Identification and Measurement of Individual Roots in Minirhizotron Images of Dense Root Systems

Alexander Gillert¹  Bo Peters²  Uwe Freiherr von Lukas¹,³  Jürgen Kreyling²

¹Fraunhofer Institute for Computer Graphics Research IGD, Rostock  
²Institute of Botany and Landscape Ecology, Greifswald University  
³Institute for Visual & Analytic Computing, University of Rostock  
{alexander.gillert, uwe.freiherr.von.lukas}@igd-r.fraunhofer.de  
{bo.peters, juergen.kreyling}@uni-greifswald.de

Abstract

Semantic segmentation networks are prone to oversegmentation in areas where objects are tightly clustered. In minirhizotron images with densely packed plant root systems this can lead to a failure to separate individual roots, thereby skewing the root length and width measurements. We propose to deal with this problem by adding additional output heads to the segmentation model, one of which is used with a ridge detection algorithm as an intermediate step and a second one that directly estimates root width. With this method we are able to improve detection and width measurements in densely packed root systems without negative effects on sparse root systems.

1. Introduction

Plant roots play a critical role in plant growth and many ecosystem processes and as such have become of increasing interest for ecosystem and climate modelling [1]. Despite their importance, research on belowground growth dynamics remains sparse due to inaccessibility of root systems and the often costly methods required for observation. With the development of rhizotrons and subsequently minirhizotrons, a nondestructive method for long-term monitoring of roots became available [5]. Rhizotrons often consist of two vertical glass panels separated by a thin layer of soil. Root growth in the soil along the inside of the transparent panels is then documented visually or photographically. Likewise, Minirhizotrons, transparent (acrylic-)glass tubes inserted into the soil, allow for in-situ monitoring of plant growth in natural conditions as root growth alongside the tube walls is documented with specialized scanners or cameras from within the tube.

The shape and size of plant root systems vary greatly between species and environmental con-
ditions. Whereas previous works on automated (mini)rhizotron imagery analysis [10, 7] mostly deal with plant species with sparse root systems with few or far apart growing roots (e.g. soy beans), in this work we are more interested in those with densely packed roots. Especially many graminoid (grass-like) species have a fibrous root system where all roots originate at the point where the aboveground plant body interfaces with the soil, resulting in densely packed root systems with roots growing in parallel and with little space in between.

Some of the most important metrics for plant growth research are total root length and root width. Estimating the root length in the images by simply taking the sum of all segmented root pixels leads to a bias towards large diameter roots. Therefore, most of the previous works [10, 7] employ a skeletonization postprocessing step to get a more accurate root length estimation. This works sufficiently well in images where individual roots are far apart, in scenarios with densely packed roots however, the segmentation network usually has difficulties classifying the boundaries between roots correctly and results in oversegmentation in those areas. The skeletonization method is then either unable to distinguish between individual roots and merges them into one, or even worse leads to loops which does not represent the true root system topology and results in incorrect measurements. This problem is illustrated in figure 1. One might argue that increasing the segmentation threshold would help to separate those roots, however this in turn would also negatively affect the detection of fine roots and width measurement.

We present a method to mitigate this problem by adding an intermediate ridge detection step. Specifically, we convolve a learned distance transform map with the second derivative of a gaussian kernel and analyze its response for curvature. Experimental results show a significant improvement in skeleton metrics for images with dense root systems, without negative impact on those with sparse ones. Moreover, we propose to estimate the width of the roots directly via regression which we have found to outperform baselines.

2. Related Work

Basic minirhizotron imagery analysis systems have been presented in [10, 7] and consist of a deep segmentation neural network based on the U-Net [6] architecture or similar. Research on improving the quality of results has focused on transfer learning [11] by pretraining on different plant species, data augmentation [7] in form of grid deformations, weak supervi-

3. Methods

We approach the described problem by adding additional auxiliary output heads to a segmentation network which learn the distance transform of the ground truth. One of them is used to detect ridges, i.e. continuous curves of local maxima and the second one is used for width estimation. A schematic overview of our method is provided in figure 2.
3.1. Network Architecture and Training

We use the U-Net [6] architecture as the base for our network and modify it to produce three output maps which we denote with \( y^{seg} \), \( y^{dist} \) and \( y^{width} \). Each output is computed by an own separate decoder network, connected to a shared encoder.

The main segmentation head is trained in a standard supervised manner with the binary crossentropy (BCE) loss, whereas the auxiliary heads are trained on the distance transform of the binary ground truth segmentation map \( y^* \). Specifically for \( y^{dist} \) we use the mean absolute error on pixels where \( y^* \) is positive and ignore pixels that are zero to avoid distraction:

\[
L_{dist} = y^* \cdot |y^{dist} - D(y^*)| \sum_i y^*_i 
\]

where \( D \) is the distance transform. During inference we zero out the predicted distance values where the output of the main segmentation head \( y^{seg} \) is below a threshold (0.5 where not stated otherwise). In the following we use \( y^{dist} \) as a shorthand notation for \( y^{dist} \cdot (y^{seg} > 0.5) \).

Although \( y^{dist} \) learns the distance transform and in theory could be directly used for width prediction, in practice this output head is biased towards small values because it also learns the pixels close to the root border. Therefore, we train the \( y^{width} \) head to directly estimate the width of the roots by only learning the distance transform of the center pixels:

\[
L_{width} = \frac{S(y^*) \cdot |y^{width} - D(y^*)|}{\sum_i S(y^*_i)} 
\]

where \( S \) is the skeletonization method [16, 9]. For both auxiliary heads no additional annotation except for the standard binary segmentation map is required.

The overall loss function is given by:

\[
L = L_{BCE} + \lambda_0 L_{dist} + \lambda_1 L_{width} 
\]

where \( \lambda_0 \) and \( \lambda_1 \) are balancing hyperparameters which we both set to 0.01. We have found larger values to have a negative effect on the main segmentation head.

We use the SGD optimizer for 15 epochs with a learning rate of 0.1, momentum 0.9 and reduce the learning rate twice by a factor of 0.1.

3.2. Ridge Detection

As discussed in more detail by Steger in [8], a well established method to detect lines in a one-dimensional function is to convolve it with the second derivative of a gaussian kernel and use the zero crossings of the result as the edges of the line. The second derivative of the gaussian kernel is defined as:

\[
\gamma''(x) = \frac{x^2 - \sigma^2}{\sqrt{2\pi}\sigma^5}e^{-\frac{x^2}{2\sigma^2}} 
\]
where $\sigma$ is the standard deviation hyperparameter which affects the width of the kernel. We use $\sigma = 3$ where not otherwise mentioned. This kernel converts the signal into a scale-space description and smooths out noise in the data.

Since it holds that $(g'' * f)(x) = (g * f)''(x)$, the response of this convolution can be regarded as the smoothed second derivative of the function $f$, i.e. it represents its curvature. By looking at negative values of the response, one can find right-handed turns in the function, i.e. local maxima.

For two-dimensional data, Steger [8] recommends applying the 1D convolution in the direction perpendicular to the line or ridge. This brings the drawbacks of firstly the need to estimate the angle at each line location and secondly many separate convolutions at different angles. In large images and with many ridges, as can be the case in minirhizotron data, this gets very costly. Therefore we opt for a simpler method of convolving in only two directions, namely row-wise and column-wise to get the response maps $r^{row}$ and $r^{col}$:

\[
\begin{align*}
    r_{ij}^{row} &= (g'' * y_{row}^{dist})_j \\
    r_{ij}^{col} &= (g'' * y_{col}^{dist})_i
\end{align*}
\]  

(5)

where $y_{row}^{dist}$ and $y_{col}^{dist}$ represent the $i$-th row and the $j$-th column of $y^{dist}$ respectively.

The magnitude of the response in $r^{row}$ is the largest for ridges running in vertical direction with still reasonable results diagonally but reduces to noise in the horizontal direction. The opposite applies to $r^{col}$. To get an acceptable response in all directions we combine both response maps with weight maps $w^{row}$ and $w^{col}$ which are constructed as:

\[
w^{row} = \frac{\delta_p(-r^{row})}{\delta_p(-r^{row}) + \delta_p(-r^{col}) + \epsilon}
\]  

(6)

where $\epsilon = 10^{-6}$ a small constant to guard against zero division and $\delta_p$ the dilation operation with a $\rho$ pixels sized structuring element. In practice we use the max-pooling operation with a kernel of size $\rho$. Where not otherwise mention we set $\rho = 11$. $w^{col}$ is defined analogously. These weight maps are then used to combine the two response maps into $\hat{r}$:

\[
\hat{r} = w^{row} r^{row} + w^{col} r^{col}
\]  

(7)

Next, we define ridges as pixels where $\hat{r}$ is negative and since the convolution operation might spill over, also check for $y^{seg}$:

\[
q = \begin{cases} 
1, & \hat{r} < 0.0 \text{ and } y^{seg} > 0.5 \\
0, & \text{otherwise}
\end{cases}
\]  

(8)

Lastly, we apply the skeletonization method [16, 9] on $q$ to estimate the centers of the roots which we denote with $S(q)$.

3.3. Width Estimation

A disadvantage of convolving only in two directions instead of perpendicular to the ridge direction is that the width of the detected ridges cannot be directly determined. Instead, we simply take the regressed values of $y^{width}$ where the center of the ridges was detected:

\[
W = \begin{cases} 
y^{width}, & \text{where } S(q) = 1 \\
\text{undefined,} & \text{otherwise}
\end{cases}
\]  

(9)

We have considered other choices, for example using the distance transform $D(y^{seg})$ instead. We evaluate this choice in the experimental section.

4. Experimental Setup

4.1. Dataset

Our training dataset consists of 64 minirhizotron images acquired from mesocosm and field experiments and an additional of 32 images is used for evaluation. The images mostly contain roots of Carex rostrata, Mentha aquatica and Equisetum fluviatile plant species and were acquired with a CI-600 In-Situ Root Imager (CID Bio-Science Inc.). All images stem from a facility that was established in fall 2018 at the Institute of Botany and Landscape Ecology in Greifswald, Mecklenburg Western Pomerania.

The dataset was carefully annotated by ecologists. Annotators were explicitly asked to leave a minimum one-pixel wide boundary inbetween the roots to make sure that the distance transform and skeletonization methods perform without issues on the ground truth.

For training and inference, the images are sliced into overlapping 512x512px patches with 32px overlap. Ridge detection and evaluations were performed on the stitched full-sized pictures.

For a more fine-grained evaluation we manually classify images as containing dense or sparse root systems. We count 22 sparse and 10 dense images in our test set. The seemingly low number of evaluation images is due to the very high cost of manual annotation which can be as high as 15 man-hours for a single image.
4.2. Metrics

Standard metrics that are often used in semantic segmentation like precision, recall or the Dice coefficient are not well suited to be directly used with skeletons due to their sparseness: a shift by a single pixel can have a large impact on the result. Therefore, our main metrics for the identification of roots are skeleton completeness $C_p$ and correctness $C_r$, which are discussed in more detail in [12] and defined as:

$$C_r = \frac{TP}{TP + FP} ~ C_p = \frac{TP}{TP + FN}$$

where $TP$, $FP$ and $FN$ are the buffered versions of the number of true positives, false positives and false negatives. They are defined with dilated skeletonized predictions $\delta_y(S(y))$ and dilated skeletonized ground truth images $\delta_y(S(y^*))$ as:

$$TP = S(y) \cap \delta_y(S(y^*))$$
$$FP = S(y) \cap \delta_y(S(y^*))$$
$$FN = \delta_y(S(y)) \cap S(y^*)$$

Intuitively speaking, a predicted skeletonized pixel is considered a true positive if it is within $\rho$ pixels distance of a skeletonized ground truth pixel. We use $\rho = 1$ for all our evaluation experiments.

Moreover, we evaluate the harmonic mean of both metrics calculated as $H = \frac{2C_pC_r}{C_p + C_r}$ (also known as the F1 score when using precision and recall) and the overall root length in an image, which we estimate with the sum of all skeletonized pixels.

We compare the width measurements via a histogram and a direct comparison metric. For the histogram metric we count the skeletonized root pixels into three categories based on the measured width: fine ($<3$px), medium (3-7px) and coarse ($>7$px). Then we compare the absolute and relative error of those bins. For the relative error we use the mean absolute percentage error (MAPE):

$$MAPE = \frac{1}{N} \sum \frac{|x^{true} - x^{predicted}|}{\max(\epsilon, x^{true})}$$

where $\epsilon$ guards against small values in the denominator. Since our ground truth values are not well scaled, i.e. can vary in a large range from zero to tens of thousands counted pixels within the same bin, we set it to the average value in the histogram bin: $\epsilon = \frac{1}{N} \sum x^{true}$.

This is also the metric that the end user would be most interested in, however it is dependent on the quality of the upstream root detection system: if a root is not detected it cannot be sorted into a bin. Therefore we also directly compare the widths at the $TP$ locations to isolate the width measurement evaluation from the detection.

The width ground truth is computed via the distance transform of the ground truth segmentation map.

4.3. Compared Methods

For skeleton metrics, we compare the following approaches:

- **Baseline**: Segmentation network trained only with binary cross-entropy loss. Skeletonization applied directly on the thresholded $y^{seg}$.
- **Weightmap**: Same as baseline, but trained with a weightmap as in [6] that puts additional emphasis on the pixels inbetween roots.
- **Multi-class segmentation** similar to that of [15]. For this method we train a segmentation network to classify each pixel into three classes, namely “root”, “border” and “background”. The annotation for the “border” class was automatically generated by applying the dilation operation onto the segmentation map with a 2px sized structuring element. For the evaluation we use only the pixels that were classified as “root”.
- **Ridge detection** applied on the segmentation head output $y^{seg}$. This method can be used with a normal segmentation network without adjustments to the architecture.
- **Ridge detection** applied on the auxiliary head output $y^{dist}$. This is our main method as described in section 3

For a fair comparison, neural networks for methods which do not require $y^{dist}$ were trained without the auxiliary heads.

For the width measurement we compare the regression based approach as in subsection 3.3 to measuring the width via distance transform on either the baseline skeleton or the skeleton as computed in subsection 3.2.

5. Results

Our main results for the skeleton metrics are presented in table 1. The multi-class segmentation method provides better skeleton correctness performance but at the cost of a worse skeleton completeness, thus no clear improvement is made with this
Table 1: Mean skeleton metrics of the compared methods. Bold font indicates best values.

<table>
<thead>
<tr>
<th>Method</th>
<th>Dense</th>
<th></th>
<th></th>
<th>Sparse</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cp↑</td>
<td>Cr↑</td>
<td>H↑</td>
<td>Cp↑</td>
<td>Cr↑</td>
<td>H↑</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.571</td>
<td>0.639</td>
<td>0.599</td>
<td>0.525</td>
<td>0.613</td>
<td>0.583</td>
</tr>
<tr>
<td>Weightmap</td>
<td>0.591</td>
<td>0.658</td>
<td>0.618</td>
<td>0.518</td>
<td>0.653</td>
<td>0.581</td>
</tr>
<tr>
<td>Multi-class Segmentation</td>
<td>0.509</td>
<td>0.649</td>
<td>0.568</td>
<td>0.460</td>
<td>0.665</td>
<td>0.543</td>
</tr>
<tr>
<td>Ridge Detection on $y_{seg}$</td>
<td>0.606</td>
<td>0.676</td>
<td>0.634</td>
<td>0.525</td>
<td>0.612</td>
<td>0.583</td>
</tr>
<tr>
<td>Ridge Detection on $y_{dist}$</td>
<td>0.596</td>
<td>0.733</td>
<td>0.653</td>
<td>0.533</td>
<td>0.676</td>
<td>0.603</td>
</tr>
</tbody>
</table>

Table 2: Mean width measurement errors for the compared methods. The values for the fine, medium and coarse histogram bins are average count errors/MAPE. Direct stands for directly compared width values at TP coordinates in pixel units. Bold font indicates best values.

<table>
<thead>
<tr>
<th>Method</th>
<th>Dense</th>
<th></th>
<th></th>
<th>Sparse</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fine↓</td>
<td>Medium↓</td>
<td>Coarse↓</td>
<td>Direct↓</td>
<td>Fine↓</td>
<td>Medium↓</td>
</tr>
<tr>
<td>Baseline</td>
<td>10837.6 / 0.299</td>
<td>7822.4 / 0.118</td>
<td>5228.9 / 0.755</td>
<td>0.997</td>
<td>2741.7 / 0.203</td>
<td>2280.7 / 0.410</td>
</tr>
<tr>
<td>Weightmap</td>
<td>9610.2 / 0.282</td>
<td>7015.0 / 0.123</td>
<td>3390.3 / 0.295</td>
<td>0.965</td>
<td>2720.6 / 0.203</td>
<td>2564.5 / 0.474</td>
</tr>
<tr>
<td>Ridge Detection</td>
<td>10860.5 / 0.303</td>
<td>6246.8 / 0.087</td>
<td>3201.1 / 0.285</td>
<td>0.905</td>
<td>2493.8 / 0.193</td>
<td>2333.8 / 0.427</td>
</tr>
<tr>
<td>$y^{width}$ Regression</td>
<td>9877.2 / 0.273</td>
<td>6357.1 / 0.089</td>
<td>1841.2 / 0.192</td>
<td>0.810</td>
<td>1972.8 / 0.180</td>
<td>1745.9 / 0.275</td>
</tr>
</tbody>
</table>

Figure 3: Influence of the segmentation threshold on the skeleton metrics and total root length. Advanced methods like ridge detection benefit more from a lower threshold than the baseline. (Evaluated on dense root systems.)

method. The weightmap method is overall more beneficial however with a slight degradation for sparse root systems.

For dense root systems, ridge detection applied to $y_{seg}$ improves both skeleton completeness (i.e. less false negatives) and correctness (less false positives). At the same time no significant deterioration in performance is observed for sparse root systems. It performs even better if applied on $y_{dist}$ where both dense and sparse metrics are improved. We explain this with the smoother surface of $y_{dist}$ compared to $y_{seg}$ which benefits ridge detection.

We note that higher performance is achievable by reducing the segmentation threshold as is shown in figure 3. In the baseline method, increasing or decreasing the threshold is mostly a tradeoff between false positives or false negatives. This is also illustrated in the example in figure 5 where either four roots get detected as two, or a fine root won’t get recognized. Ridge detection on the other hand overall benefits from lower thresholds. This is because the full information of the raw decimal values is used directly, instead of just binary thresholded values.

The threshold should also be adjusted to achieve a better total root length estimate. With the default threshold of 0.5 the baseline seemingly performs better on this metric, however, this is due to a higher number of false positives which balance out the general underestimation of the overall root length. With a lower threshold of around 0.2, our method gives the best length estimate of the compared methods.
Figure 4: Influence of hyperparameters on the performance. Top: The width of the gaussian kernel $\sigma$ has an optimum in the range $3$ to $4$. Bottom: The dilation parameter $\rho$ for the width map computation should have minimum value of $9$. Larger values do not lead to much improvement but only increase computation times. (Evaluated on dense root systems.)

For the width measurements in table 2 we observe that regression gives the best performance in almost all measured metrics, especially for sparse root systems. We attribute this to oversegmentation in the segmentation-based approaches. With fine roots, oversegmentation is due to the difficulty to reliably segment thin objects whereas with coarse roots this is rather due to their proximity to each other. Regression is immune to this.

Some qualitative results are shown in figure 6. More hyperparameter tests can be seen in figure 4.

6. Conclusion

We have presented a method for improving the topology reconstruction of dense root systems in minirhizotron images as well as width estimation of individual roots. This is done by performing an intermediate ridge detection step on a learned distance map before the commonly used skeletonization method. This helps to mitigate the oversegmentation of close or overlapping roots. Images with predominantly sparse root systems are not negatively affected.

As a slight drawback of our width estimation method can be regarded that it is based on regression, thus it functions as a black box and lacks interpretability. However, in our experiments it clearly outperforms baselines.

ACKNOWLEDGEMENTS

This work has been supported by the European Social Fund (ESF) and the Ministry of Education, Science and Culture of Mecklenburg-Vorpommern, Germany under the project “DigIT!” (ESF/14-BM-A55-0015/19).
Figure 6: Some qualitative example results. From left to right: input image, baseline skeletonization, our method. In the top row green pixels represent TP, red pixels FP, blue pixels FN. Note the blurriness due to high humidity in this image. The bottom row shows an example with a sparse root system, almost no differences between the two methods in this case. (Skeleton images dilated for better visibility.)

References


[7] Abraham George Smith, Jens Petersen, Raghavendra


