Augmented Reality in The Hounsfield Facility

Submitted Sep 2022, in partial fulfillment of
the conditions for the award of the degree **MSc Computer Science.**

XUHAO ZHOU
20349921

Supervised by Tony Pridmore

School of Computer Science
University of Nottingham

I hereby declare that this dissertation is all my own work, except as indicated in the text:

Signature XUHAO ZHOU

Date 9 / 9 / 2022

I hereby declare that I have all necessary rights and consents to publicly distribute this dissertation via the University of Nottingham's e-dissertation archive.

Public access to this dissertation is restricted until: 9/9/2022

Abstract

The X-ray computed tomography (CT) method is an imaging technique that allows researchers to reconstruct scanned objects in three dimensions and then study plant roots non-destructively[26, 47]. In experiments using X-rays to study plant roots, the pots and soil from which the roots were sampled blocked human vision, making it difficult to see the roots compared to plants in transparent gels. However, the results of such soil-based studies can be more accurately extrapolated to the outdoor environment.

Based on the sample situation and instrumentation at the Hounsfield Facility[12], we have designed an augmented reality (AR) application to compensate for the inability to view plant roots due to soil obscuring the view while assisting botanists in studying plant roots. By scanning a QR code marker on a sample, the app displays a model of the root system of the sample in question. It interacts with the plant segment by clicking on a partial part of the model to reveal further information about that root segment. Such a method is needed in experiments on a non-invasive investigation of three-dimensional root growth in soil.

Additionally, several common platforms for experimental plant research are summarized here, platforms based on paper, gel and hydroponics, and X-rays. These are analyzed, compared to derive possible needs, and extended to meet the relevant needs using this AR procedure. In addition, a converter is designed to read RSML[38] files to obtain 3D models of plants and root system information data.

Keywords: augmented reality, roots, Hounsfield Facility[12], X-ray Computed Tomography

Acknowledgements

I have received help from many people in the writing of this thesis.

I would first like to thank my supervisor, Tony Pridmore, whose expertise has been invaluable in formulating the research questions and methodology. Your insightful feedback pushed my thinking, and you guided me many times in the production of my project, making my work take shape with your help.

I would also like to thank my group members, who have given me invaluable guidance throughout my studies. You have given me knowledge on using tools in my project's concrete implementation and helped me complete my thesis.

Finally, I would like to thank my parents for their wise advice and sympathetic ears and for supporting me in coming to another country to study. You have always been there for me.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Food production is anticipated to need to increase by at least two times by 2050[5] in this case due to the population growth and rising food consumption, with the demand for rice already outpacing supplies between 2010 and 2012[14] and the amount of five different types of food harvested reaching 2.9 billion tones in 2014[54]. The environment is becoming more hostile under the impact of climate change, with increased risks such as droughts, floods, extreme temperatures, storms and high winds. All of these issues make it challenging to meet this target.

As a plant organ in direct contact with the soil, the root system provides stability and nutrients to the plant. The growth of the lateral roots may be related to specific nutrients in the soil[22], and the water content in the soil may also affect the growth of the primary plant roots[21]. In addition, reconstructing root structure from images has received a lot of attention (see Table 2.1). These techniques lead to picture analysis and a root structure representation. With the advent of methods, plant scientists need effective ways to access and examine this data.

The study of plant root phenotypes by botanists can further assist breeders in creating plants that are more resilient to environmental stresses, such as drought, disease, and insects[5]. Understanding phenotypes and identifying root features throughout plant growth can help farmers better water and fertilize plants for various field conditions[14]. However, choosing crops based on root system architecture (RSA) studies to fit various environmental circumstances could be challenging[5].

Here we have designed an augmented reality (AR) application to assist botanists in researching plant root systems. By scanning a QR code marker on a sample, the program can display a model of the root system of the associated sample and interact with the plant segment by clicking on the local section of the model to display further information on this root segment. Such approaches are needed in experiments related to the non-invasive investigation of three-dimensional root growth in soil[45]. In experiments using X-rays to monitor plant roots[22], the pots and soil in which the roots are sampled block the human view, making it difficult to see the roots compared to plants in transparent gels. However, the outcomes of such soil-based research can be more accurately extrapolated to outdoor circumstances[5].

Root System Markup Language (RSML)[38] was used to collect plant root structure information for the data input. It is an XML standard file that contains root routes for each root segment, information on root system parameters, metadata, and a root structure in each scene, which may be derived from a single or a series of 2D and 3D images. The methodology section presents a model generation tool to read RSML[38] files to generate a 3D structural model and corresponding information files, which can subsequently be used as model inputs for this AR application.

The desktop application, called RootsVis, can read RSML[38] files to display the model. It also can combine two models, a split view and a join view, and display them simultaneously in the middle of the software for comparison. This structure allows users to see the subordination between the primary and lateral roots. Each model can be viewed from a different angle by moving the mouse to change the angle of its display. These functions have been improved or added to during the development of AR software. Multiple image target recognition, for example, was inspired by this comparison function. In the methodology part, the intersections and extensions between this and the AR application are thoroughly detailed, and the conclusion section discusses potential future advances that will complement one another.

The cylinder that separates each pair of points is utilized to create the overall model as a component of the set; nevertheless, the large number of successive points will reduce the

effectiveness of the model building. The polyline tag in the RSML file will provide the root system coordinates to the 3D model. A Douglas-Peucke[33] simplification function is used to remove part of the straight points, and three different simplifications are evaluated side by side. The AR application will also display corresponding information simultaneously, such as image analysis data on the plant segments, which was also collected from the files. We then used a 'midpoint' model approach (with a focus point for each root segment and corresponding information described in the methodology) to trigger these data, relying on scripts in the Unity engine[23]. On the iPhone platform, the final AR application will be put to the test using AR Foundation in Unity.

Finally, to determine how the AR application's functionality is complementary and interoperable with the desktop application RootsVis, a desktop application for plant roots created in C++, the paper will also examine the viability of using this application in experiments, including invasive and non-invasive studies. There will also be a discussion about potential applications for this application in the Hounsfield Facility[12]. Moreover, feasible future upgrades are proposed, as well as solutions.

Chapter 2

Literature Review

The ability of a plant to detect its environment is often reflected in its organs. Research into deeper roots can assist plants in better exploring the soil since they aid in water absorption, promote crop fitness, and increase nutrient uptake[34]. The root system architecture (RSA)[53], which refers to the spatial distribution of all roots in a particular growth environment, is frequently impacted by environmental changes, including pH, moisture, temperature, and the dispersion of soil microbes. The study[53] also demonstrates a relationship between plant yield and root system characteristics[27, 30]. Studying such root characteristics would enable a better choice of ideal root features for diverse external situations, such as drought[66], pests, soil bacteria and others.

2.1 Plant Root Data

Experiments in studying plant roots are usually divided into two categories, destructive and non-destructive.

Among the strategies for growing plants mentioned in the paper[53], the "Filed method" is the usual destructive cultivation method, which has the advantage of relevance in terms of physiology and practice. Shovelomics[65] is typically associated with low throughput studies and is not appropriate for studying large genetic populations[64] because it involves digging plants out of the root crown and washing the soil out of the roots. On the other hand, the disadvantage is that it intensifies root cleansing and destructive detection, which

Table 2.1: The chart collates information on the four different experimental platforms, covering the articles cited, the software used to present or process the image data that appears in the paper, and the strengths and weaknesses of each experimental platform, as mentioned in the cited articles.

Root Cultivation Platform	Image analysis and data presentation software	Advantage	Disadvantage	Reference
paper-based growth system	GIARoots[15] WinRhizo[56] SmartRoot[37] GROWSCREEN-Root[50] ROOTMAP[10] SimRoot[40] SVM[7]	1. Suitable for quantitative research 2. Easy to handle, faster analysis controlled 3. Repeatable in limited space	1. Differences in colour background paper in lower accuracy 2. Not for crown roots 3. Root overlap	[61] [60] [32] [45] [17] [59]
Agar	PlaRoM[72] SmartRoot[37] NIES-Element 3.20[13]	1. Holds plant roots in place. 2. The translucency of the agar is suitable for photo generation.	1. Some defects (bubbles, scratches, development of pathogens)	[72] [28]
Hydroponic	SmartRoot[37] ImageJ plug-in[8]	1. Possible to studying a high number if individual over a long time period	1. Roots may change their position	[28]
X-ray or magnetic resonance imaging	RootTrak[42, 44, 43] NMRooting[69] SmartRoot[37] DIRT/3D[34] RootTrace[49] RootNav[57]	1. High precision 2. Resolving the more complex interrelationships between soil and root systems 3. Three-dimensional root growth	1. Limited throughput or require major investments for automation 2. The firmness of the soil and its 3. water content may affect the imaging results	[26] [24] [39] [20] [29] [69] [5] [34] [22]

means that the entire root system might be destroyed when the plant is dug up. It is an invasive method, and certain features risk being destroyed, making it unsuitable for ongoing studies.

Non-destructive, on the other hand, is a non-destructive method that leaves the plant root system intact. At the same time, it is often impossible to quickly carry out a non-destructive analysis of plant roots because of their hidden nature[32]. The article, therefore, proposes a paper-based growth system method, in which plant roots will be grown between two sheets of moist paper and evaluated and analysed through image analysis. In addition, various other phenotype platforms include hydroponics, aeroponics, agar, and gel. Non-invasive methods for detecting root growth in soil, such as X-rays, have advantages and disadvantages. Non-destructive phenotyping might take more time and require more labour if the measurement is done manually[58].

The plant data mentioned here may primarily relate to root traits (root length, mass, angle, diameter, number of tips and spatial distribution) [69].

2.1.1 Non-destructive experiments to obtain data for plant root studies

Paper-based Growth System

The basic design of this experiment[32, 45] is typically a multi-layered, two-dimensional structure with a central piece of plexiglass to stabilise the system. Germination paper

covers both sides to provide water and nutrients during development, and transparent aluminium foil covers prevent the roots from drying out.

The embryo grows in the space between the glass and the paper, and in this state, the plant roots grow significantly, and a fungicide is used to suppress fungal growth. Finally, using picture software analysis, plant root features were discovered[32].

Using hyperspectral microscopic imaging in the studies, the scientists evaluated the efficacy of several germination materials. Several image analysis techniques were used to evaluate their accuracy and root structure for various background plates and illumination.[32]

A similar experimental platform was used to investigate the effects of temperature[60], low water potential[61] due to polyethene glycol, and nutrients. Using "not fully opaque paper," as suggested in studies from Pejman[59], can significantly save expenses compared to agar gel experiments, which are automatically optimised in terms of contrast due to their limited light permeability.

The GrowScreen-Page (paper germination) mentioned the following three points[17]: Phenotypic research,180 oilseed rape variants were evaluated for their natural variety, phenotypic diversity, and plasticity. The time series of growing root systems, the root growth dynamics, and the development of 52 barley genotypes under two distinct nutritional situations were also studied. This experiment concentrated on a high-throughput experimental research strategy and revealed that most features varied significantly depending on how the genotypes were treated. The benefits of paper germination for experiments are emphasised.

Overall, this platform focuses on quantitative trait analysis based on the reproducibility of the experiments. The quick analysis of large batches and its advantages in terms of the width of the paper sheet, which enables the root system to expand indefinitely in three dimensions (but is determined by the size of the paper, which has a time limit of 21 days in GrowScreen-Page)[61]. The overlap of the root system on a two-dimensional plane may also affect the measurement of the experiment.[32].

Equations and image analysis graphs were used to illustrate and study specific samples of genotype, characteristics, and patterns[17, 61]. The data from image analysis, such as the

number of primary branch roots and branch angles, have poor form heritability[17, 59]. As a result, the actual sample scenario and information in this example are not adequately integrated. Similar colour-coded figures were employed for the barley plant temporal developmental control experiment[17].

Agar and Hydroponic

Hydroponics is the growing of plants in a nutrient-controlled liquid and has been used primarily in studies where nutrient control is required to study the plant root system[45]. Due to the entanglement of the plant with the liquid medium, quantifying such a system would not be possible[45]. Agar is similar to this immobilising the root of plant system compared to hydroponics while also controlling the external nutrients[72]. The disadvantage is that it is not designed for long periods, as pathogens may appear in the medium, which is not conducive to photographing[32].

Yazdanbakhsh and Fisahn[72] used agar as a research platform to record changes in plant root growth under different control conditions, using time as a sequencing basis. A 3D RSA imaging platform developed by Lyer-Pascuzzi[28] investigated the genetic composition of rice under different control traits based on gels and using cameras outside transparent growth tubs to take pictures of their root systems at different angles and record them continuously for 14 days, after which the pictures were analysed to extract different root characteristics. Experiments highlighting transparent experimental tubs, chronological order, and 3D model reconstruction. Laura and the team[45] have devised an innovative hydroponic rhizomes design.

In contrast to the paper-based platform, the roots are grown in two mesh-like fabrics that hold the root system in place while providing a way to measure the length of the root tip meristem. Again, this experiment examines the entire life cycle of the plant.

Hydroponic rhizotrons are used to examine the RSA of mature Arabidopsis plants. By adjusting different nutrients to control plant growth, the rhizopodia setup in Arabidopsis plants makes it possible to measure the root surface area, length, depth, width, and lateral root density[45].

In summary, the experimental platform based on hydroponics and agar focuses more on qualitative experiments. However, it can cope with small volumes of quantitative experiments due to the reduced price of experimental materials and the increased ability to process pictures[45]. Also, the whole plant growth cycle was used as the basis for several experiments[28, 72], so the time and plant root models may also be a need.

X-ray or Magnetic Resonance Imaging

The X-ray computed tomography (CT) method is an imaging technique that allows researchers to reconstruct scanned objects in three dimensions and then study plant roots non-destructively and non-invasively[26]. This method was employed by Hainsworth and Aylmore[24] to investigate the geographical distribution of water content in plant soils. Lontoc-Roy[39] examined the development of maise roots in various strata under dry and wet circumstances and acquired the valve for several samples. As mentioned in the paper[47], studying the variation in plant root size at depth is also an aspect of this research, and the use of various root washing devices can impact the measurement of root mass[20, 70]. Moreover, when the number of samples increases, so does the work and time required to study and test the plants.[58]

Other techniques for extracting 3D RSA from soil include positron emission tomography (PET) and magnetic resonance imaging (MRI). While PTE is currently restricted to coarse resolution[29], MRI employs radiofrequency waves to activate the hydrogen in water for imaging[69]. MRI may be utilised for structural or functional imaging of plants, such as the distribution of water in plants[5]. Also, CT is biased toward studying small pots (pots less than 34 mm in diameter), whereas, for larger diameters (larger than 81 mm), MRI can detect more of the root system than CT[46]. Whereas PTE performs better in soils with high water content, its scan time is usually greater than MRI and CT, so it is not easy to carry out analyses for large-scale genetics[29, 46].

Therefore, it is reasonable and appropriate to use CT for the situation in Hounsfield Facility[12]. In order to view the root system of each sample in detail, it is possible to view the root system hidden under the soil using a more convenient mobile device.

2.1.2 Analysis and study of Plant root traits

Paper-based Growth System

In the tests of[32], the team explored the literature. It analysed several intelligent 2D root image software for studying complex root RSA before settling on three image analysis programs and evaluating them. Aspects of the data, such as lateral roots, total root length, the number of lateral roots, and inter-root angles, were extracted from the image and analysed in the experiments. Both GIARoots[15] and WinRhizo[56] tools can segment photos, separate roots from background plates, and convert them to binary images. After manually or automatically modifying thresholds, the results are displayed for locating weaker lateral roots. SmartRoot[37] can analyse many photos and find correlations between them. The accuracy of the image analysis process is influenced by several parameters, including the degree of detail in the image analysis, contrast, texture noise, and the efficiency of the process.

The program[17] for image analysis in high-throughput trials, GROWSCREEN-Root[50], was also used to measure the size, depth, and geographic distribution of the root system, as well as the lengths of the primary root, total length, and the number of branches, as well as the branching angle. The paper also notes that a variety of root features identified by its phenotyping platform may be applied to simulation root software, such as ROOTMAP[10] and SimRoot[40], to enhance research on resource usage efficiency.

Agar and Hydroponic

An experimental platform based on agar[72] was set up to record changes in plant root growth in different controlled environments based on time sequencing and to show high-resolution images with individual root extension curves by PlaRoM[72].

The length of root apical meristematic tissue was analysed in the experiments[45] using NIES-Element 3.20[13], images of roots and shoots were analysed using the ImageJ plug-in, and due to its experimental setup, the images were easily obtained with optimal contrast. Finally, the length of the non-meristematic zone and lateral root density were calculated using SmartRoot to analyse selected roots in the images.

Lyer-Pascuzzi and team[28] used SVM[7], a supervised learning method to rank RSA traits for analysis of top-ranked feature diversity, indicating the breadth of variation in the selected rice. The experiments were analysed and ranked according to to plant root depth, mean root radius and surface area, and finally presented as a heat map.

X-ray or Magnetic Resonance Imaging

In terms of extracting data from 3D models, the software SmartRoot[37] is mentioned in the article[47] as a method developed to analyse root structure from scanned images of plant roots. Its software allows comparison of changes in the structure of this plant at different points in time through a vector representation of the root structure but was not suitable at the time to handle high throughput analysis. In addition, the Centre for Integrative Plant Biology at the University of Nottingham developed the software RooTank[44, 43], which reconstructs a three-dimensional RSA using the different X-ray attenuation between the soil and the root system. Subsequent upgrades to the program have allowed for the separation of interdependent root systems[42].

Van Dusschoten and team[69] used the image analysis software NMRooting[69] to extract plant root traits from a 3D model in aggregate. Through image data, the DIRT/3D plant 3D root phenotyping system creates a coloured 3D point cloud model and calculates and extracts the associated plant root characteristics[34]. Also mentioned in the article[47], some tools, RootTrace[49] and RootNav[57], can connect pixels into a coherent graph for showing the more complex topology of plant root systems. Furthermore, deep learning has also been applied to solve root-soil style problems[47].

In the experiments, the plants were tomography to obtain a section image of each root system, and from each section image, the soil and the plant root system were segmented to extract the plant root system. While studying experiments on plant root entanglement, Stefan and his team attempted to use the volumetric segmentation data generated by the above method when recovering plant roots from CT images. They obtained root cross-sections by 3D Manhattan and other methods. The results of the 3D structural data are represented in RSML[38] format for their nested root structures.

2.2 Advantages of using the Root System Markup Language to store plant root information

The Root System Markup Language, also known as RSML, was proposed in 2015 to address the problem of creating a standard format for extracting static, dynamic features between different types of images and species[38].

The plant and root attributes and geometry, continuous functions along individual root routes, and a sequence of comments at the image, plant, or root scale at one or more points in time are all stored in the RSML[38] format. This format may hold 2D or 3D image information. This format also uses a polyline segment as the primary collection structure to capture the shape of the collection in the root system structure. The polyline segment contains an ordered list of points, each of which has points with coordinates on the x and y axes; the z points are used to create the three dimensions[38].

In the section comparing experimental platforms above, it was also mentioned that information on the coordinates of the root system, and its root system characteristics were obtained by pixel analysis (via software) of photographs of the root system. Thus, as a basis, multiple 2D scans are obtained by tomographic scanning of the plant root system, which is then analysed and reconstructed by software. Finally, the 3D data is stored in RSML[38] format. However, the possibility of directly acquiring the 3D collection data is also mentioned in the article[38].

In terms of practical applications, archiDART[9] is proposed for processing RSML generated from the DART[18] platform and calculating global RSA features, root growth rates, and direction parameters, in addition to its mapping function feature, designed to visualise the dynamics of root growth. The updated version has added a topological analysis method to analyse plant root systems by comparing persistent homologous data. In addition, it is possible to analyse time series from RSML[38] files and share descriptions of 3D and 2D root system structures.

In experiments[36] to improve the accuracy of the root image analysis pipeline through a systematic model, image data is stored with an RSML[38] file and read through the

ImageJ RSML Reader plugin[8] to generate an image file and morphological parameters in a way that demonstrates the generation of relevant image datasets for training.
Stefan and his team proposed the RootForce[16] method for segmenting large-scale volume data from Computed X-ray tomography (CTX) and allowing precise modelling of root systems as small as a few microns in diameter in pots of different diameters. The software includes a section to calculate the root system structure in RSML[35], while the segmented root system can be saved as an RSML[35] file.
In summary, different levels of root structural complexity can be accommodated by RSML[38] (from the 2D prediction of a single root system to the 3D representation of the complete root system)[38]. These data can also be used as marker points for studying models using the timeline RN52, RN7. RSML[38] has the advantage that it can be shared between different software teams such as ImageJ[8] SmartRoot[37] and other image analysis software, and there is also data analysis software to analyse this data format. It also defines a standard that can be easily reused and shared across different teams.

2.3 Augmented Reality Framework

According to a July 2020 assessment on the hype of emerging technologies, the position of AI-Augmented development is at the highest predicted peak. The usage of AR and VR in research and other sectors is gaining in popularity[62]. There is also much interest in these technologies, such as the dynamic mathematics-based GeoGebra system's mobile app[55]. In order to imitate space objects, use universe sandbox2[48].

2.3.1 Examples of AR tool Applications in Botany

They[71] developed a plant management system that uses mobile AR devices to search, compare, and identify plants side by side. The user can inspect the specifications of the selected outcome by holding a marked voucher and moving the head horizontally and vertically to select the sample item to be shown.
Sean[71] and his team have designed a head-mounted plant identification AR for use in

the field to provide comparative reference of plant leaf images. This experiment demonstrates the possibilities of using augmented reality technology with head-mounted displays and the feasibility of using marker paper (Vouchers) sheets simultaneously to differentiate between different plants. Furthermore, the paper contributes to the design of the program interface and provides a framework for integrating botany and augmented reality technologies.

Zheng and his team have designed an AR program, citeRN60, that combines AR land data visualisation with field navigation to help farmers make the right crop planting decisions based on soil sample analysis and sensor information. The experiment was conducted using an AR headset and required GPS data, a chip built into the headset and a voice and gaze interaction system to meet the operating conditions. In this experiment, soil data, such as pH and humidity, is used. It can be stored in a local database along with current GPS information to reach a set of coordinates for a location. This experiment provides the basis for applying AR in combination with a database, GPS and sensors as a data acquisition aspect.

2.3.2 Augmented Reality Tools

ARKIT and ARCore

Apple-specific ARKit[3, 52] is an AR development tool. Its traits include environment analysis, real-world vertical and horizontal plane detection, image tracking, and the ability to superimpose virtual information on an image. ARKit can aggregate the brightness level and apply it to virtual objects by utilising the brightness sensor[51].

ARKit[3] does this for both vertical and horizontal plane identification using points taken from and monitored in the scene. These points might be edges of things, corners, lines, or differences in colour or form. ARKit[3] continually analyses the scene, and when it identifies a bigger or smaller plane, it changes the value of the rectangle according to the size of the identified plane. Each plane gives the predicted position and shape of the surface, allowing the virtual item to be placed accurately in the scene[52].

In image tracking, similar to scene feature detection, ARKit[3] creates several feature

points that analyse the features of the images in the database. These points serve as the characteristics of the discovered image in the real-time image collected by the camera during image detection.[52].

With the help of the size of the horizontal surface and location as well as the lighting sensor, ARCore[1]can follow the phone's location in its surroundings and allow the device to detect natural light. While ARCore[1] is available for Android and uses Java or Kotlin, ARKit, the framework for developing augmented reality for iOS, employs Swift or Objective-C.[11]

Vuforia

Vuforia SDK is a popular AR tool that works with most smartphones, tablets, and digital eyewear[62]. Additionally, it is very tightly connected with many platforms. With Unity[23]'s enhanced graphical interface, developers can quickly build AR applications on Android and iOS devices, streamlining the coding process.

One characteristic is the ability to detect many items, such as images, cylinders, and solid objects. Many visual objects are gathered into a regular geometric shape, such as a box or a plane, with multiple objects inserted randomly, to build an animation model that correlates to the matching detection when there are many things. VuMarks, customised Vuforia-type markers, are used in augmented reality applications to give distinctive identification and tracking. They work similarly to QR codes.

The tracked image target's name and target type may be established in its particular device database, and feature analysis can be applied to match the ensuing detection requirements[2].

ARFoundation

ARCore and ARKit are similar regarding real-world location, level detection, illumination estimate, image, and face identification of motion tracking sensors[51]. Additionally, the AR foundation for Unity[23] has integrated the two features into a single module and made it available for IOS and Android users[51]. The AR Foundation library combines

and expands upon the native libraries of two prior platforms to offer a new layer and a shared programming interface that can be used to launch applications on both platforms with only one programming language[11].

AR Tracked Image Manager[67] is used for tracking and creating GameObjects for each image found in the environment. The manager needs to see a set of reference images programmed into the reference image library before the image detection will be valid, and only the images in the library will be detected. Using makers from the reference image library, the image tracking subsystem will be utilised to find and track 2D images in the surrounding area.

The reference image library[4, 68](Figure 2.1), which keeps a set of reference markers[4], makes it possible to manage the complete collection of detected photos exceptionally effectively. Additionally, it offers the image tracking subsystem a library of reference images so that it knows what to look for while searching[67]. The name of each reference image is established to ensure no conflicts between duplicate names; yet, this name aids in identifying which reference image has been discovered, and the subsystem will not use it. Unlike Vuforia, which handles the relationship between individual marker images and the 3D model separately[2].

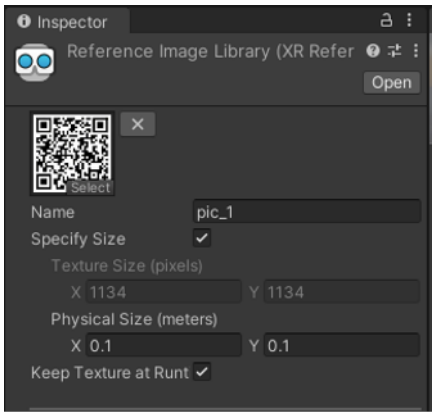

Figure 2.1: Reference image library

2.3.3 The Comparison Between AR Foundation and Vuforia

The SDK (Software Development Kit) of AR Foundation[31] and Vuforia[2] are imported into Unity[23], a more commonly used platform for creating AR and VR[11], in order to

create applications and satisfy the demands of experimentation. ARCore[1] and ARkit[3] are not compared here since the native libraries are already included in the two extension packages, which may publish applications to run on their respective mobile platforms.

According to the Mircea-NSMuseum project paper from Ioan[11], Vuforia has a more significant advantage over AR Foundation in terms of compatibility with older devices because it can work with at least IOS 9.0 and Android 6.0 platforms, check to see if those platforms support ARKit and ARCore, and then take advantage of those features. AR Foundation, on the other hand, only supports IOS 11.0 or Android 7.0.

The report also notes that both devices perform well for image tracking, but what stands out is that smartphones employing Vuforia have a 'Vuforia' watermark that is very hard to get rid of because the framework makes use of the SDK.

Significant design and development variances exist as well. Vuforia emphasises more straightforward integration and does not utilise a single line of code to accomplish the feature. However, multiple image tracking will be problematic. A prefab 3D object must be assigned to each marker set in the structure tree of Unity[23] for the matching 3D item to be displayed. The maximum limit for this list would be 100 instances, which would be a time-consuming process if designed in this set manner.

We can get a 3D model in AR Foundation by importing a single prefab or prefab collection. The code dynamically obtains the filename and image name to link the two. AR Tracked Image Manager can identify any maker in its collection, and the Reference Image Library component is required for this operation. However, the image tracking object in Vuforia needs to be configured into the scene one by one[11]. This way, the code can be used more efficiently to import, analyse and match 3D models to markers.

Chapter 3

Methodology

3.1 Reasons for Using AR

The emphasis is on a closer examination of one sample or comparing two samples because of the unique characteristics of the AR method. Therefore, it must be demonstrated in a different experimental setting, such as a high-throughput experiment[17] with a particular sample or a quantitative experiment with a single sample. In this method, the data information may be visualised differently.

3.1.1 Programming Requirements in Different Experimental Situations

The paper-based platform is geared towards high throughput studies[17], where large amounts of data are analysed to derive root characteristics, whereas hydroponics[45] and agar[72] culture are more qualitative experiments but can also handle partially quantitative experiments[45]. Two main types of experimental needs may be derived based on the above experiments.

- Detailed viewing and comparison of specific samples.
- Presentation and comparison of data in conjunction with physical objects.

Firstly, in the high-throughput experiments mentioned above[47], a large number of

RSML[38] files with different segments were recorded, different 'special cases'[17, 61] may arise, and AR can provide botanists with different perspectives on how to study roots. There are combining AR programs with physical comparisons (inspired by the model comparisons of the desktop software RootsVis). The data from the analyses acquired in the experiments indicated above were presented in various graphical methods to analyse root features, and the form of showing the root structure data on 3D root models may be a novel approach to present them.

The need for this is evident in experiments such as X-ray or magnetic resonance imaging because most of the sample roots are hidden under the soil. It is impossible to observe the complete root system with the naked eye, whereas with such instruments, more complete data of the root structure can be obtained and therefore combined with AR, can help botanists.

As the design is based on the Hounsfield Facility[12] scenario, each object is in a pot, and there is a marker on each pot, which is used to position the corresponding plant root system. The bar-code label on each sample unit is also used for collection and identification, so in combination with this bar-code, the generated AR model can be similarly overlaid on the original plant samples for comparative studies to compensate for the inconvenience of observation due to camera contrast or other reasons and to help the researcher.

In summary, the two types of experimental needs mentioned in the above paragraphs can be met or remedied by borrowing AR to help the user, while studying the plant root system in a different light.

3.1.2 Programming Requirements in the Root Data Format

Based on the standard root data collection format, RSML[38] can collect structural data on different root systems, which is a problem for generating plant models with this accuracy. However, this accuracy problem can be solved because of the commonality of the format in different software, the advantages of each software, and the upgrading of the testing instruments. This recording format will be employed since RSML[38]offers to

store data that satisfies this need.

The AR program combines the root trait data (obtained through various image analysis software) with the root data from the RSML[38] file, combining these two and presenting them. Botanists can get different perspectives by examining phenotype changes in the root system in several experimental situations using augmented reality applications.

Aspects such as the study of the development of plant roots over time, as mentioned above, are one of these needs and can be viewed and compared in detail as only 2d images of one plant root are analysed at a time. By analysing the exact image, converting the acquired data into RSML[38] format, and converting it into a model in chronological order, the 2D or 3D root growth of a particular sample can be observed more visually.

3.2 The AR Programming Process

The AR application we created will instantly display a model of the corresponding plant root system after scanning the QR code on the pot. As the phone is rotated, the perspective of the root model will also adjust.

3.2.1 AR Tool Selection

Based on the comparison of the different platforms in chapter 2.4.3 and the functions that may be required for the project, we will use AR Foundation[67] as the framework for Unity[23] development[67]. There are several reasons for this tool selection.

- Multi-target detection.
- Extensions to functionality using code.
- Implement ability for the IOS platform.

The first reason is that there will be problems with multi-item identification as we will use the mobile terminal to scan markers on several plant samples to create matching root models. As we have already explained in chapter 2.4.3, the setup in Vuforia is tedious and time-consuming[11].

First, there will be issues with multi-item identification since we will be utilising the mobile terminal to scan markers on various plant samples to generate models of matching plant roots. The setup in Vuforia is laborious and time-consuming, as mentioned in chapter 2.4.3.

Direct code implementation of the multi-object feature is possible in AR Foundation[67]. The Reference Image Library component[68] and the AR Tracking Image Manager[67], both of which can identify any creator in their collection, are needed for this process. Since scanning several QR codes yields the same model in the default mode, which only contains one 3D model corresponding to many photos, this mode needs to be expanded using code.

The program is designed to associate markers with the filenames of the 3D models, using the same names as the basis for displaying the corresponding models. This feature is done based on the inheritance of the two primary components. Each entry in the list of models represented by the placeable prefabs in the diagram below (Figure 3.1) corresponds to a model of a plant root; further information about the code will be provided in chapter 3.5.

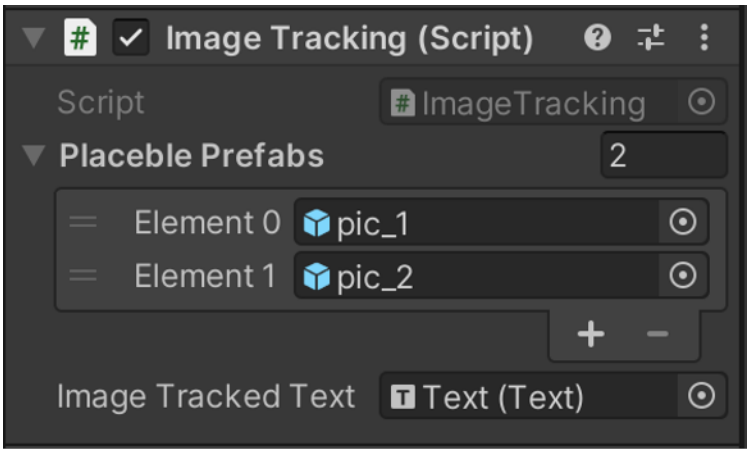

Figure 3.1: Place prefabs Store the same root system model as the picture name

The last reason is that distinct outcomes were seen when the two frameworks, AR Foundation[67] and Vuforia, were applied to the iPhone platform. The "Vuforia" watermark is displayed in the bottom left corner of the screen, indicating that the SDK is operating correctly, and the required content is not properly displayed when installed on the iPhone using Unity[23]. This problem is likely because of ARKit[3] or other settings,

as the Vuforia does not run correctly when these settings are present. After trying to apply AR Foundation[67], it works correctly and displays the required content.

3.2.2 How the Displayed Physical Model is Obtained

The RSML[38] file will be used as input to create the model needed for the AR application. Here, a data model converter will be created using PyTorch 1.9.1 and pytorch3d 0.6.2 to construct the matching plant root model utilising the data from the RSML[38] file.
The advantage of utilising the RSML[38] file is that it is the more typical file format used in plant root investigations, with basic plant name information and structural coordinates included in the file. The function of the program is to extract this information, either as coordinates of the plant root structure for the creation of obj files[19] or as detailed data on the root system in the form of CSV[63], excel, and graphs.
Python is used because it has a wide range of development tools that can combine model meshes and extract data from XML files[6], which is appropriate. Since the format of RSML[38] files is similar to XML(extensible makeup language)[6], it is feasible to use these packages. Furthermore, there are not too many instances of this tool built in Python; instead, most people utilise data generating tools and other languages to work with it, so it makes sense to use this approach.

3.3 Tool: Model Generator

3.3.1 RSML[38] and Model Generator

As shown in the lower part of the figure (Figure 3.2), the functions in the model cylinder file obtain the coordinate points of the upper and lower base of a cylinder, and by combining these coordinate points, a model of the cylinder can be obtained. The coordinates of the root system in the RSML[38] file are continuous, so entering two consecutive points will give a root system model on this segment.
The model-cylinder has three input parameters, the centre coordinates of the start and end (every two consecutive points in the RSML[38] file[38]) and the radius of the column

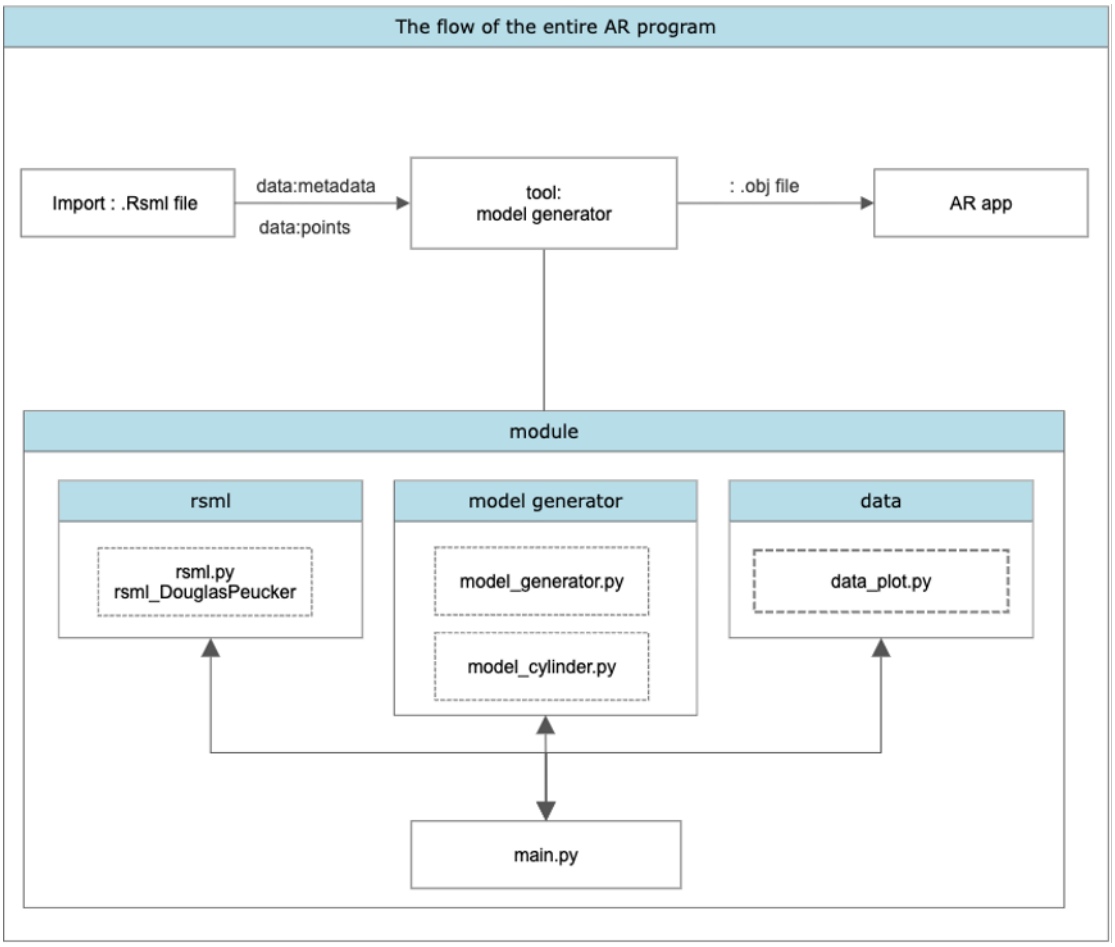

Figure 3.2: The flow of the entire AR program: The picture above shows the whole process of processing the RSML[38] file, from the input of the RSML[38] file to the model converter in which the root system information data (metadata) and the structural coordinates (points) are imported into the AR program. Below the image are the sub-components included in the model generator.

(fixed at three units in this experiment) in the central part of the model generator shown above. The model generator is used to create the assembly obj file[19], which is essentially a mesh file of a 3D model with different combinations of points and faces, and in this case, it can be understood as a 3D model file with point and face data.

Moreover, here we use the time complexity to evaluate the algorithm's efficiency, which is essential after counting the number of executions per code, so its size is usually related to the amount of data input. Here the time complexity of the entire function to read the RSML[38] file is O(MN), with M being the number of sub-root segments in the total root system. N is the number of consecutive coordinates in each sub-segment, and O indicates that the algorithm's speed converges towards the worst case of this M times N.

After extracting the data from the RSML[38] file, each coordinate point is stored as a coordinate structure in an ordered list. Each coordinate structure has sequential parameter to record the order in which each coordinate point was extracted. This list will later be used as input for the parameters of the simplified equations. This is because some of the coordinates will be rounded off during the simplification process so that the resulting simplified list will be "misaligned."(Figure 3.3) This misalignment has been found in subsequent tests to cause errors in the model generation (discontinuity of the model, misalignment of the generated cylindrical model)(Figure 3.3), so this has been added in subsequent upgrades to correct the errors.

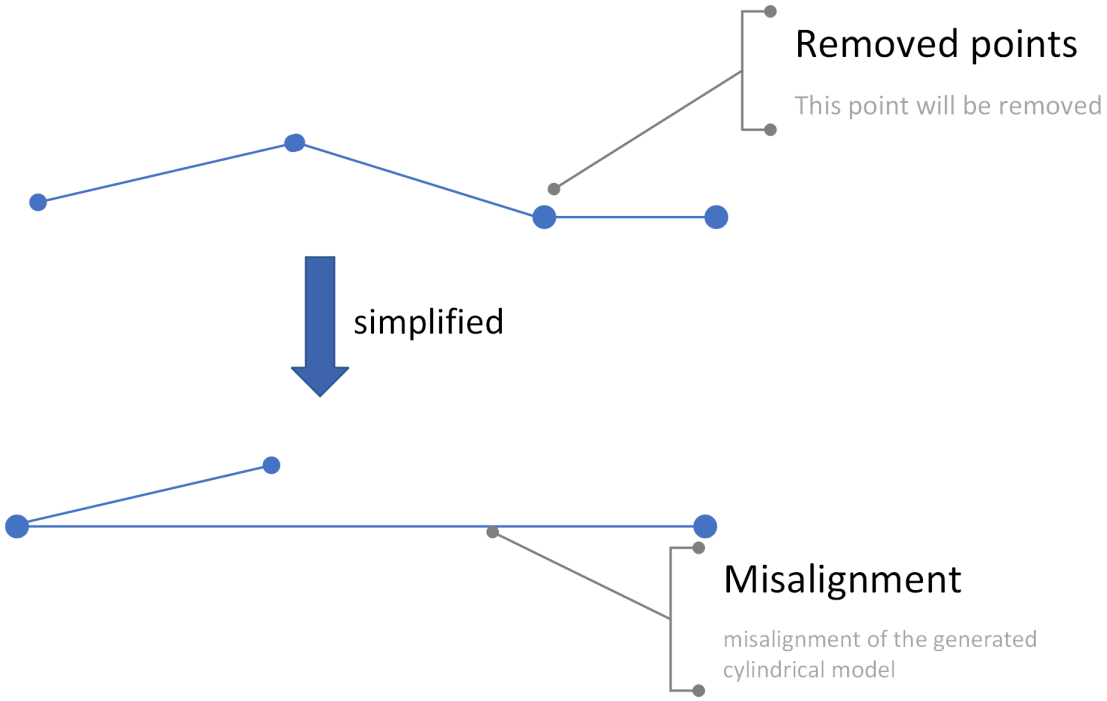

Figure 3.3: If there is no sequential parameter in each point, the 'misalignment' situation described above will occur because of the randomness in adding the points to the data list.

3.3.2 Methods for Simplifying Lists

As seen from the literature in section 2.1.2 (last subsection on X-ray or magnetic resonance imaging), the coordinates in the RSML[38] file can be obtained by sectioning the plant root system and analysing the image[41]. Therefore, the more times they are sliced, the more accurate the coordination on the plant root system becomes and the number of

continuous coordinate points in the data increases.

An analysis of the algorithm's efficiency above shows that the algorithm's speed is related to the amount of data input (in this case, the number of plant coordinates). Therefore, the amount of data input can be optimised to optimise the algorithm's efficiency. In addition, the level of detail of the 3D model of the plant will vary for the different needs of the experiment; some experiments may focus more on observing the root structure, while others make comparisons of the details of the plant root system. Therefore, an optimisation method is used here, removing the more straight points to speed up file creation, reduce file size and adapt to different experimental situations. The simplification algorithm is explained in chapter 3.3.3.

In the evaluation section, three methods will be compared, no optimisation, same interval and best optimisation. This evaluation is done by comparing the results of the three methods in terms of 3D model creation, file size, running speed, and the number of coordinates optimised for each region of the plant root system.

3.3.3 Douglas-Peucker[25]

The Douglas-Peucker[25] method uses point-edge distances as the error metric. The procedure first determines the distance from each intermediate vertex to the edge that connects the initial and end vertices of the original polyline (Figure 3.4) (the orange line in Figure 1, with the red points being the initial and end points in each case). If the distance between the edge and the vertex is greater than the tolerance value, the vertex closest to the edge is marked as the critical point (Figure 3.4)(the point in the diagram where the longest distance lies) and is included in the simplified result, indicating that this point is a 'feature point' in the entire line segment. In contrast, other points that do not fit the requirement are mostly points that cause the segment to be unsmooth.

This process is repeated for each edge of the current simplification until all vertices of the original polyline are within the permissible error range from the edge being checked.

In line 87 of this function, the coordinate list is condensed using the Douglas-Peucker[25, 33](Figure 3.5) technique, and the list is then acquired using the call-in line 88. The

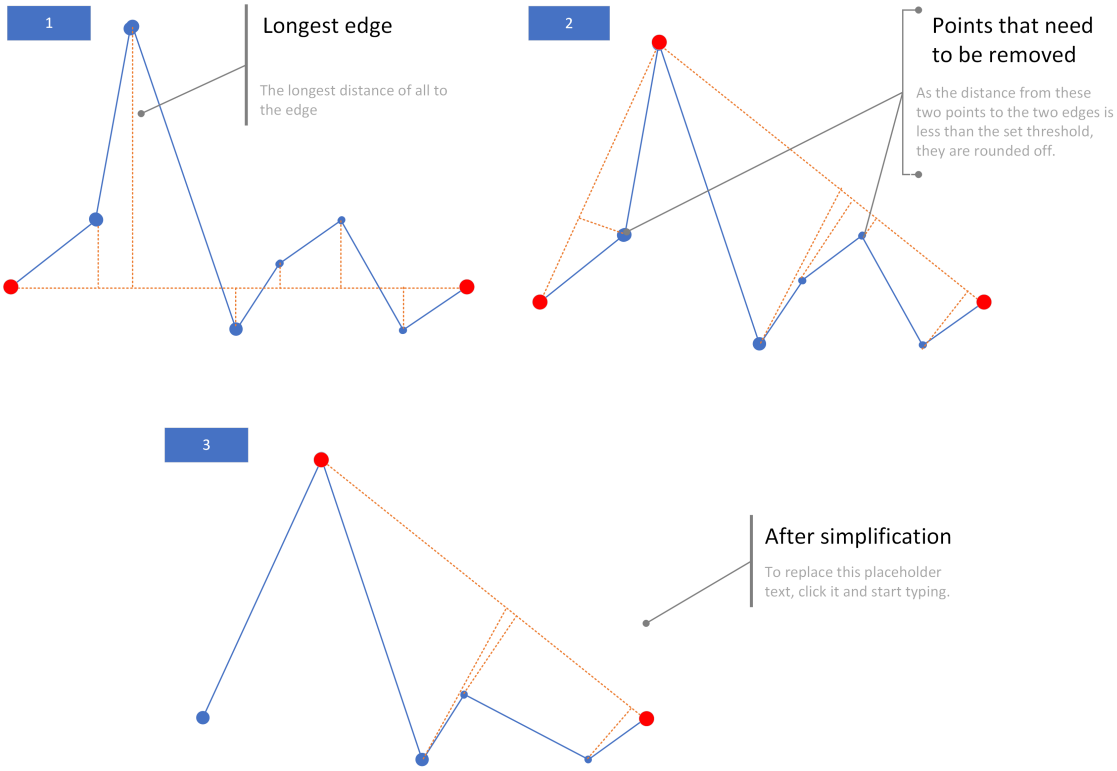

Figure 3.4: Graphical interpretation of the Douglas-Pick algorithm: Figure 1 shows the longest distance (compared to the threshold).In Figure 2, the first and last points are joined to the point with the largest edge, and then the mismatched points are removed in each of the two sectionsleft and right hand. Figure 3, the result of the removal, continues on the right-hand side until all the non-matching points have been removed.

arrangement of the coordinate points is probably wrong in the condensed coordinate list, in any case. In this instance, the index key is utilised to descend the list and supply a set of sequential coordinates from which the suitable model may be identified.

The diagram (Figure 3.5) above shows the details of this algorithm, as shown on the left-hand side of the diagram (Figure 3.5), where the data size of tolerance is added to the incoming coordinate column, and then this optimisation algorithm is called. However, in a compressed list of coordinates, the alignment of the coordinate points is likely to be incorrect. In this case, the index key drops the list and provides a continuous set of coordinates from which the correct model can be determined.

The application part of the code, the recursive technique used to optimise the eligible coordinate points in the corresponding sectors in lines 73 and 74, and the recursive method used to calculate the separation between the third point and the line in the sub-equation above are all depicted on the right-hand side of the diagram.

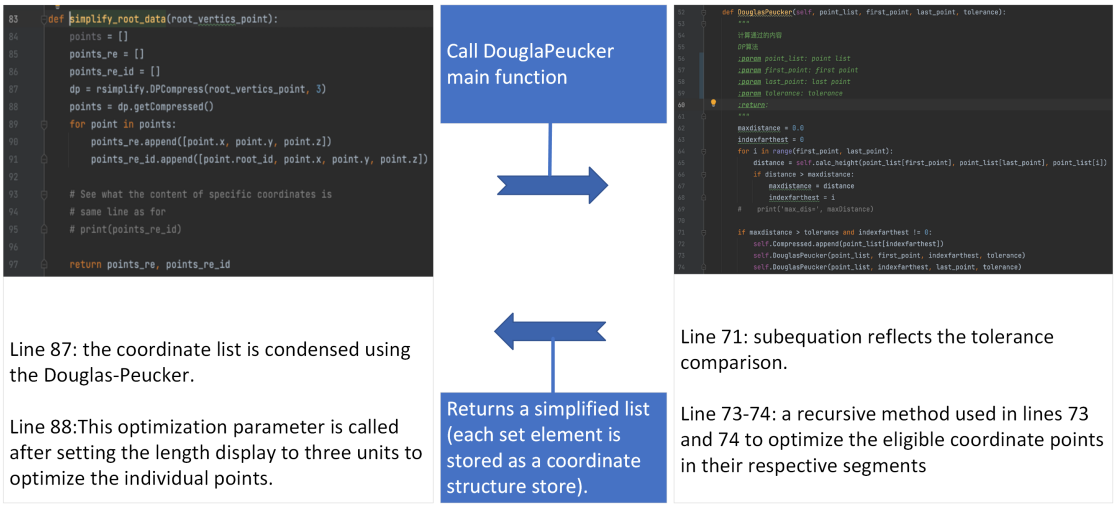

Figure 3.5: Simplifying the use of methods in codeThe simplified process of the equation is shown on the left and right respectively.

Again, here we use the average time complexity to evaluate the algorithm's efficiency (this method is also mentioned in chapter 3.3.1). The average time complexity of this method, where M is the number of points in the simplified fold, is O (N log M), while the worst time complexity is O(NM). The method is, therefore, sensitive to the output. The algorithm can be executed quicker if M is small.

3.4 AR Application

In order to integrate AR functionality in Unity[23], the AR application was built using two extension packages: Unity[23] XR and Lean touch. The former contains the basic AR functionality and parameters, while the latter allows for gesture-free procedural control, such as pinching the model to zoom in.

And here we use Lean touch which can be applied by code to accomplish special gestures such as pinch to zoom, and of course in AR Foundation[67] there are also touch methods to receive tap inputs.

Two functions are implemented in the code to enable additional information to be displayed when the model is tapped; the first function combines the input 3D root model file with a CSV[63] data collection file[63] (containing root data for each plant) to produce an input model prefab for the AR application; the second function incorporates the basic

principles of image tracking to enable multi-object detection so that after scanning various 2D codes can be scanned to display a variety of related content and data information.

3.4.1 Combination of two models

Initially, we wanted to add more information to the model file by code-converting the material. However, after trying this method, it was not feasible to generate it directly via scripting, as the 3D root model (obj file[19]) created using the model converter would only include the model data for the mesh. Therefore, to add more information using a script in Unity[23], a CSV file[63] will be generated in the model converter in conjunction with the 3D model file.

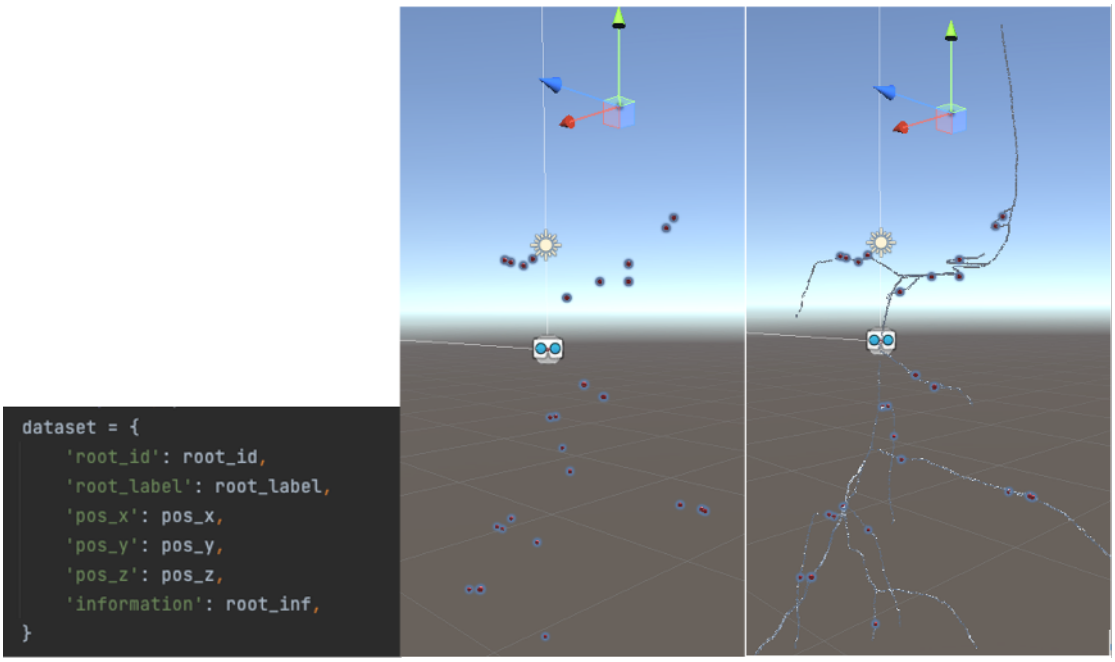

Figure 3.6: Each sub-data is contained in the CSV[63] data file in the leftmost diagram. The middle and right-hand diagrams show the 3D model of the data generated via Unity[23] scripting and the combined OBJ file.

The RSML[38] file contains the ID number of the root, the surname, the midpoint of each segment (figure 3.6), and additional details about each segment, such as whether it is a dead root, as shown on the far left of the image above. All of this data will be presented in the AR application in the form of a CVS file[63].The red sphere (middle picture (Figure 7)), which was created for each midpoint as a child of the root component list for each segment, matches the original OBJ file[19] better since the list of this data was received

during the production of the obj file (as shown in the rightmost image (Figure 3.6)).

3.4.2 Comparing the Two Ways of Generating Models

The two sets of models produced as described in Chapter 4.4.1 were created in two ways: root information data was passed through a CSV file[63] called in a Unity[23] script, and the second was generated directly using python code. Because each red sphere created in this way has its inherent properties (red material, text object box to hold additional information), the properties are built into Unity[23].
In order to use each sphere as the entity's location for the click trigger, it is therefore created here by providing the centre coordinates. The underlying reason for this is that the 3D structure of the root system is created using an OBJ file[19], and no additional root information parameters can be added.

3.4.3 A position Issue Involving Two sets of Models

The script in Unity[23] will build the models, resize the two sets of models and store them in a prefab of the parent-child set suitable for subsequent application of the AR program, which is sized to the actual size of the AR program and at the origin of Unity's[23] coordinate space. The following problems may arise with this program and the solutions to each of them.

- Problems with the generated obj model.
- Model specs in several 3D coordinate systems converted.
- Moving the positions of two sets of models is a difficult task.

The 3D model file generated by the converter uses the coordinate points from the RSML[38] files, so the size and position created in this case do not match the Unity[23] coordinate system format. The solution, shown below, is to generate a temporary instance of the 3D model in Unity's[23] script (line 25 of the code in the image (Figure 3.7)) to prevent direct modification of the source file. Failure to do so would result in shrinkage and rotation

being applied to the same file, contaminating the original 3D file and making the two sets of models consistently mismatch.

```
22
23          // load the OBJ file , the file should be store in Resources folder
24          GameObject obj_loader = Resources.Load(OBJPath) as GameObject;
25          GameObject obj_prefeb = GameObject.Instantiate(obj_loader) as GameObject;
26
```

Figure 3.7: This image is sourced from a script file in Unity[23] to explain the use of an obj file to generate a 3D model instance.

In contrast to the right-handed legal coordinate system used to construct the coordinate system imported by CSV[63], the kind of coordinate system used in Unity[23] is left-handed, meaning the y-axis faces upwards. The starting coordinates of each red dot are created by changing y into a negative number and then rotating it towards the z-axis to complete the overlap between this model set and the OBJ model to provide a more logical structure for the "red dot model set."

```
81          //center point
82          Vector3 center_point = inf_parent.transform.GetChild(0).transform.position;
83
84          inf_parent.transform.position = -center_point;
85          obj_prefeb.transform.position = -center_point;
86
87          // add all into the one gamepbject parent
88          inf_parent.transform.SetParent(root_file.transform);
89          obj_prefeb.transform.SetParent(root_file.transform);
```

Figure 3.8: The code shown here is from a Unity[23] script designed to process the centroid coordinates from a CSV[63] file and convert them to conform to the Unity[23] coordinate system.

The model must then be moved for the AR application to show it accurately since specific models are constructed using a reference point that is not in the model's centre. Here, the central reference point to match both is the centroid on the main root or the first value of the list. The first child element's coordinates are retrieved in the code (Figure 3.8) as shown below, and the two models are then translated to one another to provide the centre point alignment. All panning procedures are carried out after the model has been shrunk since it also influences how the centre point is located.

3.5 Multiple Target Image Tracking

The system has an AR Tracked image manager in Unity[23] that tracks several pictures, but only one model is presented. To meet the demand of implementing various QR codes

with various trigger models, we created a code to achieve the requirement.

Our design is based on the AR Foundation[67] application in Unity[23] and is mainly linked to the AR Tracked image manager, through which the former is applied for some basic tracking tasks. The primary purpose of the code is to link multiple QR codes to the corresponding 3D root model prefab based on the same filename. When the phone detects the QR code stored in the reference library, the application displays the root model with the same file name.

The specific content of the procedure can be found in the appendix A.1.

Chapter 4

Evaluation and Discussion

4.1 File Generation Efficiency

By comparing cross-sectionally the file size, the amount of time required to create the file, and the number of processing points, the effectiveness of the simplified method is assessed. The same RSML[38] file is used as the evaluation's input. The data indicated above are presented in the form of a graph, and then the efficacy of the output model is compared.

4.1.1 Comparison of File Sizes

The following table compares three different simplification methods (method 1: Douglas-Peucker, method 2: fixed step size set to 3, method 3: no simplification) in terms of time taken to generate the model file, the number of the remaining points, and file size are compared.

The first table shows that method 1(Douglas-Peucker) consumes the least time, with the

Table 4.1: Results obtained for sample 0380 roots.

	Method	Time_cost	Points	File_size
0	method 1(Douglas-Peucker)	3.75s	364	1.7 MB
1	method 2(Fixed Step Size Set)	17.66s	3417	18.0 MB
2	method 3(No Simplification)	49.70s	10260	55.7 MB

Table 4.2: Results obtained for sample 0414_roots. rsml

	Method	Time_cost	Points	File_size
0	method 1(Douglas-Peucker)	14.54s	1382	6.3 MB
1	method 2(Fixed Step Size Set)	57.14s	10457	56.4 MB
2	method 3(No Simplification)	156.18s	31413	179.0 MB

number of optimized points reduced from 10,260, the number of all points in method 3(no simplification) to 364, and the file size is only 1.7 MB. Method 2 and method 3 consume three times as much time, and since the step size is 3, it is clear that the corresponding time spent and file size are also three times proportional.

Finally, it was found that the file size was proportional to the number of points; the greater the number of points, the larger the file, as can also be seen by comparing the third record in Table 4.1with the second data in Table 4.2, where the time spent when processing approximately 10,000 data is almost very close to the practical consumption.

4.1.2 Comparison of Data on Root Segment

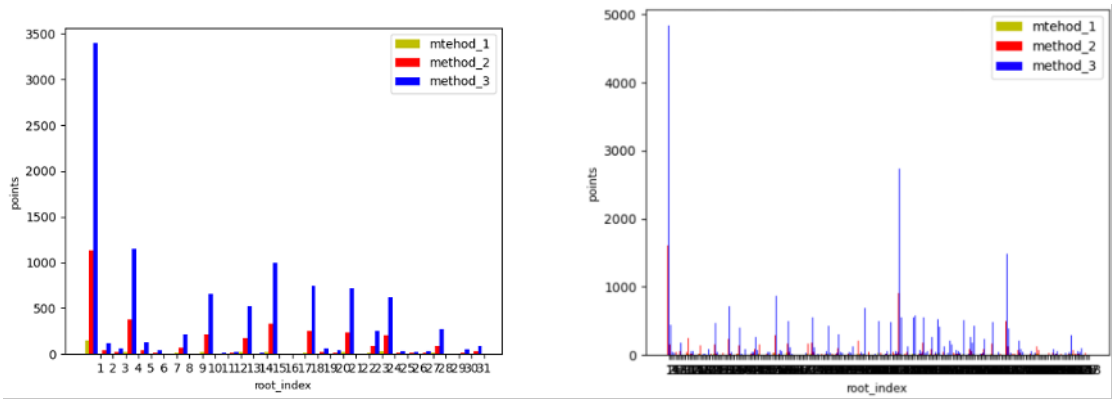

Figure 4.1: number of points optimized: The figure shows the number of points conserved on the different root segments of the RSML[38] file, the bottom number is the serial number of the root segment (due to the number of segments on the right, the serial numbers overlap resulting in a black line).

The graph (Figure 4.1) above shows the number of points present in each part of the root for two different input files, which allows a clearer view of the trend in the development of the root by comparing the number of points that were optimized in each method (method 1 (Douglas-Peucker): marked in yellow, method 3 (no simplification): in blue). Since this

optimization method aims to remove the uncouthness of the model, the presence of too many points optimized away means that there are many unsmooth points in this segment. In addition, due to the size of the set tolerance, the non-conforming coordinate points are removed accordingly, which is what causes the optimal size and time of its generated file.

Comparison of Generated 3D Models

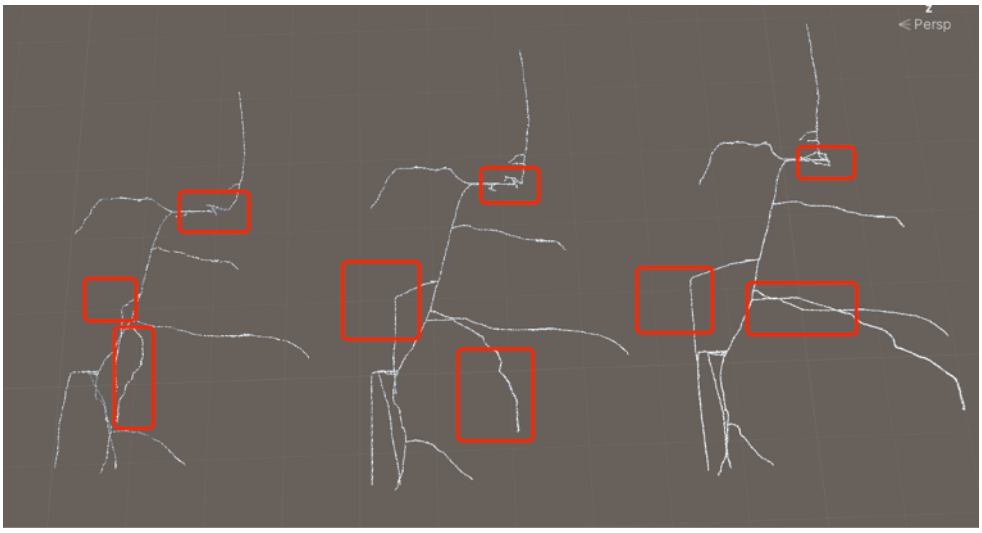

Figure 4.2: The red boxes represent the areas of difference between the three models. From left to right in the diagram are the results of the reduced level of simplification, and it is clear that some of the features have been removed in the cases where the simplification algorithm has been used.

From left to right, the model shown above (Figure 4.2) is methods 1 to 3. The rightmost model is the reference base, which retains all the coordinates, so the details are the most complete, but each red box shows a clear difference and a mistake in the shape of the plant roots. The reason for this can be understood as the omission in 2 causes some of the features on crucial points to be lost, as the step distance is the same each time, and the point removed each time may have this feature.

The disappearance of some of the plant roots in model 1(Figure 4.3) is probably due to the setting of the tolerance, a threshold value that causes points smaller than this to be removed so that some key points may be removed, eventually leading to a change in the orientation of the lateral root model, which can also be seen in the red box above, where the disappearance of several points leads to a change in the orientation of the root system

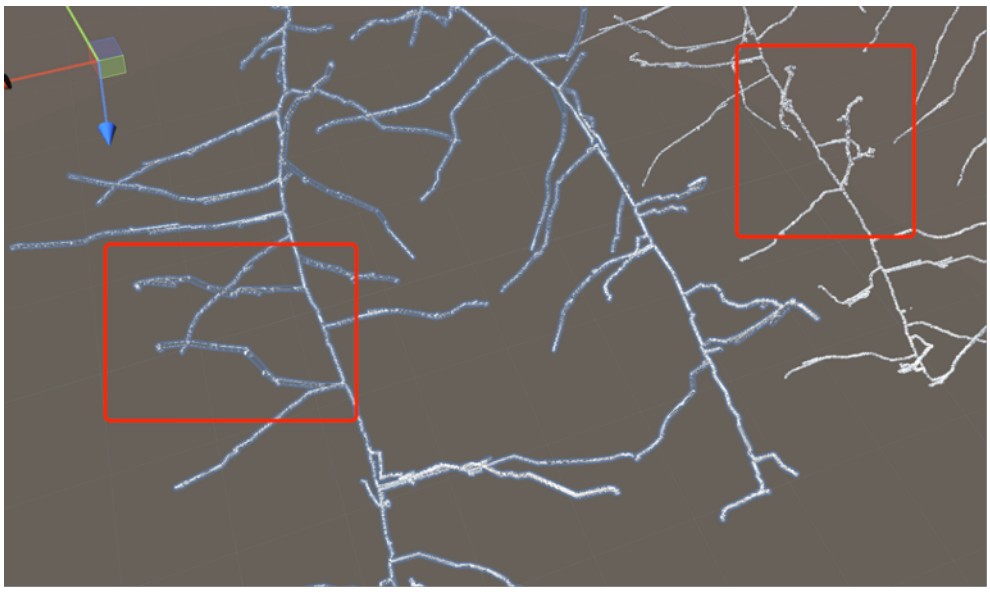

Figure 4.3: Comparison of the generated 3D models more detail, a reduction in the left side of the root system characteristics between the two can be seen

in the red box on the left. As the tolerance is reduced, the bending becomes closer and closer to the original, fully optimized model.

4.2 Model Combiner Evaluation

- Check to see if the two models are placed correctly and mesh the two sets of models' layouts.
- The read file's source file does not change.
- The size of the output prefab places follows the standard.

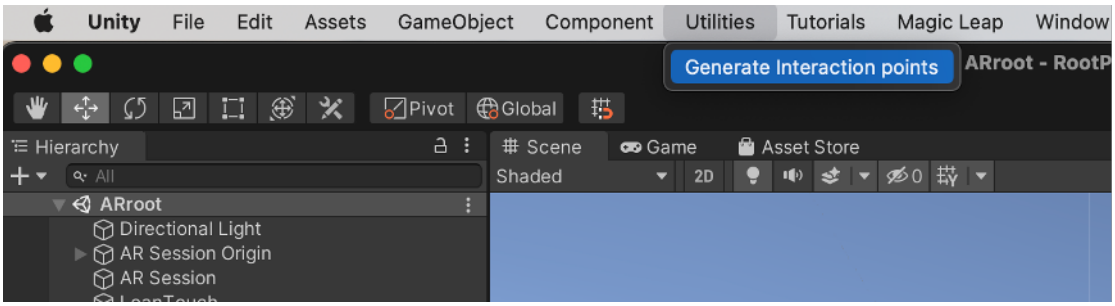

Figure 4.4: Location of tool calls in the menu.

They are selecting Utilities from the menu above Unity[23] (see above (Figure 4.4)) and

choosing to Generate Interaction Points creates a prefabricated part of the model that meets the requirements listed in section 3.4.1. The text components in the unit model for each red marker point meet the requirements (see below (Figure 4.5)); each red sphere marker point is located in the centre of each segment of the root, and all are generated accurately.

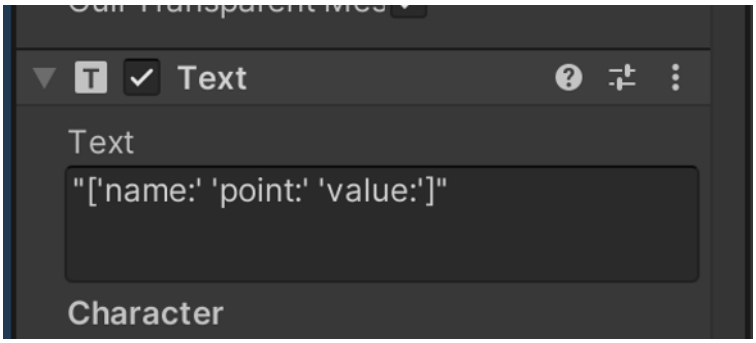

Figure 4.5: Text Component: in red sphere marker point.

The input is a 3D model file read from the OBJs folder in the resources and a dataset file read from the CSV folder, both converted and output as a prefab in the assets folder (see below (Figure 4.6)). The files are generated in the correct location and remain in the correct location of the model after multiple conversions. The files are created in the correct area and remain for the model even after multiple conversions.

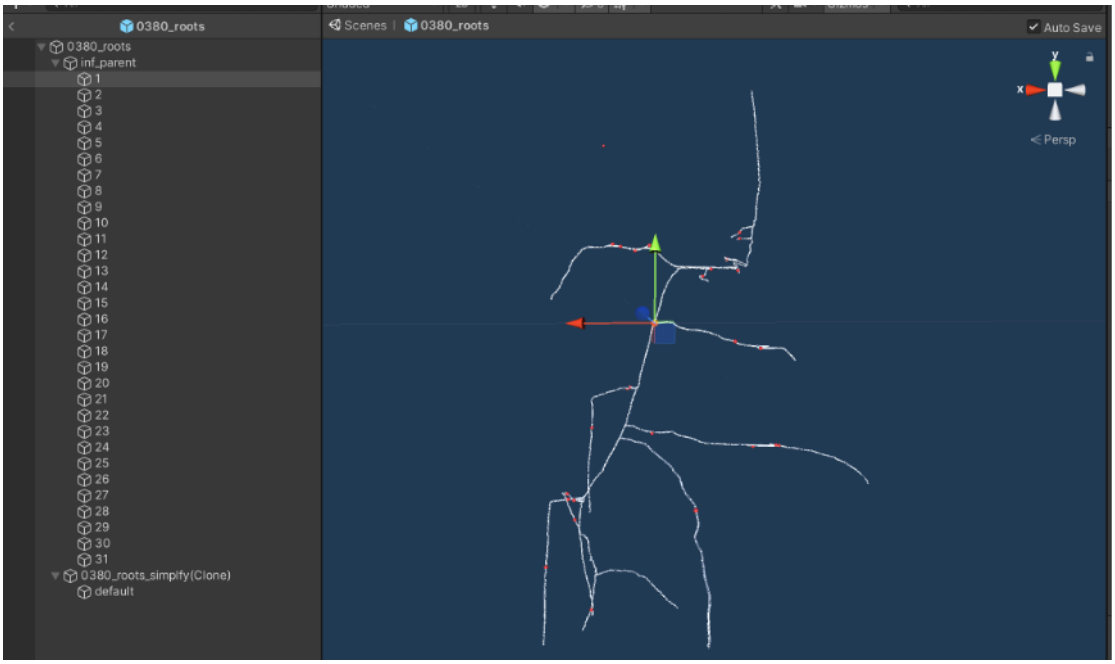

Figure 4.6: Text Component: in red sphere marker point.

4.3 To Evaluate Augmented Reality Applications

The model's name here corresponds to the image name as in the previously discussed approach. It is used to identify which item is now being scanned when using multi-target detection. The object's name is displayed in the centre of the screen during image detection. When the relevant model is presented, as seen in the image (Figure 4.7) on the far left, the matching model appears in the bottom centre of the marker.

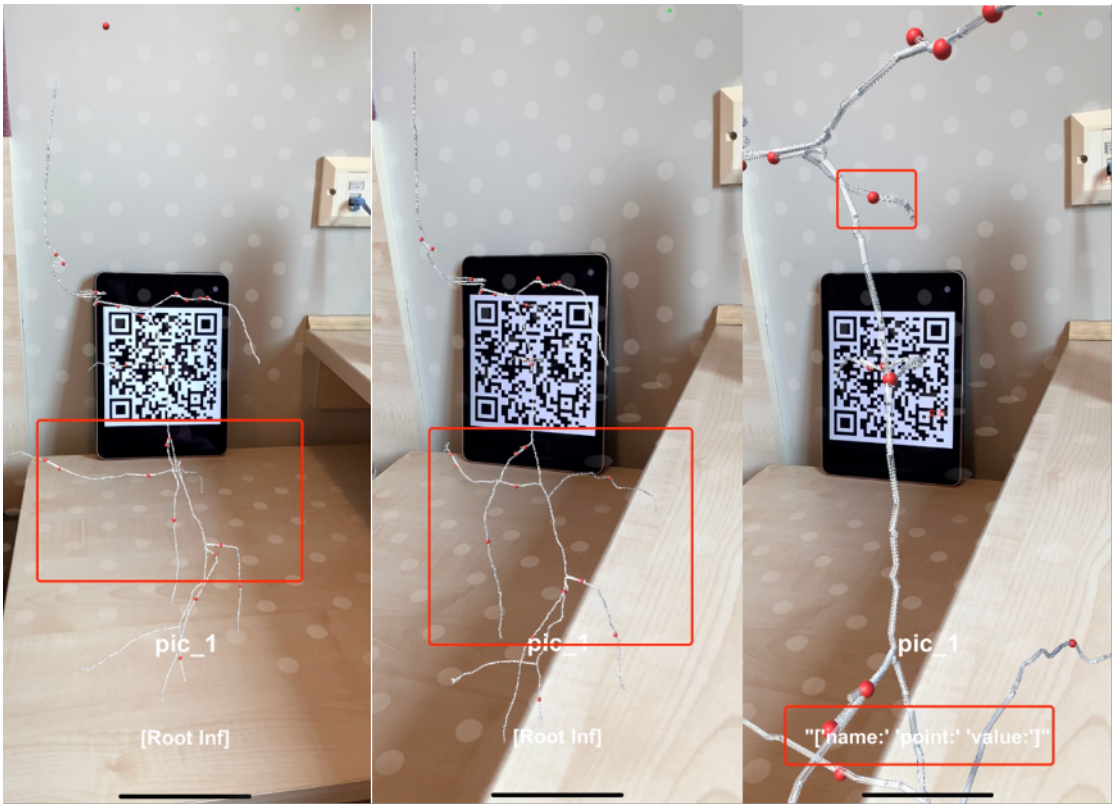

Figure 4.7: Example test in an AR application, viewed at different angles in the middle left, with the effect after using the zoom on the far right. In the middle diagram from the left-hand path, when the viewpoint changes, the viewpoint of the model in its red box also changes accordingly. The rightmost figure verifies the zoom function and the ability to click on the root information to show the corresponding information in the interface after clicking on the red box at the top right.

The model that the camera view is now observing changes with the camera under the current goal. In keeping with how the mobile phone lens moves, the model's size decreases as the marker does, and its location also shifts in response to the marker's movement. Figure (Figure 4.7 right) demonstrates that the model and red dot can be seen in the rightmost picture when zoomed in. The information imported from the CSV[63] file is

presented in the information box at the bottom of the screen when the model is zoomed in using the pinch technique, and its contents are saved in the text component, which is displayed immediately underneath the screen.

Chapter 5

Summary and Reflections

To conclude, the paper analyses four of the more common root systems currently available - paper-based, Agar, Hydroponic and X-ray - and compares each platform's advantages and disadvantages to determine each experiment's needs. An AR program was designed to meet the needs of X-ray-based experiments at the Hounsfield facility, solving the problem of root systems not being visible in the soil. The experiment also discusses the requirements that may arise in three other types of experimental platforms, but none of them has been tested in a real-world environment.

The paper also compares the root image analysis software and data presentation methods used in the different experimental platforms. Based on the current software's advantages and disadvantages and the platforms' interoperability, the RSML[38] format was chosen as the input file for the experimental data. Three simplified approaches were also compared, reducing the amount of input of coordinate points. However, in the evaluation, it was found that the application was not practical and that the best algorithm aimed to make the model smoother and therefore discarded many of the characteristics of the root system model. Moreover, its generation time is related to the file size and the number of input coordinates.

In addition, the article discusses and compares different frameworks for AR and finally settles on AR Foundation as the undeveloped platform. The problem of 3D model files' inability to display additional root information is solved by using two sets of overlapping models. The currently designed program can only be relatively simple CSV[63] data files.

The issues that arose in the design are discussed and described in the methodology. The program works fine in the evaluation chapter.
Verified by the evaluation above, the program completes the verification of the mentioned function. As the functions and information it displays are not yet complete, the help it can offer researchers is still relatively small, so the following upgrades and future upgraded versions could be better.

5.1 Limitation

Several limitations to the procedure and this experiment need to be explained. Firstly, the model generator can currently only read a single file. In contrast, the amount of data is enormous for a large number of samples in the laboratory. Therefore, developing appropriate algorithms for multi-file processing and optimisation is necessary.
Additionally, the Simplified coordinate algorithm employed here is inconsistent with the actual goal of eliminating some of the contiguous points while keeping the plant root features. Instead, the algorithm concentrates more on the model's "unsmooth" regions while omitting many root features.
The red trigger points are now utilised as the information triggering technique for the AR program, which is relatively straightforward. The application has not been tested in a specific experimental environment, despite being considered for various scenarios. Because of this, the application based on 2D image tracking was unable to thoroughly validate and illustrate the position of its 3D root system in a real-world setting.
Regarding data presentation, the current data acquisition method is to read the root system data from the RSML[38] file and present it on the AR interface through a CSV[63] data-set file, which is a single way of presenting this root information, and that is not detailed enough.

5.2 Future upgrades

5.2.1 Model Generator

- Code run efficiency.
- Some of the data in the code is extracted incorrectly.
- Multi-document processing.

In terms of the python model converter, there is a lot of code redundancy in this part because multiple functions of code are used to parse the RSML[38] file, and further work should be done to encapsulate its various parts and reduce subcategories.
In extracting some of the root data, the recursive approach is used to get the data needed through the advantage of the python package. However, when assigning values and re-processing them, there is a loop nested in a loop. The time complexity of this changes as the number of data increases, so the algorithm should be further optimised.
When the model generator extracts the root information data from a file, it looks for functions to get the data under a specific root tag. However, the current type of data extraction does not work well due to problems with some of the code. Also, when dealing with one-to-one correspondence, without any simplification, error outliers occur when processing to a certain point does not comply with the rules of mathematics and a control situation occurs.
When reading a file, only one file is processed at a time for model creation since the current file processing limit is set to one. In order to achieve the extended functionality of the read file function, it was only necessary to extend it to multiple file reads when comparing the effectiveness of three different optimised functions because the combination of three function calls tried in the primary function worked, and files could be generated.

- Cylinder model generating optimization.

The mesh of the model used here can be thought of in practice as a triangular dispersion of the model surface rather than a smoother model surface. The model is constructed

by providing two coordinate points, start and end, radii to generate the corresponding cylinders. The entire model is typically made up of several such cylinders. The answer is to optimise each cylinder using the cylinder function included in the Python CSG package(Python package). If this is successful, we can add smooth components to our function and round off the top and bottom of each cylinder to perform even more optimisation.

5.2.2 Simplified coordinate algorithm

As the algorithm(Douglas-Peucker[25]) was proposed very early, subsequent updated versions are also available with more upgrades to its efficiency and methodology, in which case experimental designs can be attempted for it using subsequently upgraded algorithms. Also, the algorithm can be modified to be more suitable for this project based on its smoothness. However, the ultimate aim is to remove straight points that do not affect the structural characteristics of the root system.

5.2.3 AR applications

- Practical application of the program in experiments

As we have not tested the program in a real situation here, many requirements and practical application aspects have not been addressed, such as the potential problem of the location of the root model.

- The position and size of the model.

In the case of the interaction points generator, the position of its model takes the location of the QR code as a reference point. Currently, the QR code's centre point corresponds to the model's centre point. However, in practice, the position of the QR code and the planting may differ. In the next step, the relationship between the two positions can be checked and applied to the code before further adjustments are made to the prefab.
A possible next step would be to combine the application with an actual situation to improve the procedure's generality.

- Optimized interface display.

The information displayed on the current interface only includes the name of the sample corresponding to the QR code currently being scanned and the plant root information generated when the segment is clicked. In order to display more information, more information, such as images and animations, can be added to the original information box. These components can be extended using unity components for the corresponding content.

5.2.4 Desktop application

Some of the functions currently designed for AR programs, such as image tracking, are similar to those of the desktop program RootVIs, so the two can be used consistently in terms of usage patterns. It is possible to use TCP/IP to transfer the samples' names or other information after scanning by the AR program to the corresponding desktop program for detailed viewing.

Comparison between two root system model samples

Figure 5.1: The function to compare 3D structures in RootVis, with buttons below each model to view more information.

In RootVis there is a function to compare two root model samples, which is also the source of inspiration for multiple image detection in AR programs. The two can view each sample's details together, such as the root system, at different angles. Then the

desktop system can receive the name of the sample name, which allows the desktop to investigate more information about each sample (as shown in (Figure 5.1)).

The multi-file input and search function in RootVis also helps the user to identify the sample to be viewed more quickly. (as shown in the picture's(Figure 5.2) search box in the top right corner).

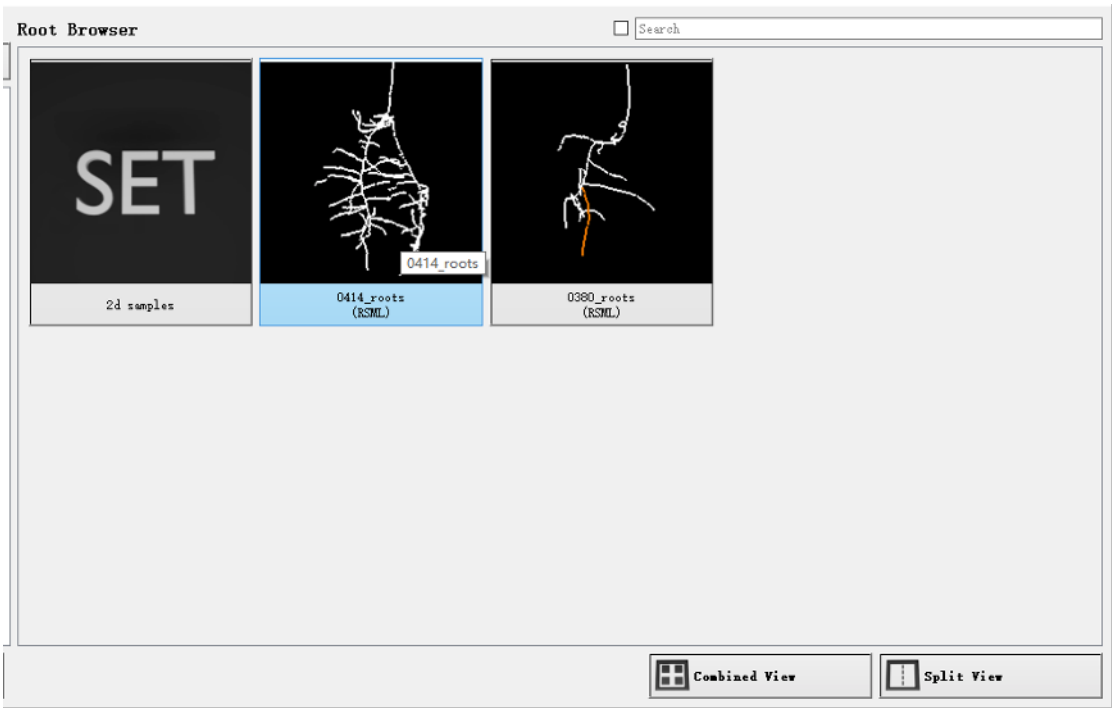

Figure 5.2: Multi-file input and search function in RootVis, with the search box on the top right in the picture.

Bibliography

[1] Arcore.

[2] Vuforia developer library, 2021.

[3] Arkit, 2022.

[4] Reference image library, 2022.

[5] Atkinson, J. A., Pound, M. P., Bennett, M. J., and Wells, D. M. Uncovering the hidden half of plants using new advances in root phenotyping. *Current opinion in biotechnology 55* (2019), 1–8.

[6] Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., Yergeau, F., and Cowan, J. Extensible markup language (xml) 1.0, 2000.

[7] Cherkassky, V., and Ma, Y. Practical selection of svm parameters and noise estimation for svm regression. *Neural networks 17*, 1 (2004), 113–126.

[8] Collins, T. J. Imagej for microscopy. *Biotechniques 43*, S1 (2007), S25–S30.

[9] Delory, B. M., Baudson, C., Brostaux, Y., Lobet, G., Du Jardin, P., and Delaplace, P. archidart: an r package for the automated computation of plant root architectural traits. *Plant and Soil 398*, 1 (2016), 351–365.

[10] Diggle, A. Rootmap—a model in three-dimensional coordinates of the growth and structure of fibrous root systems. *Plant and soil 105*, 2 (1988), 169–178.

[11] Dragota, M.-I., and Sabou, A. Nsmuseum-a case study for developing cross-platform mobile applications in augmented reality. In *RoCHI*, pp. 74–81.

[12] FACILITY, H. Hounsfield facility, 2014.

[13] FRENCH, A. P., WILSON, M. H., KENOBI, K., DIETRICH, D., VOSS, U., UBEDA-TOMÁS, S., PRIDMORE, T. P., AND WELLS, D. M. Identifying biological landmarks using a novel cell measuring image analysis tool: Cell-o-tape. *Plant Methods 8*, 1 (2012), 1–8.

[14] FURBANK, R. T., AND TESTER, M. Phenomics–technologies to relieve the phenotyping bottleneck. *Trends in plant science 16*, 12 (2011), 635–644.

[15] GALKOVSKYI, T., MILEYKO, Y., BUCKSCH, A., MOORE, B., SYMONOVA, O., PRICE, C. A., TOPP, C. N., IYER-PASCUZZI, A. S., ZUREK, P. R., AND FANG, S. Gia roots: software for the high throughput analysis of plant root system architecture. *BMC plant biology 12*, 1 (2012), 1–12.

[16] GERTH, S., CLAUSSEN, J., EGGERT, A., WÖRLEIN, N., WAININGER, M., WITTENBERG, T., AND UHLMANN, N. Semiautomated 3d root segmentation and evaluation based on x-ray ct imagery. *Plant Phenomics 2021* (2021).

[17] GIOIA, T., GALINSKI, A., LENZ, H., MÜLLER, C., LENTZ, J., HEINZ, K., BRIESE, C., PUTZ, A., FIORANI, F., AND WATT, M. Growscreen-page, a non-invasive, high-throughput phenotyping system based on germination paper to quantify crop phenotypic diversity and plasticity of root traits under varying nutrient supply. *Functional Plant Biology 44*, 1 (2016), 76–93.

[18] GODEFROID, P., KLARLUND, N., AND SEN, K. Dart: Directed automated random testing. In *Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation*, pp. 213–223.

[19] GOGUEN, J. A., WINKLER, T., MESEGUER, J., FUTATSUGI, K., AND JOUANNAUD, J.-P. *Introducing obj*. Springer, 2000, pp. 3–167.

[20] GREGORY, P. J. *Plant roots: growth, activity and interactions with the soil*. John Wiley Sons, 2008.

[21] Grieneisen, V. A., Xu, J., Marée, A. F., Hogeweg, P., and Scheres, B. Auxin transport is sufficient to generate a maximum and gradient guiding root growth. *Nature 449*, 7165 (2007), 1008–1013.

[22] Griffiths, M., Mellor, N., Sturrock, C. J., Atkinson, B. S., Johnson, J., Mairhofer, S., York, L. M., Atkinson, J. A., Soltaninejad, M., and Foulkes, J. F. X-ray ct reveals 4d root system development and lateral root responses to nitrate in soil. *The Plant Phenome Journal 5*, 1 (2022), e20036.

[23] Haas, J. K. A history of the unity game engine. *Diss. WORCESTER POLYTECHNIC INSTITUTE 483* (2014), 484.

[24] Hainsworth, J., and Aylmore, L. The use of computer assisted tomography to determine spatial distribution of soil water content. *Soil Research 21*, 4 (1983), 435–443.

[25] Hershberger, J. E., and Snoeyink, J. Speeding up the douglas-peucker line-simplification algorithm.

[26] Hounsfield, G. N. Computed medical imaging. nobel lecture, decemberr 8, 1979. *Journal of computer assisted tomography 4*, 5 (1980), 665–674.

[27] Hufnagel, B., de Sousa, S. M., Assis, L., Guimaraes, C. T., Leiser, W., Azevedo, G. C., Negri, B., Larson, B. G., Shaff, J. E., and Pastina, M. M. Duplicate and conquer: multiple homologs of phosphorus-starvation tolerance1 enhance phosphorus acquisition and sorghum performance on low-phosphorus soils. *Plant physiology 166*, 2 (2014), 659–677.

[28] Iyer-Pascuzzi, A. S., Symonova, O., Mileyko, Y., Hao, Y., Belcher, H., Harer, J., Weitz, J. S., and Benfey, P. N. Imaging and analysis platform for automatic phenotyping and trait ranking of plant root systems. *Plant physiology 152*, 3 (2010), 1148–1157.

[29] Jahnke, S., Menzel, M. I., Van Dusschoten, D., Roeb, G. W., Bühler, J., Minwuyelet, S., Blümler, P., Temperton, V. M., Hombach, T., and

Streun, M. Combined mri–pet dissects dynamic changes in plant structures and functions. *The Plant Journal 59*, 4 (2009), 634–644.

[30] Kell, D. B. Breeding crop plants with deep roots: their role in sustainable carbon, nutrient and water sequestration. *Annals of Botany 108*, 3 (2011), 407–418.

[31] Kirmeyer, G. Ar foundation. *Optimizing Chloramine Treatment: American Water Works Association* (2004).

[32] Le Marié, C., Kirchgessner, N., Marschall, D., Walter, A., and Hund, A. Rhizoslides: paper-based growth system for non-destructive, high throughput phenotyping of root development by means of image analysis. *Plant methods 10*, 1 (2014), 1–16.

[33] Liu, J., Li, H., Yang, Z., Wu, K., Liu, Y., and Liu, R. W. Adaptive douglas-peucker algorithm with automatic thresholding for ais-based vessel trajectory compression. *IEEE Access 7* (2019), 150677–150692.

[34] Liu, S., Barrow, C. S., Hanlon, M., Lynch, J. P., and Bucksch, A. Dirt/3d: 3d root phenotyping for field-grown maize (zea mays). *Plant physiology 187*, 2 (2021), 739–757.

[35] Lobet, G. The story behind the root system markup language, 2015.

[36] Lobet, G., Koevoets, I. T., Noll, M., Meyer, P. E., Tocquin, P., Pagès, L., and Périlleux, C. Using a structural root system model to evaluate and improve the accuracy of root image analysis pipelines. *Frontiers in plant science 8* (2017), 447.

[37] Lobet, G., Pagès, L., and Draye, X. A novel image-analysis toolbox enabling quantitative analysis of root system architecture. *Plant physiology 157*, 1 (2011), 29–39.

[38] Lobet, G., Pound, M. P., Diener, J., Pradal, C., Draye, X., Godin, C., Javaux, M., Leitner, D., Meunier, F., and Nacry, P. Root system markup

language: toward a unified root architecture description language. *Plant physiology 167*, 3 (2015), 617–627.

[39] Lontoc-Roy, M., Dutilleul, P., Prasher, S. O., Han, L., Brouillet, T., and Smith, D. L. Advances in the acquisition and analysis of ct scan data to isolate a crop root system from the soil medium and quantify root system complexity in 3-d space. *Geoderma 137*, 1-2 (2006), 231–241.

[40] Lynch, J. P., Nielsen, K. L., Davis, R. D., and Jablokow, A. G. Simroot: modelling and visualization of root systems. *Plant and Soil 188*, 1 (1997), 139–151.

[41] Mairhofer, S., Johnson, J., Sturrock, C. J., Bennett, M. J., Mooney, S. J., and Pridmore, T. P. Visual tracking for the recovery of multiple interacting plant root systems from x-ray

ct images. *Machine Vision and Applications 27*, 5 (2016), 721–734.

[42] Mairhofer, S., Sturrock, C. J., Bennett, M. J., Mooney, S. J., and Pridmore, T. P. Extracting multiple interacting root systems using x-ray micro-computed tomography. *The Plant Journal 84*, 5 (2015), 1034–1043.

[43] Mairhofer, S., Zappala, S., Tracy, S., Sturrock, C., Bennett, M. J., Mooney, S. J., and Pridmore, T. P. Recovering complete plant root system architectures from soil via x-ray -computed tomography. *Plant methods 9*, 1 (2013), 1–7.

[44] Mairhofer, S., Zappala, S., Tracy, S. R., Sturrock, C., Bennett, M., Mooney, S. J., and Pridmore, T. Rootrak: automated recovery of three-dimensional plant root architecture in soil from x-ray microcomputed tomography images using visual tracking. *Plant physiology 158*, 2 (2012), 561–569.

[45] Mathieu, L., Lobet, G., Tocquin, P., and Périlleux, C. "rhizoponics": a novel hydroponic rhizotron for root system analyses on mature arabidopsis thaliana plants. *Plant methods 11*, 1 (2015), 1–8.

[46] Metzner, R., Eggert, A., van Dusschoten, D., Pflugfelder, D., Gerth, S., Schurr, U., Uhlmann, N., and Jahnke, S. Direct comparison of mri and x-ray ct technologies for 3d imaging of root systems in soil: potential and challenges for root trait quantification. *Plant methods 11*, 1 (2015), 1–11.

[47] Mooney, S. J., Pridmore, T. P., Helliwell, J., and Bennett, M. J. Developing x-ray computed tomography to non-invasively image 3-d root systems architecture in soil. *Plant and soil 352*, 1 (2012), 1–22.

[48] Mykoliuk, D. V. Using interactive technologies to study the evolution of stars in astronomy classes.

[49] Naeem, A., French, A. P., Wells, D. M., and Pridmore, T. P. High-throughput feature counting and measurement of roots. *Bioinformatics 27*, 9 (2011), 1337–1338.

[50] Nagel, K. A., Putz, A., Gilmer, F., Heinz, K., Fischbach, A., Pfeifer, J., Faget, M., Blossfeld, S., Ernst, M., and Dimaki, C. Growscreen-rhizo is a novel phenotyping robot enabling simultaneous measurements of root and shoot growth for plants grown in soil-filled rhizotrons. *Functional Plant Biology 39*, 11 (2012), 891–904.

[51] Oufqir, Z., El Abderrahmani, A., and Satori, K. Arkit and arcore in serve to augmented reality. IEEE.

[52] Oufqir, Z., El Abderrahmani, A., and Satori, K. Arkit and arcore in serve to augmented reality. In *2020 International Conference on Intelligent Systems and Computer Vision (ISCV)* (2020), IEEE, pp. 1–7.

[53] Paez-Garcia, A., Motes, C. M., Scheible, W.-R., Chen, R., Blancaflor, E. B., and Monteros, M. J. Root traits and phenotyping strategies for plant improvement. *Plants 4*, 2 (2015), 334–355.

[54] Patrício, D. I., and Rieder, R. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Computers and electronics in agriculture 153* (2018), 69–81.

[55] Pochtoviuk, S., Vakaliuk, T., and Pikilnyak, A. Possibilities of application of augmented reality in different branches of education. *Available at SSRN 3719845* (2020).

[56] Pornaro, C., Macolino, S., Menegon, A., and Richardson, M. Winrhizo technology for measuring morphological traits of bermudagrass stolons. *Agronomy Journal 109*, 6 (2017), 3007–3010.

[57] Pound, M. P., French, A. P., Atkinson, J. A., Wells, D. M., Bennett, M. J., and Pridmore, T. Rootnav: navigating images of complex root architectures. *Plant physiology 162*, 4 (2013), 1802–1814.

[58] Rahaman, M. M., Chen, D., Gillani, Z., Klukas, C., and Chen, M. Advanced phenotyping and phenotype data analysis for the study of plant growth and development. *Frontiers in plant science 6* (2015), 619.

[59] Rasti, P., Demilly, D., Benoit, L., Belin, E., Ducournau, S., Chapeau-Blondeau, F., and Rousseau, D. Low-cost vision machine for high-throughput automated monitoring of heterotrophic seedling growth on wet paper support. In *BMVC*, p. 323.

[60] Reimer, R., Stich, B., Melchinger, A. E., Schrag, T. A., Sorensen, A. P., Stamp, P., and Hund, A. Root response to temperature extremes: association mapping of temperate maize (zea mays l). *Maydica 58*, 2 (2013), 156–168.

[61] Ruta, N., Liedgens, M., Fracheboud, Y., Stamp, P., and Hund, A. Qtls for the elongation of axile and lateral roots of maize in response to low water potential. *Theoretical and Applied Genetics 120*, 3 (2010), 621–631.

[62] Semerikov, S., Mintii, M., and Mintii, I. Review of the course "development of virtual and augmented reality software" for stem teachers: implementation results and improvement potentials. CEUR Workshop Proceedings.

[63] Shafranovich, Y. Common format and mime type for comma-separated values (csv) files. Report 2070-1721, 2005.

[64] Smit, K., Van Asselen, B., Kok, J., Aalbers, A., Lagendijk, J., and Raaymakers, B. Towards reference dosimetry for the mr-linac: magnetic field correction of the ionization chamber reading. *Physics in Medicine Biology 58*, 17 (2013), 5945.

[65] Trachsel, S., Kaeppler, S. M., Brown, K. M., and Lynch, J. P. Shovelomics: high throughput phenotyping of maize (zea mays l.) root architecture in the field. *Plant and soil 341*, 1 (2011), 75–87.

[66] Uga, Y., Sugimoto, K., Ogawa, S., Rane, J., Ishitani, M., Hara, N., Kitomi, Y., Inukai, Y., Ono, K., and Kanno, N. Control of root system architecture by deeper rooting 1 increases rice yield under drought conditions. *Nature genetics 45*, 9 (2013), 1097–1102.

[67] unity. Ar tracked image manager, 2021.

[68] unity. reference image library, 2021.

[69] van Dusschoten, D., Metzner, R., Kochs, J., Postma, J. A., Pflugfelder, D., Bühler, J., Schurr, U., and Jahnke, S. Quantitative 3d analysis of plant roots growing in soil using magnetic resonance imaging. *Plant physiology 170*, 3 (2016), 1176–1188.

[70] VAN NOORDWIJK, M., and Floris, J. Loss of dry weight during washing and storage of root samples. *Plant and Soil* (1979), 239–243.

[71] White, S., Feiner, S., and Kopylec, J. Virtual vouchers: Prototyping a mobile augmented reality user interface for botanical species identification. In *3D User Interfaces (3DUI'06)*, IEEE, pp. 119–126.

[72] Yazdanbakhsh, N., and Fisahn, J. High throughput phenotyping of root growth dynamics, lateral root formation, root architecture and root hair development enabled by plarom. *Functional Plant Biology 36*, 11 (2009), 938–946.

Appendix A

Code Explanation

A.1 Code for Multiple Target Image Recognition

A.1.1 Multiple Image Recognition Process

As the default image recognition in the system, only one model can be displayed; here, we extend the code for the image detection function.

```
37      private void OnEnable()
38      {
39          trackedImageManager.trackedImagesChanged += ImageChanged;
40      }
41
42      private void OnDisable()
43      {
44          trackedImageManager.trackedImagesChanged -= ImageChanged;
45      }
46
47      private void ImageChanged(ARTrackedImagesChangedEventArgs eventArgs)
48      {
49          foreach(ARTrackedImage trackedImage in eventArgs.added)
50          {
51              UpdateImage(trackedImage);
52          }
53
54          foreach (ARTrackedImage trackedImage in eventArgs.updated)
55          {
56              UpdateImage(trackedImage);
57          }
58
59          foreach (ARTrackedImage trackedImage in eventArgs.removed)
60          {
61              spawnedPrefebs[trackedImage.name].SetActive(false);
62          }
63      }
```

Figure A.1: Code for image recognition extensions in unity.

As seen previously, it inherits from the system's AR tracking image manager. Changes include adding a list of events to respond to changes in the image. This can be thought of as Unity's Update function, which handles changes in real-time, and its main role is to detect changes to the image in real-time (shown on lines 39 and 44 in the image (Figure

A.1).The code (line 61 in the image (Figure A.1)) also shows that the model associated with the image name affects how it is displayed as active and inactive.

```
66      private void UpdateImage(ARTrackedImage trackedImage)
67      {
68          imageTrackedText.text = trackedImage.referenceImage.name;
69          string name = trackedImage.referenceImage.name;
70          Vector3 position = trackedImage.transform.position;
71
72          GameObject prefab = spawnedPrefebs[name];
73
74          prefab.transform.position = position;
75          prefab.SetActive(true);
76
77          foreach(GameObject go in spawnedPrefebs.Values)
78          {
79              if(go.name != name)
80              {
81                  go.SetActive(false);
82              }
83          }
84
85      }
```

Figure A.2: Script used for updates when recognizing images in unity.

After obtaining the name that matches the image (on line 68 of the image(Figure A.2)), the objective of this function is to construct an instance of the model with the corresponding name and activate the display (lines 72, 75 of the image(Figure A.2)) and set other models that do not match the name of the current image to "not visible" (line 81 of the image(Figure A.2)), then models that do not correspond to their QR code will not be displayed.

A.1.2 Set the Prefab for Multiple Object Recognition

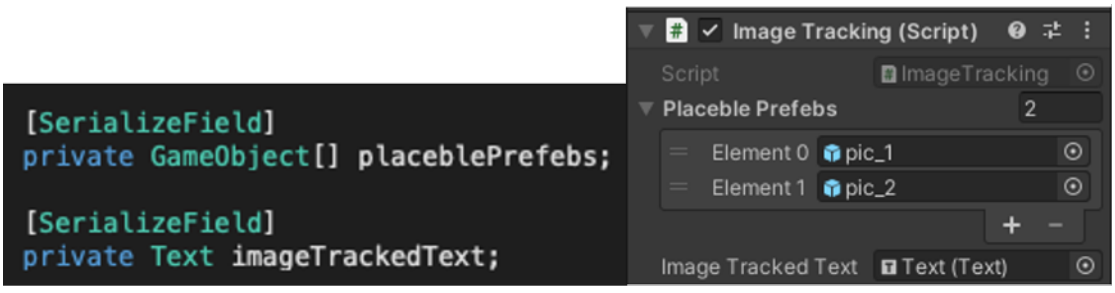

Figure A.3: On the left side of the image are the set variables in the script, the dataset with multiple prefabs, and the data box on the AR interface that displays the name of the current experiment sample. The right-hand side of the image shows the corresponding interface generated by the script in Unity.

As shown in the image (Figure A.3) above, this script is attached to the previously mentioned image recognition reference library and is used to implement the multi-image

recognition function. In addition, due to Unity's interface system, the script can be displayed directly on the design interface to improve its visualization.