IMPERIAL COLLEGE LONDON

REPORT OF PROJECT 3

RootSkel – A software tool to measure curved plant root tips

Author:
Felicia BURTSCHER

Supervisor:
Dr Giovanni SENA

A thesis submitted in fulfillment of the requirements for the degree of MSc Bioinformatics and Theoretical Systems Biology

in the

Department of Life Sciences

August 22, 2018

Word count: INSERT #WORDS HERE

"If you thought that science was certain – well, that is just an error on your part."

Richard Feynman

IMPERIAL COLLEGE LONDON

Abstract

Department of Life Sciences

MSc Bioinformatics and Theoretical Systems Biology

***RootSkel* – A software tool to measure curved plant root tips**

by Felicia BURTSCHER

Plant morphology studies the form and structure of plants and how these develop over time under different circumstances.
One phenomenon that, despite being first identified more than 100 years ago, has not been researched much and we know little about its underlying molecular mechanisms, is electrotropism. Electrotropism describes the response of a plant to an electric field; one common response is the curving of a plant's root tip. Unlike gravitropism [DO I HAVE TO EXPLAIN IT HERE?] there exist to our knowledge no tools specifically designed to assist biologists in studying this effect, most importantly the angle resulting from the curving of root tips. This is especially relevant as the experiment setup is more complex and the images tend to be more error-prone, which is often the reason why standard gravitropism study tools fail and biologists compute angle resulting from the curved root tip manually.

Here we present *RootSkel*, a novel and stand-alone software for image processing developed in MATLAB, whose pipeline was optimised for noise-intensive electrotropism images. Unlike when doing the angle computation to measure the curvature of a root tip manually, our tool ensures a standardised version of the angle computation. To make the tool more user-friendly, we developed a graphical user interface (GUI) that will help the user in processing the images and computing the angles in a standardised and controlled fashion.

Additionally we computed the angle of a an Arabidopsis root tip over one whole experiment (approximately 5 hours) and evaluated the results by comparing it to previously manually computed angles on the same images. The results show correspondence with manual angles; however, unlike the manually computed angles, using a standard definition of the angle, our software delivers angles free from human bias which makes the results comparable. Indeed, we show that on this example we can eliminate in average [INSERT VARIANCE HERE, STATE HOW MUCH HUMAN VARIANCE WE COULD ELIMINATE BY STANDARDISING IT] of human bias; further validation on more images is to be done. Previously computed angles can be checked by using our software, and more angles of curved roots can be computed in the future and hopefully reveal interesting insights in electrotropism. Moreover, our tool is not limited to roots but could theoretically be used on any curved object; slight modifications in the code and the GUI might be necessary.

Acknowledgements

This project was conducted under the supervision of Dr Giovanni Sena, whom I thank for his advice and guidance. Thanks also to Nick Oliver for their help and expertise from the biological side and other members from the Plant Morphogenesis Laboratory at Imperial College London, as well as Suhail A Islam for fixing technical issues and his incredible patience. Finally, thanks to Prof Michael PH Stumpf, Prof Michael Sternberg and others involved in conducting and overseeing the MSc in Bioinformatics and Theoretical Systems Biology at Imperial College London.

Contents

Abstract iii

Acknowledgements v

1 Introduction 1

1.1 Biological background 1

1.2 Literature review 2

1.3 Motivation 3

1.3.1 RootTrace 3

2 Methods 4

2.1 Data set 4

2.2 Experiment setup 4

2.3 Image processing 4

2.3.1 Digital images 5

2.3.2 On spatial resolution, gray levels and coloured images 5

2.3.3 Image processing operations 6

2.4 *Matlab* as a programming language 7

2.5 Pre-processing: Getting the root's skeleton 7

2.6 Computing the angle 7

2.6.1 Other ways of curvature and angle measuring 9

2.6.2 Computing the curvature of the root tip 9

3 Results 10

3.1 Key components 10

3.2 Workflow of skeletonisation of root 11

3.3 Graphical User Interface (GUI) 11

3.4 Validation: Comparing the automated calculated angles with the manually calculated angles 11

4 Discussion **12**

4.1 Challenges 12

4.1.1 Low contrast 12

4.1.2 Objects interferring with objet of interest 12

4.1.3 Skeletonisation 12

4.1.4 Scalability 13

4.1.5 Reproducability 13

4.1.6 Highly pixeled images 13

4.1.7 Problems 13

4.1.8 High user-interaction 13

4.2 Suggestions for future data acquisition 13

4.2.1 More focus on images 13

4.2.2 Resolution 13

4.3 Broader application of this tool 13

4.4 Further work 14

5 Conclusion **15**

List of Figures

List of Tables

List of Abbreviations

EF	**E**lectric **F**ield	
GUI	**G**raphical **U**ser **I**nterface	
ie	*latin* **i**d **e**st	that is

Chapter 1

Introduction

In the following chapter we will briefly familiarise the reader with the biological background of this work, summarise recent literature regarding the problem at hand to put our work into context and show in which way it contributed to the field.

1.1 Biological background

An important biological phenomenon studied by plant morphologists is *tropism* [ADD TO GLOSSARY], which is used to indicate the turning movement of a biological organism, here of a plant, when exposed to different environmental simuli [INSERT REFERENCE]. Usually the stimulus involved is added to the name, eg *phototropism* as a reaction to sunlight; it can be either *positive*, ie towards the stimulus, or *negative*, ie away from the stimulus. The most frequently observed and best studied tropism in *gravitropism*, which describes the process of how plants grow as a response to gravity. It was firstly scientifically documented by Charles Darwin [INSERT REFERENCE] and can be observed in higher and many lower plants as well as other organisms [INSERT REFERENCE, INSERT FIGURE DIFFERENT TROPISMS FROM G SLIDES]: Roots show *positive gravitropism*, ie they grow in the direction of the gravitational pull whereas stems grow in the opposite direction. An easy experiment to do is to lay a potted plant onto its side; over time the stem will begin to turn upwards and thus show negative gravitropism.
A far less studied process is *electrotropism* which describes the growth or movement of a plant when exposed to an electric field and which was the tropism under study in this project.

A high-overview explanation of the experiment setup and data collection to study electrotropism can be found in section [INSERT REFERENCE TO SECTION METHOD].

1.2 Literature review

The majority of traditional root development bioassays only consider a small number of points [REFEREENCE PERRY ET AL, 2001 AND MAIN ROOTTRACE PAPER]. They are informative in terms of long-term effects on root growth; however, small and temporary changes can not be captured [REFERENCE MAIN ROOTTRACE]. Recently developed tools have considered a higher number of time points which allows a better study of how the growth process develps and the plant's responses [REFERENCE MAIN ROOTTRACE; ISHIKAWA AND EVANS, 1997; VAN DER WEELE ER AL, 2003; CHAVARRIA-KRAUSER ET AL, 2007; MILLER ET AL, 2007].

Image sequences can contain a representative "snapshot" description of a plant's developmental stage; also they can be generated at high speed [REFERENCE TO EXPERIMENT SETUP IN CHAPTER ....]. From manual time-lapse photography [REFERENCE VAN DER LAAN, 1934; MICHENER, 1938] to measure lengths of seedlings after applying different external stimuli, digital camera technology has improved and low digital storage cost has become available which has made it comparatively easy to collect large, time-stamped digital image data sets monitoring root growth [REFERENCE MAIN ROOTTRACE]. Once the root growth changes have been extracted from the imge data, one can correlate their timing with the impact of different external signals including hormonal and environmental on processes such as cell division and cell expansion [REFERENCE MAIN ROOTTRACE]. This will help to understand root development in general.

Analysing image data manually, however, is time-consuming, very subjective and thus error-prone [REFERENCE MAIN ROOTTRACE]. When the analysis is done "by eye", it becomes difficult to reproduce measurements as they are not standardised by any automatised approach, ie made objective, and subject to human bias. Also, subtle phenotypes, ssuch as a delay in the response, might be missed [REFERENCE ROOTTRACE].

Different groups have developed tools for gravitropism and have shown the power of automatised image-analysis techniques compared to manual methods. An overview was created in figure [INSERT REFERENCE HERE].

1.3 Motivation

The work described here is motivated by mainly three factors: definition and standardisation or objectivity of the so far manually computed angle, flexibility and user-friendliness in the pre-processing step, and adaptability to standard consumer cameras. The software tool was designed to be used by a user, typically a biologist, with no need of specific knowledge in image processing nor plant morphology.

1.3.1 RootTrace

RootTrace [INSERT REFERENCE] has been developed to measure root lengths across time serie image data; biologists have also used it to measuer highly curved roots. A graphical user interface (GUI) implemented within the RootTrace framework makes it easy for the user to handle. However, this tool has failed on our image data set [INSERT REFERENCE HERE], probably due to the high noise level found in electrotropism images compared to gravitropism images.

Chapter 2

Methods

The purpose of this methods section is to give an overview on first the data that was used to develop the software tool, how the data were collected as well as image processing basics, and what programming languages, definitions and methods chosen when developing the tool. This will help the reader to replicate, understand and modify the methods and source code of the tool.

2.1 Data set

Our data set consisted of high-throughput time-lapse images of Arabidopsis roots taken by a standard Raspberry Pi V2 camera from 5 experiments with 5-6 roots each containing between 32 and 36 images over a period of approximately 5 hours.

It should be noted that these images had already been collected in a previous project and are not subject but only the basis of the work presented here.

However, to better understand the nature of the data, we will give a high-level explanation of the experiment setup that was used to collect the data.

2.2 Experiment setup

[SEE SLIDES, THESIS]

2.3 Image processing

Having these images as a basis, the nature of this work was mainly of image processing nature.

Image processing pools together a lot of different domains including physics (optics), signal processing and pattern recognition/ Machine Learning (ML) to ultimately feed computer to understand how do interpret images and make decisions based on them.

The idea behind image processing is to extract information from an image information in a form that is suitable for computer processing [INSERT REFERENCE HERE].

Figure [INSERT REFERENCE HERE] shows fundamental steps in image processing, that served as a guideline for the presented work.

[INSERT FLOWCHART FIGURE HERE, adapted from....]

2.3.1 Digital images

In image processing, we operate on digital (discrete) images. An image refers to a 2D light intesity function $f(x, y)$, where (x, y) denote spatial coordinates and the value of f at any point (x, y) is proportional to the brightness or gray levels of the image at that point [INSERT REFERENCE HERE]. A digital image is an image $f(x, y)$ that has been discretised both in spatial coordinates and brightness, ie we

- Sample the 2D space on a regular grid
- Quantise each sample, ie round to nearest integer.

What we get is an image represented as a matrix of integer values; the elements of such a digital array are called image elements or pixels.

[INSERT EXAMPLE PICTURE HERE]

2.3.2 On spatial resolution, gray levels and coloured images

The storage and preprocessing requirements increase rapidly with the spatial resolution and the number of gray levels. For instance, a 256 gray-scale image of size 256 *times* 256 occupies 64K bytes of memory. Figure [INSERT REFERENCE HERE] shows images of different spatial resolution.

[INSERT FIGURE: Images with decreasing spatial resolution, taken from .....]

Also, an insufficient number of gray levels in smooth areas of a digital image may result in false contouring, see figure [INSERT REFERENCE HERE].

[INSERT FIGURE: Images of different gray-level quantisation]

Since we deal with coloured images, we chose the RGB colour model which is an additative colour model in which red, green and blue light are added together in various ways to reproduce a broad spectrum of colours [INSERT REFERENCE HERE].

2.3.3 Image processing operations

An image processing operation typically defines a new image g in terms of an existing image f.

- Transform the range of f

$$g(x, y) = t(f(x, y))$$

- Transform the domain of f

$$g(x, y) = f(t_x(x, y), t_y(x, y))$$

IMAGE PROCESSING OPERATIONS USED IN THIS WORK INCLUDE:

A point operation are functions that are performed on each single pixel of an image, independent of all the other pixels in that image [INSERT REFERENCE HERE]. These include operations like inversing, changing the brightness or contrast of an image, changing the gamma of an image, binarising an image and logical operations.

"Local operations are functions that calculate the new value of each single pixel with regard to an area in its surrounding, called its neighbourhood [INSERT REFERENCE HERE]. They can be divided in two groups, linear and nonlinear filters, where a filter is an operation, which converts a source image via some kind of transformation to a result image. Another possibility for the subdivision of local operations is the division in low-pass and high-pass filters. Low-pass filters are used to inhibit noise and small details on images, whereas high-pass filters are used to emphasize edges."

2.4 *Matlab* as a programming language

As a language we chose *Matlab* as it is very popular in academic circles for image/ data processing given the number of built-in functions, including well documented image processing toolbox (*MATLAB Image Processing Toolbox*). Alternative languages that were considered were *Python* and *Julia*. Another recommended language is *OpenCV* as it is very fast and well documented. Other non-open source software such as *ImageJ*, a Java based image processing program, and *Avizo* which is a general-purpose commercial software application for scientific and industrial data visualisation and analysis with a nice GUI, could not be investigated further in this work.

2.5 Pre-processing: Getting the root's skeleton

The goal of the preprocessing step was to extract the skeleton, ie the coordinates of the pixels that represent the root on the image. This task turned out to be highly challenging on noisy image data and can be regarded as the actual achievement of the presented work. The highly elaborate pipeline of the preprocessing step can be found in chapter [INSERT REFERENCE HERE].

For the The initial problem of coming up with a "good" definition by comparing different angle and curvature definition shifted. The focus was now on the preprocessing, the extracting the skeleton. However, we will still present some possible definitions of angles and curvature, also not implemented in the current version of the tool.

2.6 Computing the angle

Once the skeleton has been extracted, one can compute the curvature and the angle of the root tip.

Various definitions of angles associated to the root tip had been considered; however, in order to make the angles comparable to the so far manually computed angles we chose an emulation of the manual computation of the angle in our final implementation.

Approaches like computing the Gaussian mean curvature, definitions of different, possibly more robust methods of computing and angle have not been incorporated in our final tool.

The angle Θ we are after, is the angle between the line through the point of the highest local (Gaussian) curvature and the root tip and the horizontal line parallel to the electric field (EF).

Figure [INSERT REFERENCE HERE] illustrates the angle Θ that is computed.

[INSERT FIGURE OF DRAWING WHICH ANGLE IS COMPUTED HERE]

Once these two lines are correctly defined, provided the point of highest local curvature is unique, it is straight-forward to compute angle *Theta*:

There are other ways such as finding vectors on the lines and using their dot products. However, we can use simple, basic trigonometric methods as shown in figure [INSERT REFERENCE HERE].

We would then compute the angles of the two lines to the x-axis in radians mode by

$$\pi - |\tan^{-1}(\frac{x(2)-x(1)}{y(2)-y(1)}) - \tan^{-1}(\frac{z(2)-z(1)}{y(2)-y(1)})|$$

or

$$180^\circ - |\tan^{-1}(\frac{x(2)-x(1)}{y(2)-y(1)}) - \tan^{-1}(\frac{z(2)-z(1)}{y(2)-y(1)})|$$

if we wanted it in degrees mode.

The single steps are shown in figure .... [PSEUDOCODE]

1. Find the slope of each line.
2. Find the inclination of each line using $\alpha = \tan^{-1} m$, where α is the angle of inclination, m is the slope.
3. Take the difference of these two angles.
4. Handle the case where this difference is not an acute, ie less than 90 degrees, angle. If we get a negative angle, we take its absolute value. We want to find an acute angle, so if we calculated an obtuse angle, oe greater than 90 degrees, we just subtract the value from pi radians or 180 degrees to get the acute values.

We refer to figure [INSERT REFERENCE HERE] to how the angle was computed in previous approaches.

2.6.1 Other ways of curvature and angle measuring

What was implemented as we found that this approach worked best on these data and this resolution.

2.6.2 Computing the curvature of the root tip

In order to detect the point of highest local (Gaussian) curvature, we need to compute the curvature at each point in the skeleton.

Chapter 3

Results

PROVIDE A HIGH-LEVEL REVIEW, NOT LOTS OF NUMBERS

REPRODUCABILITY OF WORK, ie make sure reader could redo it

The following section of this report will briefly explain the tool from a technical perspective; . We will present some feature highlights of the tool and we will guide the user through one example to illustrate how the tool is used in practice.

Additionally, we will show some validation of the tool by comparing the angle computed by our tool to the one computed manually on one time-series image data set of one Arabidopsis root.

The code is open-source and publicly available on [INSERT GITHUB REFERENCE HERE]; all previous versions including log files can be found on [INSERT GITHUB REFERENCE HERE]. [EMPHASISE THIS: MAKE RESEARCH TRANSPARENT]

3.1 Key components

Figure [INSERT REFERENCE HERE: PIPELINE FROM GUI] explains the key components of this image-analysis software tool to address the problem of highly noisy electrotropism consumer camera images of Arabidopsis roots and a standardised way of computing the angle for the curved root tip.

This tool takes the form of a MATLAB program with a graphical user interface (GUI).

3.2 Workflow of skeletonisation of root

The workflow developed via many iterations on different images and

elaborate pre-prococessing tool for skeletonisation

high functionality, reiterated process

Generic image processing workflow:

However, every single case is different.

Skeletonisation

fill in holes

[INSERT FLOWCHART WITH SINGLE STEPS]

Two approaches: 1. Converted into gray channel images. 2.

3.3 Graphical User Interface (GUI)

To make the program more user-friendly, we developed a graphical user interface (GUI).

- User interaction
- Error messages if user does not enter right values.

add error messages
INCLUDE FIGURES HERE.

3.4 Validation: Comparing the automated calculated angles with the manually calculated angles

Chapter 4

Discussion

PUT RESULTS IN BROADER PICTURE

4.1 Challenges

4.1.1 Low contrast

There is a very low contrast between the background and the roots, so that one could hardly recognise the roots on some images [INSERT EXAMPLE IMAGE HERE]. Inverting the image was not enough. We intensified the contrast by ... / making the background darker, we increased the brightness. We used further filtering so that the difference between two pixels is more pronounced. It was very much a trial & error process.

Classify only things without structure as noise. Noise was subtle/ hiding and could only be seen on filtered images.

Many objects of no interest which can be not

[INSERT 4 EXAMPLES HOW DIFFICULT DATA WAS]

4.1.2 Objects interferring with objet of interest

Eg specs not so much of an issue as separate from the root. Issue when it is connected.

4.1.3 Skeletonisation

Get rid of loop.

How to go along the curve?

4.1.4 Scalability

A lot of paramter tuning to single images. Work with single images to tune.

4.1.5 Reproducability

angle computation yes, but still variability in pre-processing step

4.1.6 Highly pixeled images

Due to compressing? Challenge in curvature computing part Eg using Hough transform from Matlab Image Processing Toolbox to detect lines in an image not applicable here as highly pixled image.

4.1.7 Problems

A lot of trial & error / hand-picking.

4.1.8 High user-interaction

Might be reduced with better-quality images, however very flexible.

standard-deviation not as high as i think?

4.2 Suggestions for future data acquisition

4.2.1 More focus on images

4.2.2 Resolution

Better camera, less waste of resolution

improve spatial resolution, other format? no compression?

4.3 Broader application of this tool

This tool can be reused for many purposes, it is not restricted to root detection. Might also be used for easier problems like gravitropism.

4.4 Further work

Here in the first version of the tool, the main goal was to standardise the angle computation; if in the future a method for handling the different noise pattern in the images was efficiently handled which require less user input, it would be desireable to automate the whole angle computation.

Use on better-quality images in the future.

Chapter 5

Conclusion

WHAT, WHY, WHY ADVANTAGE, WHAT WAS CHALLENGE?

Here we present a novel and stand-alone image-analysis-based software tool developed in MATLAB optimised for noise-intensive electrotropism images that is able to compute the curvature and angle at the root tip in a standardised fashion with a user-friendly and very flexible pre-processing step to extract the skeleton of the root from possibly very noise image data sets. It offers the possibility to extend the analysis by including different angle definitions to capture the curvature of the plant root and compare them. Also, the software tool is not limited to compute the angle of root tips but due to the flexibility of the pre-processing step can be used for any other curved or polynomial-like structure.

The software has been designed using an extensive amount of different filtering techniques optimised on the image data set described in this work [INCLUDE REFERENCE TO SECTION] and can therefore be used to work with standard images fro costumer digital cameras.

Automated image capturing as well as the design of the software presented here both aim to reduce the time-consuming process of the biologist quantifying the root tip curvature manually but more importantly, it standardises the computations and makes the results reproducible and comparable over a large amount of data.

This will contribute to understand highly complex and poorly-understood phenomenons like eletrotropism in plants and possibly other tropisms, as well as plant growth in general.

Hope to be able to standardise it in the future.