

This WACV 2023 paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the accepted version; the final published version of the proceedings is available on IEEE Xplore.

Tracking Growth and Decay of Plant Roots in Minirhizotron Images

Alexander Gillert¹ Bo Peters² Uwe Freiherr von Lukas^{1,3} Jürgen Kreyling² Gesche Blume-Werry^{2,4}

¹Fraunhofer Institute for Computer Graphics Research IGD, Rostock ²Institute of Botany and Landscape Ecology, Greifswald University ³Institute for Visual & Analytic Computing, University of Rostock ⁴Department of Ecology and Environmental Science, Umeå University {alexander.gillert, uwe.freiherr.von.lukas}@igd-r.fraunhofer.de

Abstract

Plant roots are difficult to monitor and study since they are hidden belowground. Minirhizotrons offer an in-situ monitoring solution but their widespread adoption is still limited by the capabilities of automatic analysis methods. These capabilities so far consist only of estimating a single number (total root length) per image.

We propose a method for a more fine-grained analysis which estimates the root turnover, i.e. the amount of root growth and decay between two minirhizotron images. It consists of a neural network that computes which roots are visible in both images and is trained in an unsupervised manner without additional annotations.

Our code is available as a part of an analysis tool with a user interface ready to be used by ecologists. ¹

1. Introduction

Quantification and close monitoring of plant root growth is of essential importance for many scientific fields, as only the inclusion of such data allows for accurate modeling of many ecosystem processes [20, 12]. For instance, depending on ecosystem type, 30-95% of plant biomass is accumulated below ground in the form of roots [13]. Rhizodeposition (release of organic compounds from roots) and root litter is a primary pathway for the transfer of atmospheric carbon into the soil [21].

Observation of plant roots, especially in-situ in the field, is difficult. Traditional methods, such as harvesting and washing out of roots, are highly destructive and can provide only a snapshot measurement as the plant is killed in the process, making observation of a single plant over longer periods infeasible. Therefore, the so called minirhizotron-technique [7] has become the most important tool for the monitoring of plant roots. Minirhizotrons are transparent tubes that are installed into the soil underneath a plant, commonly at a 45° angle. After this initial intervention, root growth at the soil-tube interface can be recorded without further disturbance with the help of specialized cameras or scanners that are inserted into the tube.

The most often used metric in root research is total root length. However, this metric does not capture the actual amount of growth that occurred inbetween measurements. Consider an observation station at a remote location such that measurements can be taken only once per year. The root length would stay roughly constant over this time period, not revealing the actual amount of root growth dynamics, i.e. their turnover. This is a real problem in root research and the time span does not have to be taken to such extremes. Fine roots which are of particular importance for water absorption and thus for the growth of the whole plant often have a life span of only a few days to weeks.

Our contribution aims to break down the total root length metric into its components, growth and decay, by comparing two images acquired on different days. An example of this problem can be seen in figure 1. Note that the problem is not simply about finding a single translation vector or a homography for a pair of images. Instead, we want to find a displacement for every root that exists in both images since roots can move over time as they grow, thereby changing the distance to each other nonlinearly. Moreover, soil can move as well for example due to swelling and shrinking caused by shifts in soil water content. This is especially prevalent in highly organic soils [9]. This movement often accounts for only a few pixels, but this is enough to distort measurements. Additional difficulties arise from the fact that roots can change their appearance (e.g. turn from white to red

https://github.com/alexander-g/Root-Tracking



Figure 1: Problem Overview. Left and center: two minirhizotron images from the same experiment acquired approximately one year apart. Right: The result of our root tracking method. White pixels represent roots that are present in both images, red pixels are roots that are only visible in the first image (decayed or obstructed), green pixels are only visible in the second image (likely new growth).

color) or from changing environmental and image acquisition conditions. Examples of those issues are illustrated in figure 2.

We approach this task by finding correspondences in the image pairs with a neural network trained to compare image patches of roots. These correspondences are then used to warp the root segmentation map of the first image onto the second one. Importantly, our method is trained in an unsupervised manner with contrastive and self-supervised losses and does not require additional annotations, except for training the root detection network which are often already available anyway.

2. Related Work

Since manual tracing of roots in minirhizotron images is a slow and tedious task, several automatic analysis systems have been already introduced such as those in [16, 19] which are based on a segmentation neural network such as the popular U-Net[14] architecture. Further improvement of the analysis results has been achieved with methods like data augmentations [16] such as grid deformations. The authors of [23] used transfer learning by pretraining on different plant species and [25, 26] use weak supervision with multiple instance learning to reduce the amount of required annotations. In [2, 3] inpainting has been used against undersegmentation i.e. to correct gaps in segmentation results and in [6] oversegmentation of plants with dense root systems has been mitigated.

All of the works mentioned above only use a single image, i.e. one point in time, and mainly aim to improve the accuracy of the total root length measurements. To the best of our knowledge, no research has been published on the analysis of root growth from comparison of time series of minirhizotron images.

There exist several methods for other types of acquisition systems which take the temporal component into account. ChronoRoot[5] for example combines CNNs with LSTMs. Yet, these methods are dealing with image data from highly controlled and standardized experiments, e.g. single plants grown in petri dishes on transparent agarized medium and a fixed camera system with high temporal resolution such as PhenomNet [24]. These methods are difficult to apply to uncontrolled real-world environments and over longer periods of time.

A well established method to find correspondeces in two or more images is to use local feature descriptors such as the scale invariant feature transform (SIFT)[11], which computes a 3D histogram of local oriented gradients around a keypoint. It is widely used for problems like image stitching or localization and mapping. We have tried out this method but have found its performance to be insufficient in our case. A comparison with our method can be found in the evaluation section.

Neural network based feature matching methods such as SuperGlue [15], LoFTR [17] or COTR [10] promise better performance, however contrary to our method, they require ground truth annotations which are expensive to obtain with our images. Moreover, LoFTR has insufficient precision for fine roots as it only matches 8×8 patches.

A somewhat similar problem is *deformable image registration* from the medical domain, with methods such as VoxelMorph [1]. The goal here is to find a dense correspondence field that aligns two images. Specifically Voxel-Morph can be trained in an unsupervised manner by min-



Figure 2: Some of the challenges encountered in this task. (a) Roots and soil can move over time, hence finding a homography is insufficient. (b) Roots can change their appearance, for example here their color. (c) Soil can change appearance, for example due to different moisture levels.

imizing the difference in RGB or grayscale values of the two input images. However, this task is not identical to our problem because in our case, roots that are present in one image are often missing in the second one (newly grown or decayed roots). Moreover, these techniques often make use of templates, i.e. a kind of prototypical representation of organs like brain, lungs etc, which is not applicable to our problem. We have evaluated the original VoxelMorph source code nevertheless, but have encountered some issues with it as explained in the evaluation section.

3. Method

3.1. Overview

Given two minimization images x0 and x1 from the same experiment where x0 is the chronologically earlier one, we want to compute a dense displacement field that maps every pixel of x0 onto the corresponding location in x1 with a particular focus on correctness at the location of the roots.

We assume a pre-trained semantic segmentation network that can classify each pixel as "root" or "not-root". Such systems have been presented in [16, 19]. Specifically ,we use an architecture based on U-Net [14] with a pretrained backbone, but the method does not depend on this choice, other segmentation networks would work as well. In the following, this network is denoted with f with $f(x)_i = 1$ being a detected root at pixel coordinate i in image x.

The core of our root tracking system consists of a second neural network g that is trained to learn the similarity of two image patches containing roots. We use the same architecture as for the root segmentation network except that we remove the last layer so that it returns the *c*-dimensional feature embeddings for each pixel (c = 32).

To compute how similar two locations in two images are, we use RoIAlign [8] to extract a $(d \times d \times c)$ -sized descriptor with a box size of $(b \times b)$ of the output g(x). Where not otherwise mentioned we use b = 64, d = b/4. $g(x)_i$ denotes an extracted descriptor from image x at coordinate i. We then normalize the descriptors along the channel dimension and compute the cosine similarity.

3.2. Training

Our system is trained in two stages, neither of which requires additional annotations. In both stages, we use the weights of the semantic segmentation network f as the initialization parameters and train with the SGD optimizer with a learning rate of 0.01 and a momentum value of 0.9 for 10 epochs where not otherwise mentioned. We have found the network to overfit quite easily as analyzed in the supplementary material.

3.2.1 Training Stage 1: Contrastive Learning

We want the neural network g to return embeddings with a high similarity for descriptors extracted at the same root and low similarity for descriptors extracted at different roots. Since we do not know which roots from two images correspond to each other, we first train on descriptors extracted at the same location in the same image. To avoid trivial solutions, we utilize augmentations, as commonly used in contrastive learning methods [4].

Contrary to those generic contrastive learning methods, we are limited in the types of augmentations we can use for positive or negative views. For positive views we only use pixel-wise image transforms like color, contrast and brightness jitter to simulate different acquisition and environment conditions. For negative views, rotations and flipping operations are used. The idea here is that although roots can move, they rarely change their shape in a local area. This



Figure 3: (a) Schematic overview of our root tracking system. (b) In the first training stage we train on augmented views of the same image. For positive views only pixel-wise augmentations like brightness and color jitter are used. Negative views consist of rotatations or flips of the same location and other locations in the same image. In the second training stage we use different images.

way, the network is trained to compare by shape rather than color or texture. Descriptors that were extracted at different locations serve as additional negative views. Example images of the augmented views are shown in figure 3b.

The losses for this stage are defined as:

$$\mathcal{L}_{pos}^{stage1} = \frac{1}{|I|} \sum_{i \in I} -\log g(x)_i \ g(\phi_{pos}(x))_i \tag{1}$$

$$\mathcal{L}_{neg}^{stage1} = \frac{1}{|I|^2} \sum_{i \in I} \sum_{j \in I} -\log 1 - g(x)_i \ g(\phi_{neg}(x))_j \quad (2)$$

where I is a random subset of the root pixel locations $\{i \mid f(x)_i = 1\}$ while ensuring that each element of this subset has a minimum distance to others and ϕ_{pos} , ϕ_{neg} are the positive and negative augmentation transforms respectively. In images where no roots could be found, completely random points are used. The final loss is then defined as $\mathcal{L}^{stage1} = \mathcal{L}_{pos}^{stage1} + \mathcal{L}_{neg}^{stage1}$.

3.2.2 Training Stage 2: Self-supervision

The network above is already able to recognize same roots from different images quite well but it struggles with image pairs where the environment has changed considerably. A second training stage on different images is needed.

For this, we use the stage 1 model to find a set of correspondences M in different images. These correspondences

are then used instead of the augmented views of the same image. The positive loss is modified to:

$$\mathcal{L}_{pos}^{stage2} = \frac{1}{|M|} \sum_{i,j \in M} -\log g(x_0)_i \, g(x_1)_j \tag{3}$$

with $i \in I$ a keypoint location in image x0 and $j \in J$ in image x1. The other losses are modified accordingly.

3.3. Inference

3.3.1 Root Matching

We are only interested in matching roots with each other, therefore we directly use the coordinates where f(x) = 1as keypoints. To reduce the number of points to match, we apply the skeletonization method [27] on the segmentation map. This effectively means that we only match the centers of roots to each other.

Similar to [11] we perform cross checking and a ratio test to reject descriptors that have ambiguous matching partners, i.e. we ensure that the best match has a significantly higher cosine similarity to the second-best match. Since our keypoints are non-sparse, i.e. many have a distance of 1 to each other, the descriptors of neighboring keypoints are often the best and second-best match. Therefore, when searching for the second-best match we exclude keypoints within a certain euclidean distance from the best match. Expressed more formally, the computation of the ratio looks as follows:

$$r_{i} = \frac{\max_{j \in J} g(x_{0})_{i}g(x_{1})_{j}}{\max_{k \in K} g(x_{0})_{i}g(x_{1})_{k}}$$

$$K = \{k \mid \operatorname{dist}\left(\arg\max_{j} g(x_{0})_{i}g(x_{1})_{j}, k\right) > t\} \subseteq J$$

$$(4)$$

where dist is the euclidean distance of the keypoint locations and t a minimum distance threshold which set set to t = 64.

As a final step in the keypoint matching procedure, outlier rejection is performed. Although roots can move over time, this movement is usually limited to small distances. Therefore, we filter out matches that deviate from the median displacement vector by a threshold.

We define an image pair as successfully matched if there are at least two matched keypoint pairs left that have a minimum distance of 64 pixels to each other.

3.3.2 Alignment and Turnover Map

Given a set of matched root keypoints in both images we warp the root segmentation result $f(x_0)$ onto $f(x_1)$. To do this, we construct a dense deformation field by performing triangulation on the matched keypoints of image x_0 . The new coordinates are then computed in each triangle via linear barycentric interpolation of the corresponding keypoints coordinates in x_1 [18]. Values outside of the matched keypoints' convex hull are extrapolated by adding additional points to the four corners of the image x_0 with values found via nearest neighbor interpolation.

Lastly, we construct the final result which we term a (root) *turnover map* containing the three classes "same" (root is detected in both images), "decay" (root is detected only in the first image) and "growth" (root is detected only in the second image):

$$T := \begin{cases} same, & warp(f(x_0)) = 1 \text{ and } f(x_1) = 1 \\ decay, & warp(f(x_0)) = 1 \text{ and } f(x_1) = 0 \\ growth, & warp(f(x_0)) = 0 \text{ and } f(x_1) = 1 \end{cases}$$
(5)

where warp aligns the segmentation map $f(x_0)$ to $f(x_1)$ as described above.

More important for the end user is the total length of the newly grown or decayed roots. The skeletonization method [27] is commonly used to estimate this property for binary segmentation maps, however it cannot be used for multiclass ones. Applying it on a single class is not an option either as it would create artifacts at root borders, for example when a root gains width. Therefore, we redefine it for our case as combinations of the binary outputs and the turnover map T:

 $S(T) \coloneqq \begin{cases} same, & S(f(x_1)) = 1 \text{ and } T = same \\ decay, & S(warp(f(x_0))) = 1 \text{ and } T = decay \\ growth, & S(f(x_1)) = 1 \text{ and } T = growth \end{cases}$ (6)

where S(x) is the skeletonization method applied on a binary image x. The total lengths are then estimated with the sum of skeleton pixels of S(T) over each class.

4. Experimental Setup

4.1. Datasets and Annotation

Our main dataset consists of 2550×2273 px minirhizotron images acquired with a CI-600 In-Situ Root Imager (CID Bio-Science Inc.). The images stem from mesocosm (outdoor pot experiments under semi-controlled, roughly constant conditions) and field (outdoor and uncontrolled) experiments. Overall 854 unannotated images were used for the training which mostly contain roots of Carex rostrata, Mentha aquatica and Equisetum fluviatile plant species.

For the evaluation, we have annotated additional 62 image pairs. We have not considered image pairs in which even annotators were not able to find correspondences. The annotation consists of matched root keypoints that human annotators regarded to be the same root in both images and the corresponding turnover maps. Creating such an annotation from scratch is an extremely tedious and slow process, therefore annotators were tasked only to correct mistakes caused by the root matching algorithm that was presented in section 3, i.e. to add missing matches or to remove incorrect ones. Despite this simplification, a single image pair can take up to two hours to annotate due to the high image resolution and sometimes many fine roots.

The annotation was created with a custom user interface which was built specifically for this task. The user interface allows to add new matches either by clicking on corresponding locations in both images or by clicking and dragging within a single turnover map. This process is illustrated in figure 4.

To test how well our method adapts to other datasets we additionally use the Sunflower subset of the PRMI dataset [22] as a secondaray dataset. This dataset has an agricultural background for which root turnover is of lesser importance. For a fair evaluation all networks were retrained on this dataset only. It is used only where explicitly mentioned, otherwise the data refer to the main mesocosm and field dataset.

4.2. Evaluation Metrics

We use the following metrics for the evaluation:

• Intersection over Union (IoU) applied on the classes "same" and "growth" of the turnover map. Note that the class "decay" cannot be used because the annotation is focused on matching roots that are present in



Figure 4: Our main annotation procedure. Users can iteratively correct mistakes in the turnover map, which we then use for the evaluation. Note that the branching decayed root (red) at the bottom changes its position in the process. The intersection with the original prediction is zero and the IoU metric would count this as incorrect although it actually isn't (the class was correctly predicted). This metric cannot be applied on the "decay" class for this reason.

both images. The location of decayed roots cannot be accurately annotated in our turnover map. This problem is also explained in figure 4. IoU is widely used in segmentation tasks but in our case this metric can be unreliable due to the mode of annotation, as dicussed in subsection 5.3.

- **Relative error** in the total number of "same", "growth" and "decay" **pixel counts** in the turnover map *T*. This can be interpreted as an estimate of total root biomass.
- **Relative error** in the total number of "same", "growth" and "decay" skeleton pixels of the skeletonized turnover map S(T). This metric can be interpreted as an estimate of total root **length**.
- Matching **success rate** with the success condition as defined in 3.3.1

4.3. Compared Methods

Since there are no dedicated methods for tracking roots in minirhizotron images yet, we source similar methods from other domains as baselines for the comparative evaluation. We have considered the following alternatives and compare them here to our method:

- **SIFT** [11] serves as a minimal baseline to compute correspondences in an image pair. These are then fed into the pipeline described in subsections 3.3.1 and 3.3.2 to create a turnover map.
- **SIFT** is not aware of the root locations in the image and thus uses keypoints at arbitrary locations. To improve performance we add additional keypoints at the **skeleton pixels**. The scale parameter for these keypoints is estimated from the median of the original SIFT keypoints. Denoted with SIFT(sk).

- VoxelMorph [1] by itself struggles to align two minirhizotron images because the offsets can be very large. Therefore, we use it as a second step after applying the SIFT method. Specifically, we train it on the segmentation maps warp(f(x0)) warped with the correspondences found with SIFT. This is the best combination that we have tried. We have used the source code as published by the authors.
- Feature embeddings of the root segmentation network f instead of g. The rationale here is that f already knows what roots look like and thus might be enough to compare them. This method has the advantage that only one network has to be trained. Denoted with f Emb.

As of our method, we compare the two different training stages:

- **Stage 1**: Trained via contrastive learning only on augmented views from the same image. This stage is described in subsection 3.2.1.
- **Stage 2**: Trained via self-supervision on different images. This is our main method and is described in subsection 3.2.2.

5. Results

Additional evaluations and full sized results can be found in the supplement.

5.1. Main Results

Our main evaluation results are presented in table 1. We observe a significantly better performance of our method on all evaluated metrics. Simply using the traditional SIFT method alone can lead to deviations of almost 50% in root length measurements. Adding additional keypoints at the root locations does help but it is still outcompeted by our method. VoxelMorph is able to align roots quite well to each other as long as they are present in both images but struggles with roots that are visible only in one of the images. As can be seen in figure 5 it often tries to extend roots from the first image into newly grown ones and even more often shrinks down decayed roots to a thin line, resulting in large errors in the "decay" class. Despite good length errors on the "same" and "growth" classes, this issue makes it less trustworthy and explainable than our method. Additionally, errors are more difficult to correct manually.

The results on the PRMI dataset (table 2) are similar. One notable difference is that the feature embeddings of the root segmentation network f perform better than our stage 1 model. We attribute this to less environmental variance, so that pixel-wise augmentations have a smaller positive effect.

	Mesocosms & Field $(n = 62)$					
Method	IoU↑	Counts↓	Lengths↓	Success Rate↑		
	s/g	s/d/g	s/d/g			
SIFT	.59/.62	.30/.29/.38	.35/.44/.45	77.4%		
SIFT(sk)	.65/.67	.22/.27/.25	.28/.41/.30	88.7%		
SIFT+VoxelMorph	.76/.64	.13/.75/.18	.10/.52/ .10	88.7%		
f Emb.	.69/.69	.18/.23/.20	.23/.33/.25	88.7%		
Ours (Stage 1)	.75/.74	.12/.20/.12	.13/.19/.13	93.5%		
Ours (Stage 2)	.83/.81	.08/.10/.09	.09/.12/.10	93.5%		

Table 1: Main results. Bold font indicates best values. Counts and Lengths are relative error values. s/d/g stands for the turnover map classes "same", "decay" and "growth".



Figure 5: Typical failure case of VoxelMorph. From left to right: root segmentation maps f(x0), f(x1), VoxelMorph output, i.e. warped f(x0). VoxelMorph is prone to hallucinating new roots (indicated by green arrows) or shrinking decayed ones (red arrows).

	PRMI Sunflower $(n = 20)$					
Method	mIoU↑	Counts↓	Lengths↓	Success↑		
SIFT	0.51	0.82	1.17	80%		
SIFT(sk)	0.58	0.53	0.77	100%		
f Emb.	0.77	0.18	0.22	80%		
Ours (Stage 1)	0.72	0.28	0.38	90%		
Ours (Stage 2)	0.86	0.12	0.16	90%		

Table 2: Results on the PRMI dataset. Bold font indicatesbest values. Counts and Lengths are relative error values.

5.2. Time Dependence

Additionally, we compare how the performance depends on the acquisition time difference of the compared images. The longer the time span the larger the changes in roots and soil, thus in theory, the performance should go down. Figure 6 shows the same metrics evaluated on a subset of the data above, broken down into 5 time frames from one week to half a year. Specifically, image pairs from 4 mesocosm experiments were used, each pair starting with the same image. As expected, the performance is best over shorter time periods, with no longer than 2 weeks being an optimum for our method. Contrary to our expectation, training on different images (i.e. stage 2) does not extend the optimum time span significantly. Instead, its main advantage is simply being able to match more keypoints as depicted in the additional figures in the supplementary material. Changing environmental conditions are still a challenge for it. Vanilla SIFT has even problems at detecting any correspondences at all over longer periods with matching success rate dropping down to zero after two months.

5.3. Annotation Bias Analysis

The annotation as used above, was created from the output of the stage 2 model as the starting point. As a result, this annotation might be biased towards this model because annotators might have skipped areas which they deemed good enough. Only a few pixels of shift in one direction might accumulate in a significant amount of bias especially for the IoU metric. An example of this type of bias is shown in figure 7.

To analyze the extent of this bias, we have re-annotated a random 10% of the same image pairs by correcting the output of the stage 1 model which we denote with Ann_{S1} . The same 10% of the previous annotation (with stage 2 as the starting point) is denoted as Ann_{S2} . We have re-evaluated the metrics for this new annotation, the results are shown in table 3.

	Mean IoU↑		Mean Counts↓		Mean Lengths↓	
	Ann_{S1}	Ann_{S2}	Ann_{S1}	Ann_{S2}	Ann_{S1}	Ann_{S2}
Ann_{S1}	-	0.845	-	0.016	-	0.014
Ann_{S2}	0.845	-	0.016	-	0.014	-
Stage 1	0.794	0.772	0.151	0.144	0.153	0.141
Stage 2	0.793	0.831	0.099	0.093	0.110	0.097

Table 3: Re-evaluated metrics for different annotations.Bold font indicates best values for each annotation.



Figure 6: Influence of the acquisition time difference on the IoU, length error and success rate.



Figure 7: Typical source for bias in the annotation. Left: output of stage 1, right: stage 2. The stage 1 model did not match this root perfectly, yet both cases were left untouched by annotators thus favoring the corresponding model during evaluation. The shift amounts to only a few pixels but it accumulates along the length of the root.

As can be seen, a bias is certainly present since the performance gain of the stage 2 model is completely gone when comparing with the IoU metric on the Ann_{S1} annotation. This bias is much less pronounced on the relative pixel count error and relative length error metrics on which the stage 2 model still performs significantly better. The same pattern can be observed when comparing both sets of annotations to each other. The rather low IoU value of 0.845 indicates a high variation on a per-pixel level, but this goes down to less than 2% for the image-level metrics.

We conclude that IoU is unreliable for our mode of annotation and focus should be put on the other metrics instead, which are of higher importance for the end user anyway.

6. Concluding Remarks

Our measurements of newly grown and decayed roots are only estimates. For one, large movements of roots still pose a challenge as illustrated in figure 8. Secondly, roots can get obstructed by soil which our method would count



Figure 8: Failure cases of our method.

(a) Large movement of roots poses a problem since our descriptors are trained to learn the shape and are not rotation invariant. (b) A root hidden behind others can be classified as "same" in the turnover map due to its rather simple definition of overlapping segmentations.

as decay. This is rather an inherent limitation of minirhizotron imagery. Future work might focus on improving the matching performance for longer time periods. Voxel-Morph shows partially promising results but has drawbacks that need to be addressed. A combination with our work might be possible.

Our method is already in a usable state and we hope that it can enable new insights in the field of root research.

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).

References

- Guha Balakrishnan, Amy Zhao, Mert Rory Sabuncu, John V. Guttag, and Adrian V. Dalca. Voxelmorph: A learning framework for deformable medical image registration. *IEEE Transactions on Medical Imaging*, 38:1788–1800, 2019.
- [2] Hao Chen, Mario Valerio Giuffrida, Sotirios A Tsaftaris, and Peter Doerner. Root gap correction with a deep inpainting model. In *BMVC*, page 325, 2018.
- [3] Hao Chen, Mario Valerio Giuffrida, Peter Doerner, and Sotirios A Tsaftaris. Adversarial large-scale root gap inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.
- [4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020,* 13-18 July 2020, Virtual Event, volume 119 of Proceedings of Machine Learning Research, pages 1597–1607. PMLR, 2020.
- [5] Nicols Gaggion, Federico Ariel, Vladimir Daric, ric Lambert, Simon Legendre, Thomas Roul, Alejandra Camoirano, Diego H Milone, Martin Crespi, Thomas Blein, and Enzo Ferrante. ChronoRoot: High-throughput phenotyping by deep segmentation networks reveals novel temporal parameters of plant root system architecture. *GigaScience*, 10(7), 07 2021.
- [6] Alexander Gillert, Bo Peters, Uwe Freiherr von Lukas, and Jürgen Kreyling. Identification and measurement of individual roots in minirhizotron images of dense root systems. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, pages 1323–1331, October 2021.
- [7] AC Hansson and ELIEL Steen. Root growth of daily irrigated and fertilized barley investigated with ingrowth cores, soil cores and minirhizotrons. *Swedish Journal of Agricultural Research (Sweden)*, 1992.
- [8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [9] Colleen M Iversen, Meaghan T Murphy, Michael F Allen, Joanne Childs, David M Eissenstat, Erik A Lilleskov, Tytti M Sarjala, Victoria L Sloan, and Patrick F Sullivan. Advancing the use of minirhizotrons in wetlands. *Plant and Soil*, 352(1):23–39, 2012.
- [10] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for matching across images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6207–6217, October 2021.
- [11] David G Lowe. Distinctive image features from scaleinvariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [12] M Luke McCormack, Elizabeth Crisfield, Brett Raczka, Frank Schnekenburger, David M Eissenstat, and Erica AH Smithwick. Sensitivity of four ecological models to ad-

justments in fine root turnover rate. *Ecological modelling*, 297:107–117, 2015.

- [13] Karel Mokany, R John Raison, and Anatoly S Prokushkin. Critical analysis of root: shoot ratios in terrestrial biomes. *Global change biology*, 12(1):84–96, 2006.
- [14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.
- [15] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 4937–4946, 2020.
- [16] Abraham George Smith, Jens Petersen, Raghavendra Selvan, and Camilla Ruø Rasmussen. Segmentation of roots in soil with u-net. *Plant Methods*, 16(1):1–15, 2020.
- [17] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 8918– 8927, 2021.
- [18] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. implemented as scipy.interpolate.LinearNDInterpolator.
- [19] Tao Wang, Mina Rostamza, Zhihang Song, Liangju Wang, G McNickle, Anjali S Iyer-Pascuzzi, Zhengjun Qiu, and Jian Jin. Segroot: a high throughput segmentation method for root image analysis. *Computers and Electronics in Agriculture*, 162:845–854, 2019.
- [20] Jeffrey M Warren, Paul J Hanson, Colleen M Iversen, Jitendra Kumar, Anthony P Walker, and Stan D Wullschleger. Root structural and functional dynamics in terrestrial biosphere models–evaluation and recommendations. *New Phytologist*, 205(1):59–78, 2015.
- [21] JM Whipps and JM Lynch. Substrate flow and utilization in the rhizosphere of cereals. *New phytologist*, 95(4):605–623, 1983.
- [22] Weihuang Xu, Guohao Yu, Yiming Cui, Romain Gloaguen, Alina Zare, Jason Bonnette, Joel Reyes-Cabrera, Ashish Rajurkar, Diane Rowland, Roser Matamala, et al. Prmi: A dataset of minirhizotron images for diverse plant root study. *AI for Agriculture and Food Systems (AIAFS) Workshops at the AAAI conference on artificial intelligence*, 2022.
- [23] Weihuang Xu, Guohao Yu, Alina Zare, Brendan Zurweller, Diane L. Rowland, Joel Reyes-Cabrera, Felix B. Fritschi, Roser Matamala, and Thomas E. Juenger. Overcoming small

minirhizotron datasets using transfer learning. *Computers* and *Electronics in Agriculture*, 175:105466, 2020.

- [24] Robail Yasrab, Michael P Pound, Andrew P French, and Tony P Pridmore. Phenomnet: bridging phenotype-genotype gap: a cnn-lstm based automatic plant root anatomization system. *bioRxiv*, 2020.
- [25] Guohao Yu, Alina Zare, Hudanyun Sheng, Roser Matamala, Joel Reyes-Cabrera, Felix B Fritschi, and Thomas E Juenger. Root identification in minirhizotron imagery with multiple instance learning. *Machine Vision and Applications*, 31(6):1–13, 2020.
- [26] G. Yu, A. Zare, Weihuang Xu, R. Matamala, J. Reyes-Cabrera, F. Fritschi, and T. Juenger. Weakly supervised minirhizotron image segmentation with mil-cam. In ECCV Workshops, 2020.
- [27] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Commun. ACM*, 27(3):236239, Mar. 1984.