: Extraction, Gray: Manual reconstruction. Green/Red: Extraction with/without corresponding manual reconstruction, Black/Blue: Manual reconstruction with/without corresponding extraction.

For each of these scans a manually annotated root graph is provided as target.

To decrease the noise in the original scan, 3D U-Net based root vs. soil segmentation [?] is applied. A total of five different models are used, with root weights 1, 10, 100, 1000 as well as the Log1 loss modification employed during training. The resulting dataset consists of $22 \cdot 5$ MRI segmentations.

A total of three parameters have been found to influence extraction performance: Cost Cutoff and Maximum Gap Length during LCC Extraction and Dilation Multiplier during curve skeletonization. Starting from a manually found parameter configuration for each file these parameters are tested using the multipliers 0.5, 1.0 and 1.5.

2) Evaluation Scheme for the Reconstructions: We developed a method to compare the extracted root structure with the manually reconstructed root.

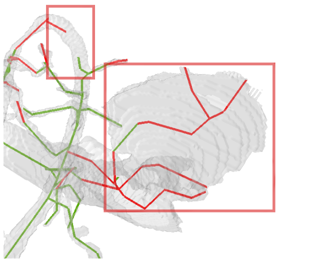
First, all vertices of the root graph are traversed, and the coordinates of each vertex together with the root direction at its location are recorded in a list l. Meanwhile, if the distance between two adjacent vertices is larger than a predefined spacing s, points on the line segment between these two vertices will also be recorded in l.

These intermediate points are interpolated so that the distance between each pair of adjacent points equals s. This step is done for both the target graph T and the extraction graph G to generate l_T and l_G, respectively (Fig. 4). The purpose of this step is to ensure that dense enough points are sampled from each root structure for further comparison.

Second, for each point p_T in l_T, each point p_G in l_G is examined to see if a correspondence between the two points can be established. Two conditions need to be satisfied in order to establish such a correspondence. The first one is that the angle between the root directions at the two points is not larger than 90 degrees. The second condition is that the distance between p_T and p_G is within a predefined distance tolerance d, or the distance from p_T to the line segment between p_G and its previous point p'_G is not larger than d.

TABLE I: Average Performance of extraction dependent on segmentation model, using cut_z. F1 is the average F1 score over all datapoints. It is not computed from average recall and precision. The distance tolerance used is 15. The models have been trained using different loss functions, see [?].

Segmentation	**F1-Score**	**Precision**	**Recall**
Log1	0.827854	0.833909	0.831218
Root Weight 1	0.776461	0.891080	0.700490
Root Weight 10	0.819381	0.846477	0.801895
Root Weight 100	0.768290	0.702945	0.868186
Root Weight 1000	0.69725	0.595281	0.889800


(a) The left marked area shows a root cut in halh. The other marked area is a flat connected area towards the top of the root. Multiple false branches are extracted inside this area.

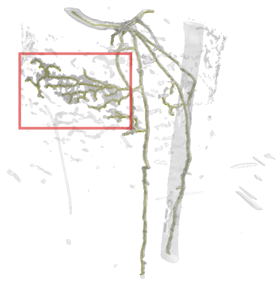

(b) Extraction inside a volume segmented using rw100. A large amount of noise merges with the root and is extracted.

Fig. 5: Green: Extraction with corresponding manual reconstruction, Red: Extraction without corresponding manual reconstruction.

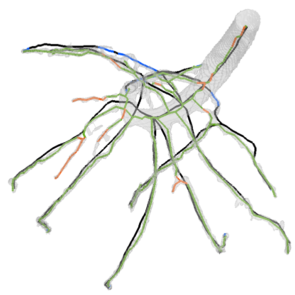
If more than one p_G satisfy the conditions above, the point with the closest distance to p_T is chosen to be the corresponding point. Once the correspondence between p_G and p_T is established, p_G is marked as used and cannot correspond to any other point in l_T.

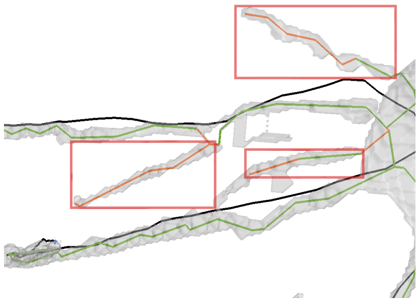
Finally, each point in l_G with a corresponding point in l_T is regarded as a true positive, each point in l_G without a corresponding point in l_T is regarded as a false positive, and each point in l_T without a corresponding point in l_E is regarded as a false negative. This is then used to compute recall, precision, and F1 score.

This method evaluates the extraction with respect to topology by penalizing large divergence in orientation and missing target in the immediate area. Also the extraction takes branch length into account by penalizing branches which are both to short or to long with respect to the target.

B. Performance

1) Performance based on Input: The algorithm's performance heavily depends on input quality. Therefore the performance of the initial U-Net based segmentation changes the performance of the extraction algorithm. Table I shows the average F1 score of the extraction algorithm dependent on the segmentation model. As can be seen, the Log1 loss modification gives the best F1 score by producing a balanced precision and recall.


(a) Full volume, extraction and manual reconstruction


(b) The marked areas contain elongated volumes resembling roots. These are not part of the target structure but may have been missed during manual extraction

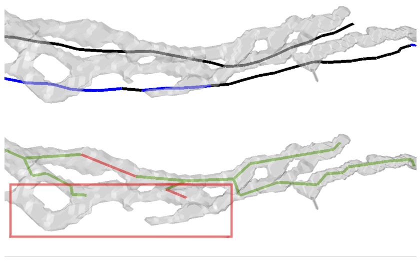
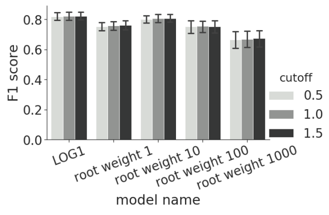
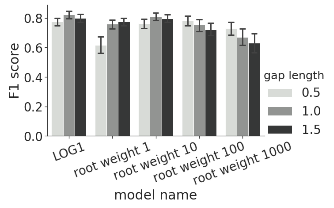

(c) Upper Half: Manual reconstruction, Lower Half: Extracted Structure. Marked Area: Instead of a complete lower branch the lower volume is connected along multiple mergepoints to the upper branch

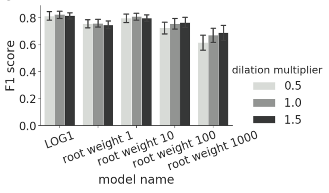
Fig. 6: Black: Manual reconstruction segments with corresponding extraction output. Blue: Manual reconstruction without corresponding extraction, Green: extraction with corresponding manual reconstruction , Red: extraction without corresponding Manual reconstruction.



(a) Changing $cost_{max}$



(b) Changing l_{max}



(c) Changing d_v

Fig. 7: Average performance of the extraction over the complete dataset when changing one of the parameters. No cut_z is used for these values.

The second best model utilizes a root loss weight of 10. The segmentation outputs have slightly enlarged root structures and noise structures. For most roots, the score is slightly lower than the Log1 score but the enlarged areas result in merged roots leading to wrong connection in more dense root systems. Another issue is an enlarged area towards the plant shoot of a root system leading to multiple wrong extractions reducing the precision noticeably. Fig. 5a shows this problem. This is addressed by employing cut_z.

Using a root weight of 1 combined with the same parametrization used for the other models results in noticeably lower recall. Further parameter testing showed that a much larger maximum gap length is needed. Using this, the recall can be increased but still stays below the other models.

The segmentation models utilizing larger root weights 100 and 1000 respectively, show problems similar to root weight 10. More dense root systems tend to merge resulting in structures combining two root branches that should be separate. Another problem is that the enlarged roots and noise areas merge in some scans leading to noise areas being treated as root, see Fig. 5b. These noise areas are large enough to have a non-trivial radius, limiting the use of radius based penalties.

2) Performance Based on Parameterization: Extraction performance was evaluated also with respect to three parameters that were shown to change extraction behaviour the most in manual assessment. Fig. 7 shows the behaviour of the extraction when altering these parameters.

Altering the cost cutoff $cost_{max}$ changes behaviour only minimally once above a certain threshold. Below this threshold $cost_{max}$ can be used to exclude higher cost areas from the extraction. This only works if unwanted areas can be sufficiently penalized. This is not the case for models with high root weighting. The threshold can vary from 15 in scans with overall high connectivity and large radius to 700 in scans containing long and thin root systems. This may lead to the changes in $cost_{max}$ having only a small effect on performance.

Changes in gap length l_{max} have a larger effect on performance. Well-chosen l_{max} can increase the performance significantly. This parameter depends on the input scan quality. As can be seen in Fig. 7b, l_{max} used and validated on models Log1 and root weight 10 is too large for the models using higher root weighting while the performance with root weight 1 segmentation increases noticeably with larger l_{max}.

The third parameter used is the dilation multiplier d_v for root extraction. Models with smaller more discernible structures, as can be found in root weights 1, 10 and Log1, show small change when changing d_v. These inputs have clearer structures leading to fewer false responses in surrounding areas. Models with high root weights often have areas creating quench points which are wrongfully included because of insufficient radius dilation during skeletonization. This can be reduced by increasing d_v as can be seen in Fig. 7c. Therefore

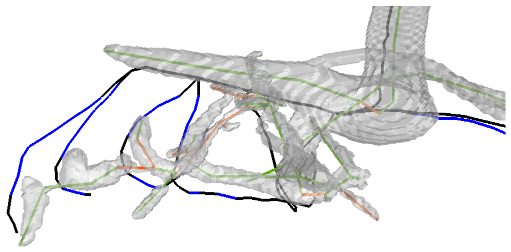


Fig. 8: Black: Manual reconstruction segments with corresponding extraction output. Blue: Manual reconstruction without corresponding extraction, Green: extraction with corresponding manual reconstruction , Red: extraction without corresponding manual reconstruction. The manual reconstruction shows intricate arc-like structures while the volume only holds these structures for a single branch. The missing volume information lead to wrong extraction.

this parameter can be used to increase precision in more noisy scans.

C. *Qualitative Assessment*

Fig. 6a shows a scan with high F1 score. The overall structure is extracted correctly. Some areas that appear to have an elongated although fractured structure are extracted as root while not being part of the target structure, see Fig. 6b. These may be roots that did not get annotated during manual extraction. While fractured the extraction creates a single connected response bridging gaps correctly.

Some structures in the segmentation merge. This leads the extraction algorithm to connect merged areas using a single branch instead of the two or more that should be used. Fig. 6c shows this. Two parallel branches merge in certain intervals leading to the lower branch not being extracted as a single branch but as segments connected to the upper branch.

This can also be observed in Fig. 5a. Towards the upper part, a root is clearly cut in half during extraction. This is due to the same structure merging creating wrong paths.

In Fig. 8 the area towards the center left has missing structure. The target shows curved downwards connections not found in the segmented MRI. Instead the lower parts are connected by the shortest gap.

As can be seen in Fig. 2d. The algorithm is capable of extracting root structures from large complicated root scans. Disconnected roots have been connected correctly using singular responses.

D. *Runtime and Memory Usage*

The test computer is a laptop with 2 cores at 2.3GHz and 8GB of RAM running Ubuntu 16.04 64-bit. Input size is between $140 \times 512 \times 512$ to $396 \times 512 \times 512$. Maximum memory usage is 10.9GB. Average memory consumption is 4-6GB. The three scans needing the most time in Fig. 9 need more than the available RAM. The increase in runtime is because of paging out of memory used by the algorithm.

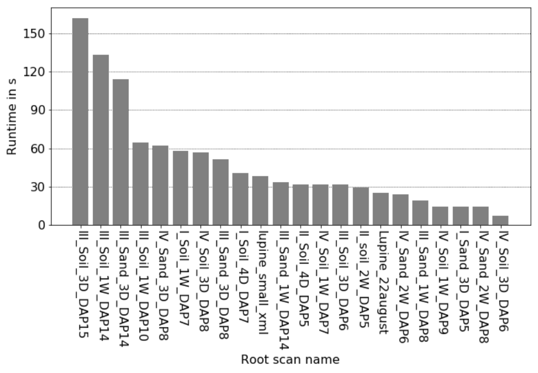


Fig. 9: Average runtime of the full algorithm dependent on input scan.

The other relevant factors concerning runtime are the maximum radius in structures and amount of extracted roots.

VIII. Discussion

In this paper, a pipeline to extract root structures from MRI images containing noise and disconnected structures has been presented. The pipeline takes a two stage approach to first reduce noise and connect disconnected areas, followed by curve skeletonization to extract a root skeleton from the scan.

The algorithm is capable of running on a laptop with limited resources fast and is capable of extracting large datasets of input images in a reasonable amount of time.

Evaluation on real MRI scans shows that performance depends on the quality of the input. Segmentations using Log1 loss modification and a slightly higher root weighting perform the best. These segmentations preserve most of the root structure while not enlarging them too much.

Two input properties negatively affecting extraction have been observed: The first one is areas of missing information in the input. This may lead to wrong connections as the algorithm takes the shortest path through the remaining volume.

The second property concerns merging root structures. The current algorithm does not take into account local branch orientation. This could be used to address the problem of merging roots as well as help with guessing correct connections missing in the input.

Overall, the algorithm is capable of extracting complex root structure from scans with low connectivity and high noise. It produced some root branches not found in the manual reconstruction, demonstrating its potential for improving not only speed, but also quality of the reconstruction.