

UNIVERSITY OF CALGARY

Annotation of Vascular Plant Structures using Haptic Assistance

by

Philmo Gu

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE

CALGARY, ALBERTA

AUGUST, 2022

© Philmo Gu 2022

# Abstract

The vascular structure within a plant is a network of vascular bundles that transport nutrients and water. Analyzing the organization of vascular bundles can lead to a better understanding of the vascular structure's role in the development of plants. One way to analyze the vascular structure is by scanning plant samples using X-ray micro-CT and then annotate the volumetric data digitally. However, this process is challenging due to the complex arrangement of vascular bundles and lack of contrast from other nearby anatomic structures. To address these problems, we developed a system to annotate the vascular structure of plant samples using force feedback to interact with the surface of objects and along the centerline of tubular structures. This system annotated the vascular structure of flower heads such as *Gerbera hybrida*, and inflorescences such as *Arabidopsis thaliana*. User study participants found the haptic assistance helpful for interacting with and annotating plant samples.

# Preface

This thesis is an extension of the work that was initially started by Andrew Owens. He developed the initial project to visualize and interact with volumetric data using haptic assistance. Note that all the following features are explained in the thesis. I improved on several aspects:

- Performance of original haptic algorithm was improved by generalizing the framework for haptic assistance, and running the relevancy thread to asynchronously from the time-sensitive main haptic loop.
- Two new haptic modes were added. Andrew developed the haptic mode “Proximity Following”. Using that mode as a foundation, the haptic mode “Object Following” further constrains the probe along a tubular structure and the haptic mode “Surface Sampling” touches the surface of volumetric objects while ignoring imperceptible noise.
- Annotation was expanded. Initially, annotation only consisted of painting using flood fill. This was expanded with “Fill” and “Speckle Fill” segmentation modes. In addition, tracing annotation was added.
- A command system was added to quickly undo and redo the most recent annotation command. The memory requirement has been greatly reduced from the previous command system for painting.

- The visualization originally consisted of direct volumetric rendering using ray-casting. This was expanded to include ambient occlusion, and depth colour blending.
- A GUI was developed using Qt to interactively edit settings using widgets.
- Performed a user study to evaluate the usefulness of haptic assistance. This involved applying for permission from the university, creating instructions for the participants, running the user study, and analyzing the results.
- Annotated the vascular structure within a variety of inflorescences and flower heads.

No part of this thesis has been previously published.

# Acknowledgements

There are several people that I would like to thank. My supervisor, Dr. Przemyslaw Prusinkiewicz, provided a wealth of knowledge, inspiration, and endless support. He stayed with me until the end, even when I felt lost. Andrew Owens created this project with the haptic mode “Proximity Following” and assisted me every step of this project. Teng Zhang and Paula Elomaa from Department of Agriculture Sciences in University of Helsinki provided the plant samples that were analyzed and provided biological insights into the development of flower heads. Mikolaj Cieslak, a senior researcher from our lab, and Jarvis Stobbs from Canadian Light Source scanned and reconstructed the plant samples using the synchrotron. My former supervisor, Dr. Sonny Chan, introduced me to the world of haptics. The former and current lab members of the Biological Modeling and Visualization Research Group will have a place in my heart for being an awesome group of people doing cool stuff. Finally, I would like to thank my family for believing in me.

# Table of Contents

<b>Abstract</b>	<b>ii</b>
<b>Preface</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>Table of Contents</b>	<b>vi</b>
<b>List of Figures and Illustrations</b>	<b>viii</b>
<b>List of Tables</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Background</b>	<b>5</b>
2.1 History of Visualization of Vascular Networks . . . . .	5
2.2 Annotation . . . . .	9
2.3 3D Volumetric Rendering . . . . .	19
2.4 Interactive Devices . . . . .	22
2.5 Haptics Rendering . . . . .	25
<b>3 Haptic Assistance</b>	<b>29</b>
3.1 Haptic Modes . . . . .	30
3.2 Implementation Details . . . . .	32
3.2.1 Background on Proxy-based Volume Haptics . . . . .	32
3.2.2 Marking Image features . . . . .	34
3.2.3 Locating the Proxy . . . . .	36
3.2.4 Returning Force Feedback . . . . .	40
<b>4 System Design and Implementation</b>	<b>42</b>
4.1 Annotation . . . . .	42
4.1.1 Voxel Painting . . . . .	43
4.1.2 Tracing . . . . .	47
4.1.3 Editing Annotations . . . . .	49
4.2 Volumetric Rendering . . . . .	50

4.2.1	Material Properties . . . . .	51
4.2.2	Lighting . . . . .	55
4.2.3	Viewing . . . . .	65
4.3	User Interface . . . . .	70
4.4	System Implementation . . . . .	73
<b>5</b>	<b>Case Studies</b>	<b>76</b>
5.1	Annotation Workflow . . . . .	76
5.2	Example of an Analysis . . . . .	80
5.3	Additional Examples . . . . .	84
5.4	Annotation Experience . . . . .	86
<b>6</b>	<b>User Study</b>	<b>87</b>
6.1	Study Setup . . . . .	88
6.2	Study Results . . . . .	91
6.3	Expert Evaluation . . . . .	97
<b>7</b>	<b>Discussion</b>	<b>99</b>
	<b>Bibliography</b>	<b>104</b>
<b>A</b>	<b>User Interface Windows</b>	<b>129</b>
<b>B</b>	<b>User Study Questionnaire</b>	<b>132</b>
<b>C</b>	<b>Keyboard Shortcuts</b>	<b>137</b>
<b>D</b>	<b>Using SHVR</b>	<b>139</b>
D.0.1	System Requirements . . . . .	140
D.0.2	File Formats . . . . .	140
<b>E</b>	<b>Copyright Forms</b>	<b>142</b>
E.1	Figure 2.2 . . . . .	142

# List of Figures and Illustrations

1.1	A variety of flower heads. The shape of the vascular structure for each of these flower heads is different from each other. . . . .	2
2.1	Illustrations of vascular structures within plants in <i>The Anatomy of Plants</i> by Nehemiah Grew (1682). . . . .	6
2.2	Two different visualizations of the xylem connections. (left) Each vertical line represents a vascular bundle, and each dot represents the location of the connection between vascular bundles. (right) High-resolution computed tomography volume rendering shows the primary xylem bundle (blue), and surrounding tissue (green) from various positions and angles (Brodersen et al., 2012). Permission to republish in a thesis is provided by Copyright Clearance Center, Inc. (“CCC”) on behalf of the Rightsholder Blackwell Publishing Ltd. . . . .	8
2.3	Automatic segmentation pipeline in Montoya-Zegarra et al. (2019). The pipeline uses a combination of automated methods to preprocess, segment, and analyze tubular structures 3D image data. Permission to republish is given by CC BY 4.0 license. . . . .	12
2.4	Progression of region growth segmentation algorithm. (left) Starting from a user-defined seed, (middle) neighbouring voxels are segmented (blue) based on an inclusion criteria. (right) Voxels outside the inclusion criteria (red) are not segmented. “X” marks the current wavefront of voxels being segmented. . . . .	14
2.5	Image analysis pipeline in MorphoGraphX. Cells of <i>A. thaliana</i> flower meristem captured using confocal microscopy is segmented and quantified from 3 different time points. Permission to republish is given by CC BY 4.0 license. . . . .	20
2.6	From left to right. Neurons in the visual cortex of a marmoset monkey labeled with green fluorescent protein are traced by placing the brush inside of the neuron, and then following along its branches while holding the trigger. The trigger is released to stop tracing (Usher et al., 2018). Permission to republish is not required for a thesis, as mentioned in FAQ regarding IEEE Permissions. . . . .	23
2.7	Small dendritic splines are recovered by using the haptic device for navigation and volume image editing. The haptic device provides 6 DOF control over the dataset, and the force feedback helps to place target points on the surface (Corenthy et al., 2014). Permission to republish is not required for a thesis, as mentioned in FAQ regarding IEEE Permissions. . . . .	27

3.1	Example of surface sampling. Surface sampling ignores relatively small objects as the probe moves away from the camera (light blue arrow). When the probe encounters a large object, the proxy (light green circle) is placed where the percentage of relevant sample points exceeds some threshold rate (dotted green circle). Moving the probe deeper into the object returns a spring force toward the proxy (red arrow). The proxy is released when the probe moves towards the camera. . . . .	30
3.2	Example of proximity and object following modes. The two dotted lines indicate the path of the proxy using object following (magenta), and proximity following (blue). The path of the probe is indicated by light blue line below. The difference in haptic feedback can be seen at (a), where a proximity following moves the proxy toward a nearby object, and at (c), where object following cannot move past the current object. The intersection at (b) moves probe away from the centerline for both object and proximity following. . . .	31
3.3	Implementation of relevance thread. It updates the local region around the probe to mark relevant regions. . . . .	35
3.4	For proximity following (a), all sample points that are within relevant regions are relevant. For object following (b), only points within a relevant region connected to the previous relevant region are relevant. . . . .	37
3.5	Calculation of sample points $s_i$ . The offset positions $o_i$ are precomputed from polar coordinates and stored in a list. Then, the offset positions are scaled by the probe size $r_{probe}$ and added to the probe position $\vec{p}_{probe}$ to create sample points. . . . .	38
3.6	Updating relevance volume for object following. Only the relevant region within a local region that is connected to the previous relevant region is updated. A probe (teal circle) is surrounded by a voxel-aligned local region (dark red square). At each time step, a seed (white circle) is produced where the current and previous local relevant region overlaps. This seed is used to apply region growth method in the local region around the probe for haptic assistance. . . . .	39
3.7	Implementation of Haptic Loop. The haptic loop calculates the position of proxy based on the current relevance volume, and returns force to the haptic device. Parameters for relevance thread includes the haptic mode, the haptic threshold value, the local region, and a seed point. . . . .	41
4.1	Segmentation of the vascular structure within a thistle flower head. The upper vascular bundles (green) are labeled with a different colour compared to the lower vascular bundles (red). In addition, the segmented vascular structure is contrasted from rest of the flower head. . . . .	44
4.2	Example of speckle fill. (left) Objects that are not touching the edge of the local region are segmented. Each non-background voxel is marked with a seed index. If a voxel is at the edge of of the local region, then it is marked with "E". If there is no neighbouring seed index to the left, bottom, and back of the voxel, then a new seed index is created. (right) An adjacency matrix keeps track of connection between seed indices. . . . .	45

4.3	Pseudo-code for speckle fill brush. This returns a set of seeds separated from the edge of the local region. These seeds are used for region growth segmentation.	46
4.4	Example of tracing. Vascular bundles within a <i>Gerbera hybrida</i> flower head are traced along the centerline. The vascular structure is classified into three sections. Connectivity between vascular bundles is easily perceived. . . . .	47
4.5	Example sequence of tracing. (a) A node (red) nearby the probe (blue) is highlighted with purple. (b) Selecting the nearby node will change the colour of the highlight to yellow. (c) A new node is created in the same place and with the same size as the probe. The colour group of the new node is from the selected colour group. An internode will connect the selected node to the new node. (d) Deleting the selected node will remove it and all the internodes connected to it. . . . .	48
4.6	(left) The volume coloured by a transfer function. (right) The transfer function widget consists of a histogram of all intensity values within volumetric data, editable control points, and lines that interpolate the colour and opacity in between each successive control point. . . . .	52
4.7	Example of density painting. It increases the intensity values within a local region to view structures that may not be visible, or difficult to perceive with the selected transfer function. . . . .	53
4.8	Example of density painting around the probe. (a) Standard volumetric rendering; (b) Thin shell iso-surface is formed within the local region surrounding the probe based on the paint threshold value. (c) When the region is painted, the intensity value of voxels within the region also increases. . . . .	54
4.9	Cases of the probe interacting with volumetric objects. A persistent silhouette surrounds the probe to help find when it is occluded by other objects. In addition, an arrow appears in the middle of the screen when the probe is out of screen. . . . .	57
4.10	Rendering process in the GPU. (a) The starting position, (b) the direction and length of rays are pre-computed for volumetric rendering. The colours on the textures correspond to the starting position and direction of rays, respectively. (c) Mesh objects are rendered first, followed by (d) volumetric rendering, and then (e) mesh objects that are on top of existing rendering, such as the arrow to find the probe. Finally, (f) the output image is stretched to fit the screen.	59
4.11	Volume rendering uses a combination of 3D image data, segmentation mask data, and ambient occlusion data. These datasets are stored in the GPU, and mapped to colour and opacity based on a transfer function, group colour settings, and ambient occlusion mapping respectively. . . . .	63
4.12	Results from distance colour blending. (a) No blending. (b) Smooth transition from volume colour to background colour. (c) Constant level of blending. (d) Sudden transition from volume colour to background colour. . . . .	64

4.13	Comparisons of rendering with visual enhancements. (a) Without visual enhancements, the structures within volumetric data can be hard to perceive. (b) Ambient occlusion darkens objects that are in close proximity to each other to differentiate between tight gaps and open spaces. (c) Distance colour blending helps differentiate nearby objects from distant objects. (d) Combining ambient occlusion and distance colour blending provides depth cues and contact shadows to help identify structures within volumetric data. . . . .	65
4.14	Moving the volume using the probe. When the “move” button is pressed, the movement of the volume matches the movement of the probe. The movement of the volume can be scaled by scaling the movement of the probe. . . . .	66
4.15	Camera position for each viewing modes. There are three viewing modes: (a) full screen, (b) stereoscopic, and (c) overview mode. The visible region is contained within the near and far clipping plane (red lines) in front of the main camera (green). Overview mode has three orthographic cameras (yellow) located at the top, side, and front of the volume. . . . .	68
4.16	Overview Mode. This provides a view from the top, side, bottom, and main camera. The colour of the lines label each edge of the viewing frustum. Edges that are closer to the camera are drawn solid, while edges further from the camera are drawn stippled. . . . .	69
4.17	Setup for SHVR. Users use a combination of Touch <sup>TM</sup> haptic device (3D Systems, n.d.), a keyboard, and a mouse to interact with SHVR. They also have the option to use the Playstation 3D Display for stereoscopic viewing. .	71
4.18	The main window consists of (a) a menu bar, (b) a graphics window, and (c) a status bar. . . . .	72
4.19	Commonly used keyboard shortcuts. The keys are grouped by similarity in functionality. Permission to use the base keyboard layout is given by CC BY-SA 3.0 license. . . . .	73
4.20	Organization of threads. The arrows show the flow of data from one thread to another. If a line does not have an arrow, that indicates information flows bidirectionally. . . . .	74
5.1	Contents of a CT-scanned flower head. (a) A flower head shows visible epidermal, vascular, and ground tissue. (b) A vascular bundle is comprised of xylem and phloem tissues that appear separated as two separate tubular structures. (c) Underdeveloped vascular bundles appear fragmented and noisy. . . . .	77
5.2	Different views of <i>Gerbera hybrida</i> . (a) A photograph of a mature flower head. (b) Top view shows the florets arranged with visible parastichies. (c) Vascular structure is divided into three sections: green represents the bract, abaxial and stem veins, red represents the adaxial veins and orange represents the florets and floret veins. . . . .	80

5.3	Tracings of <i>Gerbera hybrida</i> . (a) Early stage shows only the abaxial and bract veins. (b) Floret veins start emerging from florets at the outer rims of a flower head. Adaxial veins emerge separately from all other veins. (c) Floret veins connect with nearby abaxial veins. More florets emerge toward the center of a flower head. Abaxial veins extend towards both the outer rims and the abaxial veins. (d) Floret veins closer to the center connect with nearby adaxial veins.	82
5.4	Comparison of adaxial veins (red) between different flower heads. (a) <i>Gerbera hybrida</i> has straight adaxial veins running radially from the center. (b) <i>Helianthus annuus</i> has drooping adaxial veins that connects with other nearby adaxial veins. (c) <i>Bellis perennis</i> has adaxial veins arranged in a regular reticulate pattern.	83
5.5	Tracing of roots in a heterogenous mixture of soil, rocks and other organic matter.	84
5.6	Segmentation of the vascular structure in <i>Arabidopsis thaliana</i> .	85
6.1	User study timeline. Participants were given instructions on how to use SHVR (yellow) before being asked to annotate a small section of a flower head (orange). The study took roughly 70 minutes per participant.	88
6.2	Section of <i>Gerbera hybrida</i> flower head. (left) Approximate location of where the section was cropped from. (right) Rendered section of <i>Gerbera hybrida</i> flower head used during annotation tasks.	89
6.3	Ground truth annotations. (left) Partial segmentation. (middle) Full segmentation. (right) Partial tracing.	90
6.4	Information about participants. Most participants were familiar with haptics and 3D graphics, but not with image annotation. Box plots show how the responses are distributed, divided into quartiles. Outliers (gray circle) indicates that a data point differs significantly from the rest of the dataset. Mean value (yellow diamond) is the average value of all the data points for a response.	91
6.5	(left) Participant feedback on navigation. Navigation was generally considered to be intuitive. (right) Distribution of used viewing modes while annotating. Red represents painting, while green represents tracing. Users preferred to use stereoscopic viewing mode over full screen and overview viewing modes.	92
6.6	Mistakes during painting. (a) Painting stops early before the end of the vascular bundle. (b) Receptacle tissue is painted, which is not a part of the vascular structure. (c) Parts of a vascular bundle are not painted. (d) Adaxial vein is not painted.	94
6.7	Comparison of tracing results between participants. (a) Section of flower head viewed from the bottom. (b) Result from Participant 6. (c) Result from Participant 8. (d) Result from Participant 10. The output tracing closely resemble the vascular structure within the sample.	95
6.8	Responses for haptic feedback. (left) Users found that painting was much easier to learn compared to tracing. Haptic feedback was useful for both annotation tasks. (right) Users preferred “Object Following” haptic mode over all other haptic modes.	96

A.1	(a) Ambient Occlusion Editor adjusts the settings for the ambient occlusion and lighting. (b) Transfer Function Editor adjusts the opacity and colour of the volume at different intensity values, haptics and painting threshold, and offset value for density painting. . . . .	129
A.2	Group Settings Widget edits the various colour groups. It is used to control the colour and opacity of annotation. It can also be used to adjust the opacity of the volume, and the colour of the background. . . . .	130
A.3	(a) Probe Settings Widget adjusts the appearance of the probe, workspace scale, painting settings and haptic settings. (b) Render Quality Settings Widget adjusts the performance of the program. (c) Visualization Settings Widget adjusts the position of cutting and clipping planes, the viewing mode, and other visual assistances. . . . .	131
B.1	Page 1. . . . .	132
B.2	Page 2. . . . .	133
B.3	Page 3. . . . .	134
B.4	Page 4. . . . .	135
B.5	Page 5. . . . .	136
C.1	Page 1. . . . .	137
C.2	Page 2. . . . .	138
D.1	Load File Widget. . . . .	139

# List of Tables

4.1	List of commands for tracing. All commands have an execute, undo, and redo component. . . . .	50
6.1	Accuracy of annotation results. From left to right, the columns stands for: participant ID, Dice score compared with partial ground truth segmentation, Dice score compared with full ground truth segmentation, tracing score for nodes, number of nodes placed outside of the region of interest out of the total number of nodes, tracing score for internodes, number of internodes placed outside of the region of interest out of the total number of internodes, number of segments traced, and number of topological error. The outputs from the ground truth annotations are included to compare with the participant scores.	93

# Glossary

**clipping planes** Planes in front of the camera. There are two clipping planes: near and far. The parts of volume beyond the far clipping plane, and before the near clipping plane are hidden from view.. 72

**colour group** An index that maps to a colour and opacity. This is used to label different groups during annotation.. x, 43, 48–50, 55, 56, 61, 72, 100–102

**direct volumetric rendering** Rendering technique that applies lighting and material properties, such as colour and opacity, to objects. This can be viewed from arbitrary camera position and angle.. 51

**gradient** The first derivative of image data that indicates the direction of greatest change.. 33

**image data** Information that is stored as a grid of values. For CT scans, each grid cell contains a scalar intensity value, which informs how much X-ray signal was absorbed by the sample at that location. As more X-ray signal is absorbed, the intensity value becomes greater.. 13, 15, 43, 54, 55, 62, 76, 99

**image feature** Information about image that can be derived from data values in image data. The simplest value from a CT scan is signal intensity value. Other potential information includes shape, size, texture, homogeneity, boundary surrounding a structure, connectivity within a structure, and proximity to surrounding structures.. 11, 14, 34, 51

**local region** A region surrounding the probe with size of the region bound to the size of the visible probe. This is represented as a region in volumetric data.. x, 39, 43, 54, 75

**mesh object** An object made out of triangular mesh. This includes the probe, nodes, and internodes.. 58

**painting** Segmentation within the local region of a point of interaction.. 74

**probe** A virtual point of interaction. The position and rotation of the probe is mapped from the haptic device.. x, 54, 70, 72

**region growth** Region growth algorithm starts from an initial seed point, and then examines if the neighbouring voxels should be added to the region based on some inclusion criteria, such as a minimum intensity threshold value, or distance from initial seed point. If the criteria is met, the process repeats from the neighbouring voxel.. ix, x, 34, 39, 44–47

**sample** A biological specimen. In this thesis, this refers to the scanned flower head and all of its components, including its vascular structures, florets, bracts, and ground tissues.. 9, 36, 42, 65, 70, 78, 86, 89, 92, 97, 99, 100, 102, 103

**segmentation** Labeling and delineating the voxels within image data to indicate a single unifying region or object.. 43

**SHVR** The system implemented in this thesis for guiding annotation of volumetric data using haptic feedback. The name is an acronym for "Segmentation using Haptics and Volumetric Rendering". Note that this system evolved to include tracing in addition to segmentation.. 99, 139

**topology** Study of properties of space that are invariant under any continuous deformation. This means that objects within the space can be freely stretched and contracted, but not broken. Objects may be differentiated by the number of connected objects, cavities, and holes. . 93

**tracing** Morphological reconstruction of image data. For tubular structures, reconstruction is either a network of nodes and internodes, or a series of voxels adjacent to each other along the medial axis.. 43, 47

**transfer function** A function that maps an output for each potential input. In the case of direct volumetric rendering, the intensity value of volumetric data is often mapped to colour and opacity.. 44, 56, 79, 85

**volume** An object that contains volumetric data.. x, xi, 21, 32–34, 42, 51, 52, 55–58, 63, 65–70, 72, 97, 100, 102

**volumetric data** Three-dimensional image data. Series of 2D image slices are captured at regular depth intervals.. x, 32, 34, 42, 49, 51, 52, 75, 94, 100, 101

**volumetric object** A connected region within volumetric data.. 3, 32, 42, 55, 57

**workspace** The region on the haptic device where the stylus can move in. This is the limit of how much the stylus can move in physical space.. 72

# Chapter 1

## Introduction

The vascular structure serves an important role in the development of a plant by delivering water and nutrients to its organs through a network of vascular bundles. However, unlike the vascular structure of leaves that is visible with the naked eyes, the vascular structure of a flower head is much less visible due to other plant structures covering the complex three-dimensional vascular structure underneath. As a result, the role of the vascular structure within a flower head is not well understood (Figure 1.1). Studying the development of the vascular structure may result in knowledge to create a model to verify existing understandings and simulate different possibilities with ease (Meijering, 2010).

One method of analyzing the vascular structure is by using an X-ray micro-computed tomography scan, or micro-CT, of a plant. Micro-CT captures the amount of X-ray absorbed at different regions of the sample non-destructively. This is created through computational reconstruction using a set of radiographs — 2D X-ray projections that capture the amount of attenuation of X-ray beam passing through a sample — around a sample to recover the spatial distribution of attenuation coefficient and produce 3D volumetric data (Withers et al., 2021). Unlike photon and electron-based tomography that can penetrate only a few layers



Figure 1.1: A variety of flower heads. The shape of the vascular structure for each of these flower heads is different from each other.

of cells, X-ray can penetrate through large and intact plant structures (Duncan et al., 2022).

Scanning time and radiation exposed to the sample may be reduced by taking fewer raw tomographic image slices, and then intelligently filling in the missing information using an image reconstruction algorithm (National Institute of Biomedical Imaging and Bioengineering).

Various software tools have been developed to visualize, interact and analyze the volumetric data. These tools include Fiji (Schindelin et al., 2012), MorphoGraphX (Barbier de Reuille et al., 2015)(Strauss et al., 2021), MeVisLab (Ritter et al., 2011), Vaa3D (Peng et al., 2014a), Amira (Stalling et al., 2005), and Dragonfly (Piche et al., 2017). Through these tools, the user's perception may be amplified by enhancing image features, removing noise, annotating

structures of interest, rendering the data, or measuring some property of the data such as the volume of a structure (Ritter et al., 2011). These can help to interpret the sample more accurately and meaningfully during analysis. However, interpreting the various structures can still be challenging due to insufficient contrast between structures, low signal-to-noise ratio, and relatively small structures. This may lead to high error rate, or highly subjective conclusion during analysis (Preim and Botha, 2014).

The objective of this thesis is to develop a system to interactively guide the annotation of 3D volumetric image data using volumetric and haptic rendering techniques. The contribution of this thesis is as follows:

1. A system named SHVR, or "Segmentation using Haptics and Volumetric Rendering", to assist the user in annotating volumetric data using a haptics device;
2. A robust haptics algorithm to touch the surface of volumetric objects and to follow the centerline of a tubular structures without relying on surface normals; and
3. A painting algorithm to quickly segment numerous volumetric objects.

To demonstrate the usefulness of SHVR, several flower heads were annotated to analyze their vascular structure. In addition, a user study was conducted to understand the usefulness of SHVR by annotating a small portion of a flower head.

This thesis is divided into seven chapters. First, information relevant to this research is presented in the "Background" chapter. It details the history of visualization of biological samples and interaction with volumetric data. Then, the technical details of the system are introduced. "Haptic Assistance" describes the haptic algorithms used to aid interaction, while "System Design" details the motivations and implementation behind the features in SHVR. To validate the usefulness of system, "Case Studies" showcases the results from annotating several flower heads, and "User Study" presents the result from participants eval-

uating SHVR. Finally, results from evaluation are summarized, and potential improvements to the system are suggested in “Discussions”. Any additional details are included in the “Appendix” chapter.

# Chapter 2

## Background

### 2.1 History of Visualization of Vascular Networks

Visualization of the internal structures of plants has a long history. Starting from the 17th century, Nehemiah Grew produced detailed illustrations of the vascular structure in plant leaves and roots (Grew, 1682) (Brodersen and Roddy, 2016) (Figure 2.1). Illustrations render what the illustrator observes by emphasizing regions of interest while omitting irrelevant details for clarity. However, skilled illustrators were required to create spatially accurate illustrations, which were time-consuming. In addition, the contents of illustrations were subjective based on the illustrator's interpretation.

Advances in photography made recording visual data more accessible. Images that were grounded to reality became faster to capture, and contained more detail, which made studying the internal structures more accessible compared to creating detailed illustrations. In addition, molecular markers have been used to show and track differentiation of specific cells. The development of vascular pattern can be tracked by photographing the expression of molecular markers of procambium — cells that differentiate into vascular bundles — over

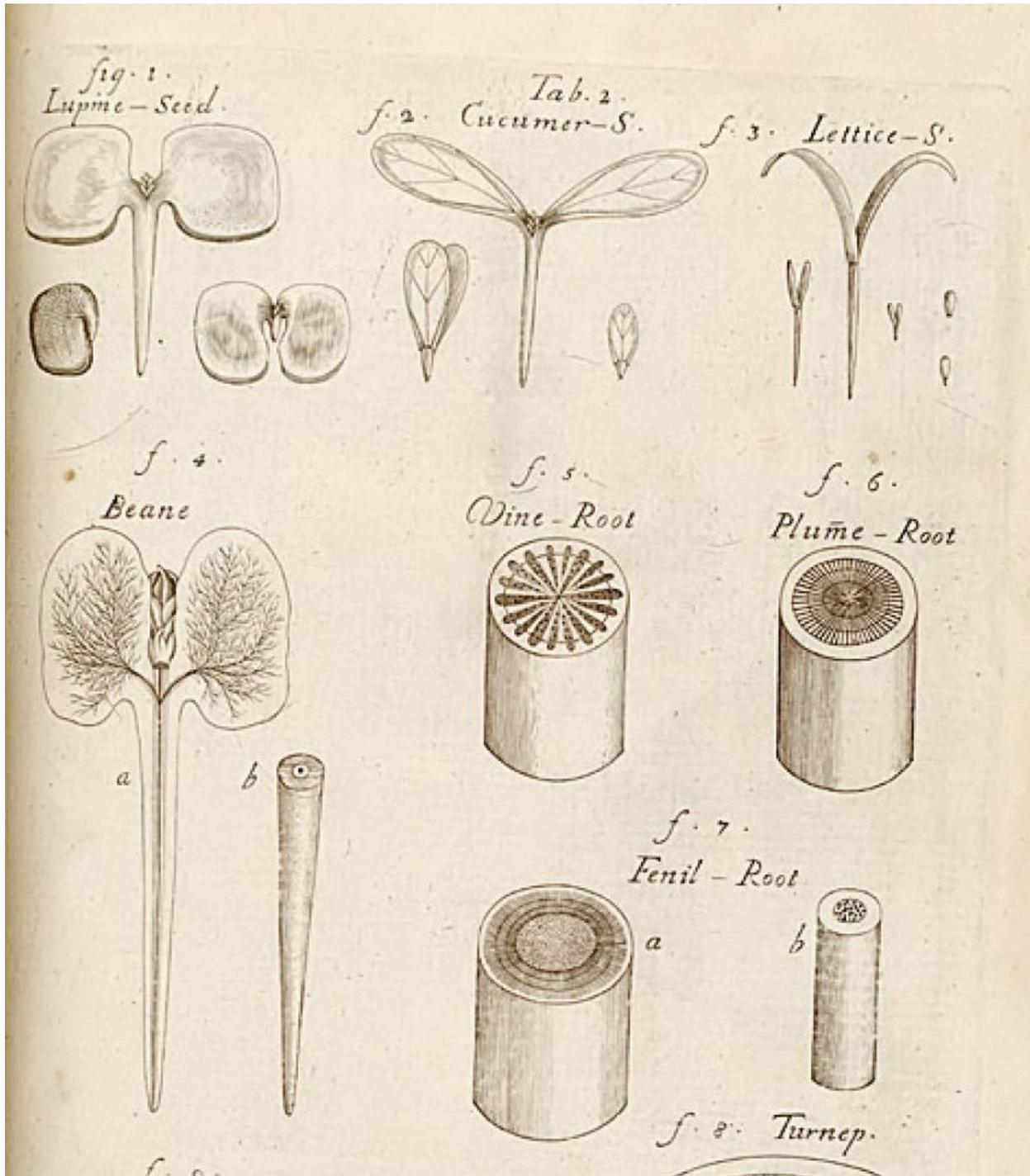


Figure 2.1: Illustrations of vascular structures within plants in *The Anatomy of Plants* by Nehemiah Grew (1682).

time (Dengler and Kang, 2001) (Scarpella et al., 2006) (Gardiner et al., 2010). Collecting and analyzing a large number of samples have also been automated, such as the vascular structure of leaves (Bühler et al., 2015), and early development of roots (Betegón-Putze et al., 2018). However, photographs tend to show both relevant and irrelevant details, which complicates interpreting the contents of a photograph. As a result, photographs did not replace illustrations but complemented them to convey ideas more precisely. More specifically, diagrams provided a simplified representation of the claims, while photographs provided actual examples to support the interpretation.

Although 2D images can lead to useful insights, spatial information cannot be accurately captured as a plant is inherently three-dimensional (Duncan et al., 2022). An early work to study the 3D vascular structure required slicing a plant into thin sections using a microtome, aligning each slice based on distinct features, and then manually annotating the path of several vascular bundles. Thin sections were used to analyze the vascular bundles in the shoots of an *Arabidopsis thaliana* plant (Kang et al., 2003) and the vascular connections between each floral organ to the vascular tissues in a *Helianthus annuus* flower head (Alkio and Grimm, 2003). However, this process was time-consuming, effortful and destructive to the original sample. Instead, a sample can be scanned to non-destructively capture a 3D image of the sample while preserving the spatial information of internal structures. This includes magnetic resonance imaging (MRI) to detect resonance due to magnetic fields (Moriwaki et al., 2014), X-ray computed tomography (CT) to detect the amount of x-ray absorbed by a sample (Dhondt et al., 2010) (Karahara et al., 2015), optical projection tomography to detect either transmitted or emissive light (Sharpe, 2004), and confocal microscopy to detect a small, focused beam of light emitted or reflected from a sample (Webb, 1996). The scans, captured at regular vertical distances, are then loaded into an image analysis software for analysis.

Studies showed that 3D scanned images complements insight made from other imaging tech-

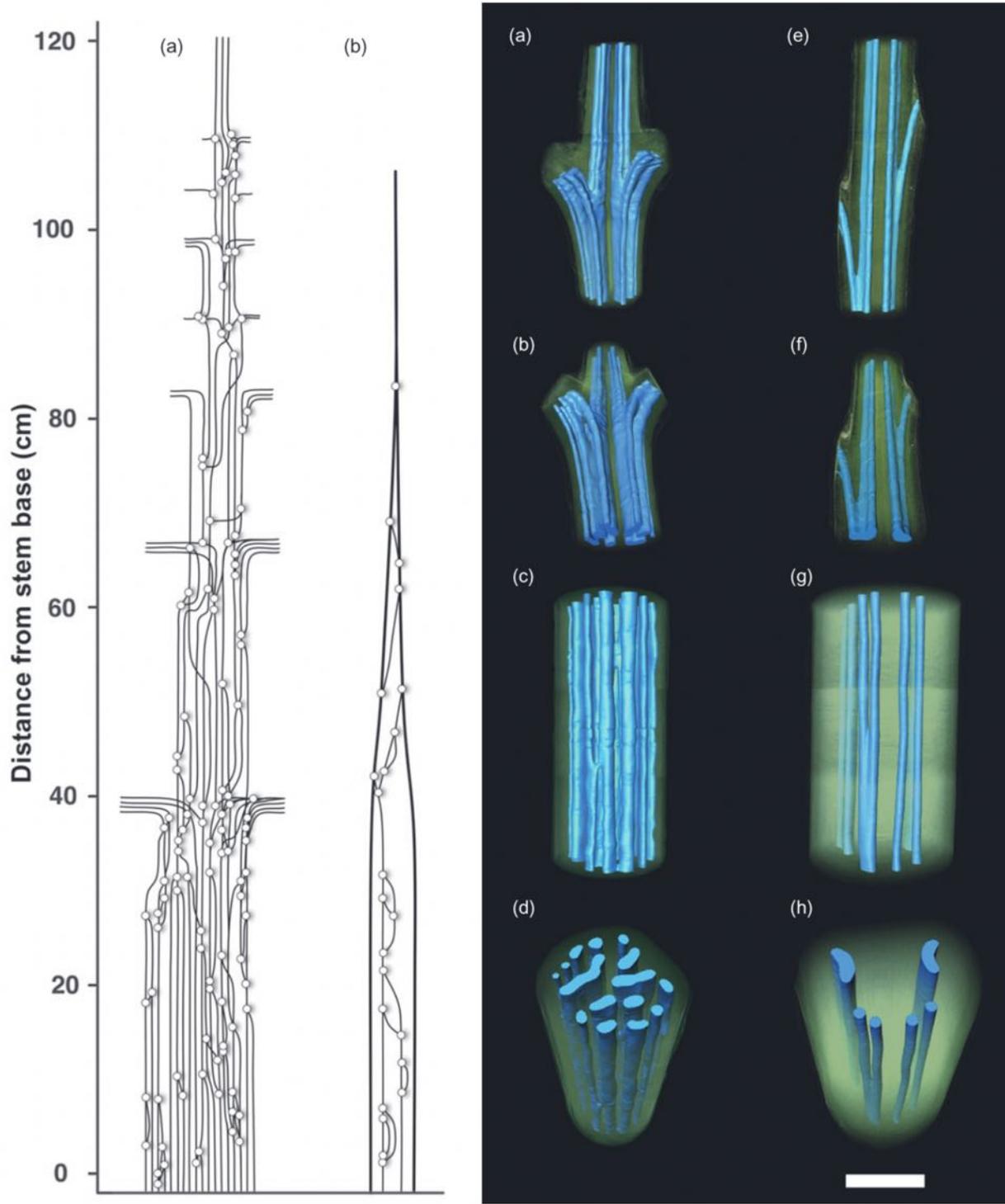


Figure 2.2: Two different visualizations of the xylem connections. (left) Each vertical line represents a vascular bundle, and each dot represents the location of the connection between vascular bundles. (right) High-resolution computed tomography volume rendering shows the primary xylem bundle (blue), and surrounding tissue (green) from various positions and angles (Brodersen et al., 2012). Permission to republish in a thesis is provided by Copyright Clearance Center, Inc. (“CCC”) on behalf of the Rightsholder Blackwell Publishing Ltd.

niques. Brodersen et al. (2012) studied gas exchange and cavity resistance in ferns by capturing the xylem connections in three different ways: as thin sections of stained samples, a network map, and CT scan of dehydrated fern samples (Figure 2.2). The CT scan confirmed the results from dissected vascular bundles, while preserving the spatial trajectory of the bundles *in situ*. Gee (2013) compared thin sections of a silicified conifer seed cone with a micro-CT scan. They liked the ease, speed, and non-destructiveness of the micro-CT scan but noticed the lack of differentiation of some plant tissues in the micro-CT scan that are visible in the thin sections. 3D visualization may also provide additional details that were missed using 2D visualizations (Duncan et al., 2022), such as the vascular structure surrounding soybean nodules (Livingston et al., 2019).

Vascular structure may be enhanced before scanning to show higher contrast from the surrounding structures. For CT scans, samples may be prepared by adding contrasting agents to a sample. Contrasting agents work by adding particles that highly attenuate X-ray such that it shows up as high intensity signals in the scan, such as gold nanoparticles. Although application of gold nanoparticles takes less than an hour, the particles must be applied *in vivo*, which can be difficult to time the stage of growth before imaging (Mahan and Doiron, 2018). In addition, the nanoparticles cannot help detect developing vascular tissues that do not yet transport water and nutrients.

## 2.2 Annotation

Image analysis is the process of extracting desired information from image data. A typical workflow is divided into preprocessing, annotation, visualization, and measurement (Ritter et al., 2011). First, preprocessing removes undesirable distortions, such as noise and background inhomogeneity, and enhances features of a sample. Second, annotation identifies

some anatomic structure of interest <sup>1</sup>. Third, visualization renders the image data and annotation. Lastly, measurement quantifies the annotated data for further analysis. Because the morphology — the structure and form — of a plant’s vascular system can be diverse even within the same species, annotation is a prerequisite to gaining insight into the vascular structure (Preim and Botha, 2014).

Annotations are represented as either segmentation — labeled <sup>2</sup> voxels within a structure of interest (Preim and Botha, 2014) — or tracing — a morphological reconstruction from image data (Meijering, 2010). Tracing of a tubular structure is represented as either a network of nodes and internodes, or a series of voxels adjacent to each other along the centerline.

Annotation methods are categorized based on the amount of user interaction involved in annotation. However, the terminology for the categorization has been fluid. Olabarriaga and Smeulders (2001) divides segmentation methods into computational and human component. Computers delineate a region of interest guided by user input, or a priori knowledge in the algorithm. User input include pictorial input (e.g. points, lines and regions), parameters (e.g. sliders and dials), and menu options (e.g. menu buttons, and accepting or rejecting results). Then, users respond to the output feedback by interacting with the results using the user interface, such as editing the output annotation. Acciai et al. (2016) categorizes tracing methods as global processing, which applies an annotation algorithm to the whole image, and local processing, which applies an annotation algorithm only around relevant structures. In addition, meta-algorithm approaches categorize algorithms that enhance some existing method by reducing computation workload, or reducing impact of image variability. Zhao and Xie (2013) distinguishes annotation methods as supervised, automatic, and interactive methods. They can also be categorized as manual, automatic and semi-automatic,

---

<sup>1</sup>Originally, the term “segmentation” was used for this definition. But as terminologies became more specific to refer to different types of output, “annotation” now refers to all types of labels that identify a structure of interest (Labelbox, n.d.) (Eliceiri et al., 2012)

<sup>2</sup>“Label” refers to a piece of information attached to each voxel. This is to contrast one voxel from another voxels with different label. This differs from “marked”, which implies a binary label of whether a label is attached to a voxel or not.

respectively (Lesage et al., 2009) (Preim and Botha, 2014). For consistency sake, the terms manual, automatic, and interactive will be used to describe annotation methods.

Manual annotation methods use pictorial input on the image data to identify a structure of interest (Preim and Botha, 2014). This may involve outlining region of interest, creating a path along the centerline of a tubular structure, or painting a set of voxels within a region. Berger et al. (2018) manually segments an object on a 2D image slice by filling closed contours as they are drawn, or paint on voxels using a mouse. This process can be assisted by constraining the segmentation within the boundaries of an object. If the result is obtained by an expert, then it can be used as a gold standard to evaluate automated annotation methods. In Gillette et al. (2011), the quality of automated tracing algorithms was evaluated using manually traced gold standard generated by several experts. Manual annotations provide robustness from artifacts, low contrast and unexpected shapes. However, they are also time-consuming and effortful to produce. In contrast, automatic annotation methods do not use pictorial inputs. Instead, structures of interest are identified by combining image features, such as signal intensity, connectivity, size, shape, texture and proximity to nearby structures, with a priori knowledge about target structures. Users modify the output annotation by tuning parameters and selecting menu options (Calhoun et al., 1999) (Lesage et al., 2009).

An automated annotation method organizes several simpler annotation methods in a pipeline to produce a desired output. For example, Autotube segments tubular blood and lymphatic vessels by using a combination of thresholding and vesselness filter (Montoya-Zegarra et al., 2019) (Figure 2.3). Thresholding separates voxels based on a range of intensity values (Otsu, 1979), while a vesselness filter determines the likelihood that a voxel belongs to a tubular structure. Analyzing the eigenvalue of the second derivative of a Gaussian kernel provides a measure of geometric similarity, which calculates the likeliness of a region being a part of a tubular structure (Frangi et al., 1998). Another standard method is watershed segmentation, which segments a region based on an analogy of a topographical landscape of ridges and

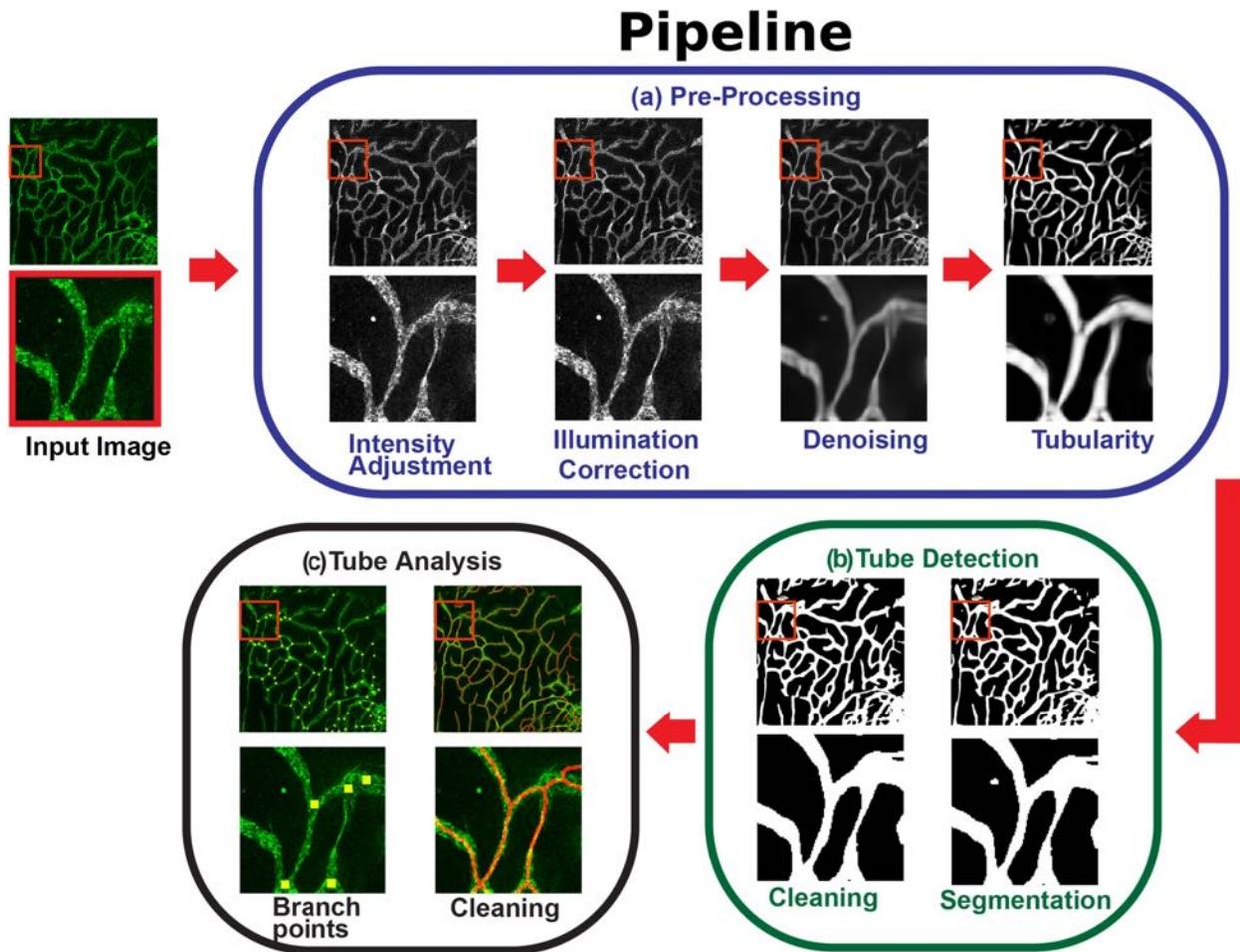


Figure 2.3: Automatic segmentation pipeline in Montoya-Zegarra et al. (2019). The pipeline uses a combination of automated methods to preprocess, segment, and analyze tubular structures 3D image data. Permission to republish is given by CC BY 4.0 license.

basins. Each region is a filled catchment basin with minima at the lowest point within the basin and ridges that separate the basins. The region fills up to the ridge of the basin, starting from the minima of each basin. The level height, measured from the bottom of each basin, is controlled by the user through a parameter. When the level exceeds the height of the lowest ridge, it merges with the neighbouring basin that shares the ridge into one larger region (Vincent and Soille, 1991).

Automated annotation methods may significantly lessen the workload for the user and save a considerable amount of time. However, it may also be more effortful than manual an-

notation due to the effort of searching for errors and correcting them (Peng et al., 2011). For samples not treated with contrasting agents, annotation requires extensive effort from the user (Verboven et al., 2015). It is worth noting that no annotation method is genuinely automatic, where human intervention is not needed to create reliable annotation of a target structure, other than for simple structures (Acciai et al., 2016). Lesage et al. (2009) recommends tailoring the annotation method to reach reasonable levels of practical efficiency for completing a task.

Interactive annotation methods use pictorial inputs and automation to identify a structure of interest, which combine the accuracy and speed of automatic annotation methods with the robustness of manual annotation methods. Licklider (1960) suggested humans provide criteria for success and handle infrequent scenarios, and computers execute routine, repetitive work and suggest possible courses of action. This is because humans are better at identifying a large variety of structures from knowing more about the structures of interest more than what can be perceived in the image data, while computers are better at performing redundant tasks on predictable structures with higher speed and accuracy (McGuinness and O'Connor, 2010). Using interactive annotation methods ease and speed up reviewing tasks, reduce the amount of interaction, and lower inter-operator variability — the difference in annotation between several people (Lesage et al., 2009). In addition, points placed directly on image data, known as seeds, help filter out unlikely regions within an image to make interactive methods more compact and more robust to signal variability (Acciai et al., 2016).

A variety of algorithms exist for interactive segmentation. Region growth method starts from an initial seed point, and then examines if the neighbouring voxels should be added to the region based on some inclusion criteria, such as a minimum intensity threshold value, based on geometric features, such as the shape, size, and location of the structure of interest. If the examined voxel is considered part of the same region, it is segmented, and the algorithm recursively examines its neighbouring voxel until no more voxels are being segmented.

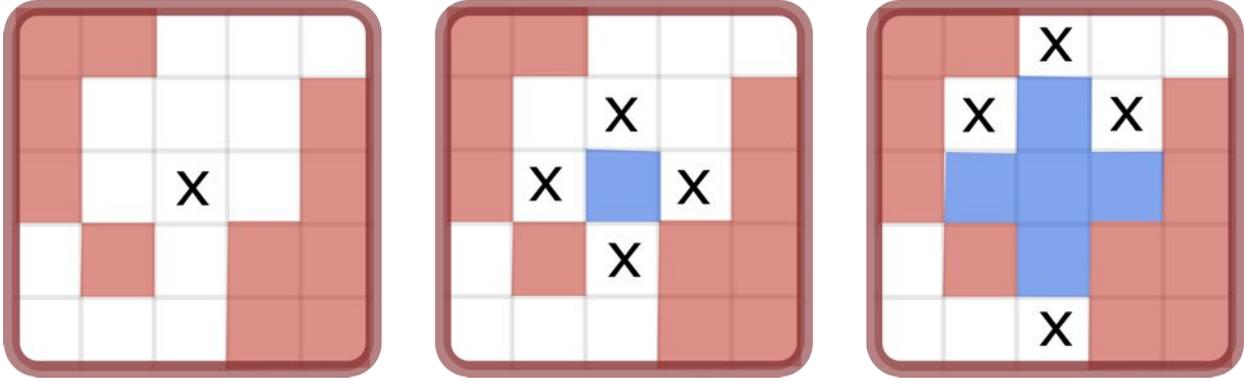


Figure 2.4: Progression of region growth segmentation algorithm. (left) Starting from a user-defined seed, (middle) neighbouring voxels are segmented (blue) based on an inclusion criteria. (right) Voxels outside the inclusion criteria (red) are not segmented. “X” marks the current wavefront of voxels being segmented.

(Law and Heng, 2000) (Figure 2.4). Pohle and Toennies (2001) expands on region growth method by learning a homogeneity criterion to automatically threshold a region of interest. This criterion finds the mean and distribution of intensity within the region of interest by sampling the region around the seed through random walks, and then terminating if the walk encounters a large variation in intensity value.

Features in image data may be given more weight by reconstructing the dataset into a graph structure. The graph consists of nodes that contain voxel intensity values, and weighted edges that connect a node to its neighbouring nodes. The weight represents the cost of using the path to annotate based on image features. By minimizing the cost, the algorithm may return a path that avoids crossing object boundary, or follows along the centerline of a tubular structure. For example, graph cut segmentation dynamically assigns object of interest using interactively placed object and background seeds to provide clue about what is and what is not a part of the object of interest, respectively. A minimum cost path around an object is found to separate the object from the background (Boykov and Jolly, 2000). Random walker segmentation assigns each voxel to a seed with the minimum cost path, which allows segmenting regions where the boundary between object and background is weak but continuous, such as a thin dividing line between organs (Grady, 2006). Live-wire

segmentation uses the order by which seeds are placed to guide the segmentation. It creates a boundary guided by image features and user-generated seeds that snaps and wraps around an object of interest. Since each new seed freezes the boundary segment from the previous seed point, the user can iteratively choose the boundary segment that best fits the region of interest until the boundary is complete (Barrett and Mortensen, 1997)(Malmberg et al., 2006).

The boundary of an object may be annotated without relying on the voxel grid of image data using splines — continuous piecewise parametric curves made from discrete points. Active contour is an energy-minimizing spline that iteratively surrounds a region of interest based on external forces that pushes the spline toward certain image features, and internal forces that smoothens the curvature of the spline. By placing a contour roughly around an object of interest, the spline iteratively snaps around the boundary of an object (Kass et al., 1988). Bresson et al. (2007) extends active contour model by incorporating image denoising — removing noise from image while retaining image features — to globally segment homogeneous region within image data.

Tracing algorithms expand on the segmentation algorithms by reconstructing the segmented region of interest as a skeleton, which depicts the elongated shape and the connectivity of vessel structures while preserving the topological properties, such as the number of connected objects, cavities, and holes, and geometric properties, such as the size and shape. Note that tracing shares many similarities with segmentation in their implementation and method of interaction. Thinning algorithm globally thins out a voxel region iteratively until the centerline of a structure remains (Lee et al., 1994). A minimum spanning tree initially constructs an overly reconstructed tracing by finding a path from each foreground pixel to a given seed. Then, the paths connecting two branching nodes are sorted by length, and pruned until the desired skeleton remains (Xiao and Peng, 2013) (Yang et al., 2019). Li et al. (2019) trace individual neurons from a pre-traced neuron cluster by categorizing based on

how much the neurites — projections from the cell body of a neuron — deviate in orientation from a given soma — bulbous cell body of a neuron. However, measuring the accuracy is subjective since the evaluation does not take into consideration about the quality of skeleton, such as the number of spurious branches, number of missing true branches, preservation of sharp corners, and smoothness of curve under different conditions of noise, resolution, and rotation (Saha et al., 2016).

Interactive tracing methods use a combination of image features and user input to guide the output reconstruction of the centerline. A method may pre-compute the possible shortest paths beforehand to maintain an interactive rate while placing seeds dynamically. 3D-intelligent scissor finds a minimum cost path between two voxels by using a combination of the image gradient and the Laplacian zero-crossing (Heng et al., 2001). Abeysinghe and Ju (2009) extends 3D-intelligent scissors by allowing user to scribble along a desired path on screen. An initial path is generated by placing end points of a path. Then, the scribbled path updates the existing curve through the cost function, which prefers finding the shortest path through spatial points in high-intensity regions, aligned with the direction of the skeleton curve and nearby user-generated scribble. Live-vessel extends the framework for live-wire (Barrett and Mortensen, 1997) to guide tracing along the centerline of vessels instead of their boundaries. This uses a combination of vesselness filter, direction of skeletal curve, edge detection, and radius to provide robustness against noise and poor image quality, and to smoothen the curve (Poon et al., 2007). For reviews on tracing algorithms, see (Meijering, 2010), (Saha et al., 2016), and (Acciai et al., 2016).

To automate results while using user input for guidance, annotated results may be used as training data for supervised learning. Supervised learning can automate annotation tasks by learning from already annotated samples. In particular, a neural network consists of nodes with weights and biases that creates an output based on the input data and the knowledge incorporated within the network of nodes. For segmentation, it organizes its nodes into layers

to extract image features at progressively higher level of abstraction. This infers a classifier that assigns voxels to the foreground or the background from unseen image data. Deep neural networks — neural networks with multiple layers in between the input and output nodes — has been shown to produce results comparable to manual segmentation by experts (Seo et al., 2020). Convolutional neural network (CNN) captures the local and global image features in image data by organizing layers of abstraction in hierarchical manner. Each layer of abstraction is progressively smaller in resolution, where image features are extracted using a 3D convolution filter. Then, segmentation at higher level of abstraction is assembled by applying a neural network that predicts the likelihood of a voxel belonging to the foreground or the background to the extracted image features (Milletari et al., 2016). Kuan et al. (2020) applied CNN to segment neurons in volumetric image data. 3D U-net expands on CNN by only requiring few manually annotated slices by relying on repetition of common structures within a sample (Çiçek et al., 2016). Dragonfly is a generalized machine learning platform that develops, trains, and applies various classifiers to segment volumetric images. By providing the relevant training data, the classifier can quickly output segmentation for similar samples (Piche et al., 2017).

The main limitation of deep neural networks is that their results are highly dependent on finding accessible, high-quality training data (Seo et al., 2020). The amount of training data used is strongly correlated with the accuracy of results. Although there are methods to increase the amount such as data augmentation — transformation of data by randomly translating, flipping, and rotating —, a large amount of manually annotated data is still desired. In addition, the results are highly sensitive to small changes to hyperparameters — parameters that need to be manually set before the learning process. Since providing sufficient number of training data is an issue, playsourcing — gamification of tasks — can produce more user-generated annotation by stimulating engagement from users. By converting the annotation task into a game where participants aim to maximize their points, the output can be used to yield plausible annotation. However, designing a game that users find

fun while producing useful annotation is a challenging balance to achieve (Hennersperger and Baust, 2017) (Huang and Hamarneh, 2017).

Applications use a combination of segmentation and tracing methods to annotate a structure of interest. For example, Herremans et al. (2015) tracked the growth of vascular bundles in an apple fruit by using a combination of thresholding, morphological filters, and skeletonization. Morphological filters operate on the shape and form of the image data. Two common operations are dilation, which includes voxels neighbouring around the boundary of objects, and erosion, which removes those voxels instead (MathWorks). Region-growing method was also applied to interactively segment the vascular bundles. Brodersen et al. (2011) traced a network of grapevine vessels by applying threshold, region growth, and skeletonization. In addition, intervessel connections were added based on distance between neighbouring vessels. Moriwaki et al. (2014) segmented the vascular structure of fruits by applying region growth method. Because fruit seeds and vascular bundles shared similar intensity values, the fruit seeds were manually removed from the initial segmentation. Bühler et al. (2015) analyzed veins from 2D images of leaves by thresholding the dense veins from the lighter lamina of a leaf, and then traced the morphology of the vascular structure using skeletonization. Verboven et al. (2015) manually segmented cell volumes within tomato leaves by using the wrapping tool in *Avizo*<sup>3</sup>. MyRoot traced the primary root length in *Arabidopsis thaliana* seedlings from the bottom of the root to the hypocotyl — the region between the shoot and the root — from a photograph. Because the roots grow vertically, the white roots separated by the dark soil were automatically traced from the bottom of the root by moving up until the hypocotyl was found. The software was trained to detect hypocotyl by providing thousands of positive and negative examples (Betegón-Putze et al., 2018).

---

<sup>3</sup>Documentation for *Avizo* does not mention what the "Wrapping Tool" does. However, it likely functions similarly to live wire method, or active contour method.

## 2.3 3D Volumetric Rendering

When the user visualizes volumetric image data as a stack of 2D image slices, they can view the sample from above the sample on a transverse plane or from the side on a longitudinal plane, which allows precise viewing and selection of voxels. However, this method of visualization is mentally taxing to visualize the three-dimensional form of a structure and to retain it in memory (Bernardo and Evins, 2018).

By assigning depth to each image slice, volumetric data can be rendered as a collection of 3D volumetric objects. Direct volumetric rendering applies lighting and material properties to objects and allows viewing from arbitrary camera position and angle, which enhances depth cues, increases the realism of the rendering, and better conveys the spatial relationship between objects. Ray-casting simulates light by casting rays from each screen pixel to render parts of the scene that are in view (Hadwiger et al., 2009). Another potential method to visualize volumetric data is maximum intensity projection, which displays the maximum intensity value sampled from ray-casting (Calhoun et al., 1999). However, this visualization method does not preserve depth during rendering, making spatial information challenging to perceive.

Polygon data may be rendered alongside volumetric data. Yong Wan et al. (2009) uses polygon data to cull regions that obscure the structures of interest by rendering the volumetric data up to the surface of the mesh, and then resume rendering after the last layer of mesh. This is used to remove the noise in between mesh layers to better reveal the spatial relationship between objects.

Several additions can be made to improve the visualization. Transfer function assigns optical properties, such as colour and opacity, to voxels in image data to reveal structures within the data while removing background regions. (Kniss et al., 2002) uses a combination of data values, such as the gradient — the first derivative of image data that indicates the

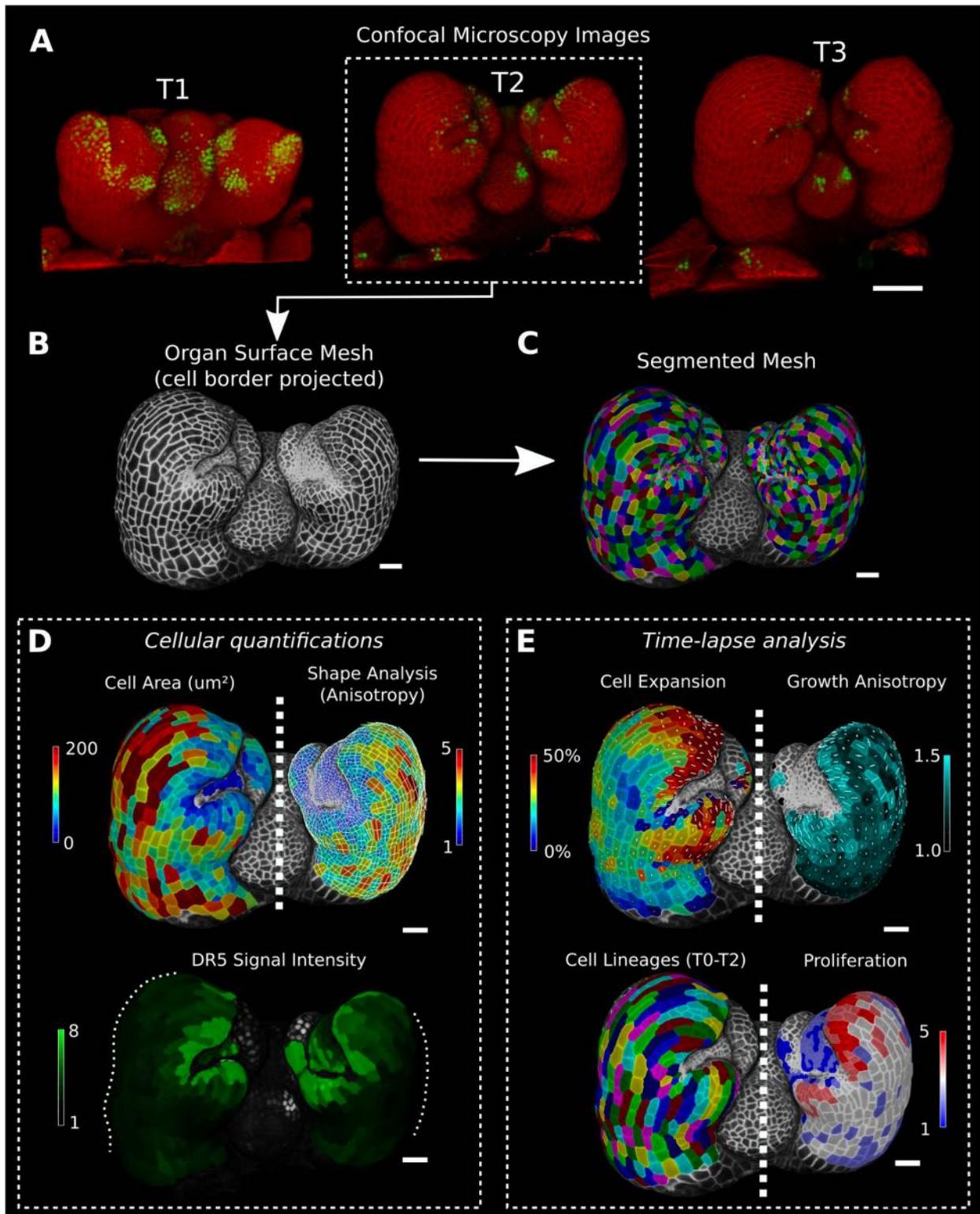


Figure 2.5: Image analysis pipeline in MorphoGraphX. Cells of *A. thaliana* flower meristem captured using confocal microscopy is segmented and quantified from 3 different time points. Permission to republish is given by CC BY 4.0 license.

direction of greatest change — and the second derivative, on a 2D transfer function to increase the chances of isolating a structure of interest by revealing more contrast from neighbouring structures compared to using only the intensity value. Additional information can be provided for each voxel by assigning multiple scalar values, which enhances the ability to classify structures by incorporating information such as colour (Kniss et al., 2002) (Wan et al., 2017), time (Strauss et al., 2021) (Figure 2.5), and a combination of different scanning modalities, such as overlaying fMRI on MRI scans (Nguyen et al., 2010) (Espadaler and Conesa, 2011).

Shading provides visual cues about the form of an object by simulating light. This can be used to visualize the boundary between materials and objects, and to strengthen perception of shapes and depth. Phong (Phong, 1975), and Blinn model (Blinn, 1977) are common lighting approximations that can easily achieve real time performance. However, these methods do not account for light occluded by nearby objects, which is important to perceive spatial structure of an object, and to increase realism (Sattler et al., 2005). Local ambient occlusion can be used to precompute the amount of light received at each voxel around a spherical region. Multiple rays are sampled to capture the amount of occlusion at each voxel (Hernell et al., 2007). Structures may also emit light to perceive them through translucent objects (Nguyen et al., 2010).

By moving away from physically accurate simulations, a wider range of appearances can be rendered. Non-photorealistic rendering methods can convey structures within a volume more effectively by enhancing image features. This includes enhancing silhouette around an object, adjusting colours based on depth, and highlighting region of interest using a different colour (Rheingans and Ebert, 2001). To mimic the appearance of illustrations, the shading on the surface of objects may be hatched (Gasteiger et al., 2008), or the silhouette of objects may be rendered as line drawing (Burns et al., 2005).

To provide informative visualization at interactive rates, streamlining rendering performance

is critical. Usher et al. (2018) moved all non-rendering work onto the background threads, used simple shading models, and strictly budgeted work on the rendering thread to meet the target frame rate of 90Hz.

## 2.4 Interactive Devices

Exploration of data is an important process for the user to familiarize themselves with the dataset. This may involve changing the view to some arbitrary position and orientation, or to adjusting the clipping planes to reveal internal structures within volumetric data (Kniss et al., 2002). With a complex dataset, more precise interaction is needed to avoid obscuring the view of the target structure and to place seeds within the structure for annotation.

The choice of devices can determine how the user interacts with the image data. Devices can be categorized by their degrees of freedom (DOF) — the number of directions of movement. For example, a mouse is a 2 DOF device that moves in two directions: up-down and left-right. Moving the mouse provides input to the computer, which is mapped to the 2D plane of a computer screen. It’s popular due to its versatility of applications and inexpensive cost. MorphographX uses a combination of mouse and keyboard to pan and rotate around the volumetric image data, adjust the clipping planes, manually align polygon data with volumetric data, and to select cells in confocal scans (Barbier de Reuille et al., 2015). Virtual Finger determines the location of a structure by projecting the position of the mouse from 2D screen space to 3D location on the visible object (Peng et al., 2014b). Despite advancements, mapping 2D inputs on 3D volumetric data is unintuitive, and has been shown to be less accurate and immersive compared to 3D interfaces at manipulating virtual objects (Li et al., 2021). In addition, participants found that 3D input device is more “fun” to select, position, and orient objects (Teather and Stuerzlinger, 2007).



Figure 2.6: From left to right. Neurons in the visual cortex of a marmoset monkey labeled with green fluorescent protein are traced by placing the brush inside of the neuron, and then following along its branches while holding the trigger. The trigger is released to stop tracing (Usher et al., 2018). Permission to republish is not required for a thesis, as mentioned in FAQ regarding IEEE Permissions.

By using a 3D device, users can translate and rotate objects in three dimensions. 3D devices may be a 3 DOF devices with three directions of translation, or 6 DOF devices with three directions of translation and rotation. Heng et al. (2001) used a combination of a stereoscopic display and a 6 DOF stylus to assign a voxel within the 3D volumetric data as a start or end point for tracing. de Notaris et al. (2014) used a dextroscope, a device that has one 6 DOF controller for each hand, to place anatomic landmarks on volumetric image data as an extension to the operator’s hands.

Virtual reality (VR) devices have been growing in popularity due to its low cost and immersion inside of a virtual environment. The user wears a VR headset on the head, and holds a 6 DOF controller on each of their hands to provide a one-to-one mapping from the user’s movement to the movement in the virtual environment. This was used to trace neurons in volumetric data (Usher et al., 2018) (Figure 2.6), and to train surgeons in an immersive environment (Pinter et al., 2020). However, VR rendering requires intensive graphical processing to maintain a high resolution for each eye at a high frame rate to prevent motion sickness<sup>4</sup>. In contrast, lower resolutions and frame rates are more tolerable on desktop displays. Users

---

<sup>4</sup>Usher et al. (2018) renders a resolution of  $2160 \times 1200$  at a smooth 90Hz frame rate.

are also likely to become more fatigued while completing tasks in VR due to performing actions in mid-air at shoulder height (Gates and Dingwell, 2011) (Hincapié-Ramos et al., 2014).

Interaction can be improved by combining different modalities of input, such as visuals and haptics. Haptics is the interactive simulation of touch (Salisbury et al., 2004), which is conveyed as tactile information — perception of contact with objects through spatial distribution of pressure in the region of touch — and as kinesthetic information — perception of position and movement of limbs through the movement of proprioceptors in the muscles and joints. It is used to understand spatial information between objects, and simulate physical attributes such as shape, elasticity, texture, and mass. Haptic devices are used to perform tasks that would normally be done using hands, such as exploring and manipulating objects (Srinivasan, 1995). Haptics is commonly used for surgery simulators to train surgeons with a deeper level of immersion (Serrano et al., 2020)(Wei et al., 2012)(Shi et al., 2020)(Fundamental Surgery, n.d.)(Bernardo and Evins, 2018). Note that a haptic device may have any number of DOF, from a 1 DOF syringe to a 6 DOF graspable device with a stylus (Salisbury et al., 2004).

A variety of haptic devices exists to provide 3D interactions such as haptic gloves (Meta), shape displays (Siu et al., 2018), and mobile robots (Suzuki et al., 2021). However, kinesthetic haptic devices, such as Touch Haptic Device by 3D Systems, will be focused on due to being the most common haptic device for interacting with volumetric data (Culbertson et al., 2018).

Applying haptics has been shown to make the task more purposeful, accessible and immersive (Enayati et al., 2016) by providing an additional channel of information that promotes situational awareness and reduces information overload (Culbertson et al., 2018). By using haptics, users learnt to do tasks faster with more confidence and less mistakes (Fang et al., 2014). Furthermore, combining both haptics and visuals improves the learning curve, and

enhance visuospatial skills. Adding haptic feedback has been shown to reduce user error, time taken to complete a task, and energy spent during a task compared to relying on visual feedback alone (Morris et al., 2007) (Wei et al., 2012) (Bernardo and Evins, 2018). In addition, a lack of haptic feedback has been shown to decrease accuracy of interaction, increase time taken to complete a task, and increase the likelihood of fatigue (Teather and Stuerzlinger, 2007). In Deng et al. (2020), only marginal improvement was shown for accuracy, but participants preferred it because it “improved confidence in their performance, increased ease of use, and facilitated more immersive user experience.” (de Notaris et al., 2014) expressed desire for haptic feedback to exploring complex structures using 6 DOF controllers.

## 2.5 Haptics Rendering

A kinesthetic haptic devices inputs position, and outputs force. The input position from the device is represented virtually as a haptic probe to interact with the virtual environment. The output haptic feedback is returned from a haptic rendering algorithm that consists of collision detection and force response. Collision detection calculates when, where, and to what extent collision has occurred based on factors such as depth of penetration, area of intersection, and more. Then, that information is used to compute a force response between the probe and a virtual object (Salisbury et al., 2004). Unlike visual rendering where the minimum frame rate of around 24Hz, the sense of touch is much more sensitive. A typical rate is roughly 1000Hz to achieve smooth feedback (Englund et al., 2018).

Haptic feedback may be represented as force field around the probe to move in some direction and magnitude. A common direction is in the negative direction of the image gradient to move the probe towards the surface of an object. Avila and Sobierajski (1996) combined ambient force, retarding force, and stiffness force in the force feedback. Ambient force is

the force independent of the image data, such as gravity and a guiding plane for the probe; retarding force slows down the probe when it moves too quickly; and stiffness force provides force in the negative direction of the gradient based on depth of penetration of probe in objects. This force feedback was used to interactively edit volumetric data. Faludi et al. (2019) provided force feedback using a force field in the negative direction of the gradient when the probe penetrated an object. The strength of the feedback was based on the opacity of voxels within a radius of influence around the probe, which removed haptic feedback in regions that were not visible. Malmberg et al. (2006) used haptic feedback to touch the surface of 3D objects or the guiding plane that shows a 2D image slice inside a volume with an arbitrary position and orientation. This was used to apply seed points for 3D live-wire to extract the boundary of object of interest.

Another method to move the probe is by using a proxy — a primitive that is constrained to a virtual surface. A spring force is provided between the probe and the proxy such that the probe is pulled towards the proxy with force relative to the distance between them (Lundin et al., 2002). In other words, the probe is attracted towards the proxy. The proxy is most commonly represented as a point, but it could be represented as a line, a plane, or a direction (Lundin et al., 2005). Vidholm et al. (2006) placed seeds inside of an object to annotate the boundary of a volumetric object. The volumetric data was visualized as three orthogonal planes aligned with the faces of the volume. The probe was placed on one of the planes through the spring force between the probe and the proxy. Kaluschke et al. (2020) transformed volumetric objects into sphere packing representation for proxy-based haptic. This filled the virtual object using tightly packed set of spheres, which provided a well-defined force direction at all times. As a result, pop-throughs were avoided even at high velocities. Corenthy et al. (2015) used proxy-based haptics to interact with the isosurface — 3D surface from points of equal values — of volumetric objects extracted using the marching tetrahedra algorithm. Marching tetrahedra algorithm was chosen instead of the more popular marching cube algorithm due to less topological ambiguity problems.

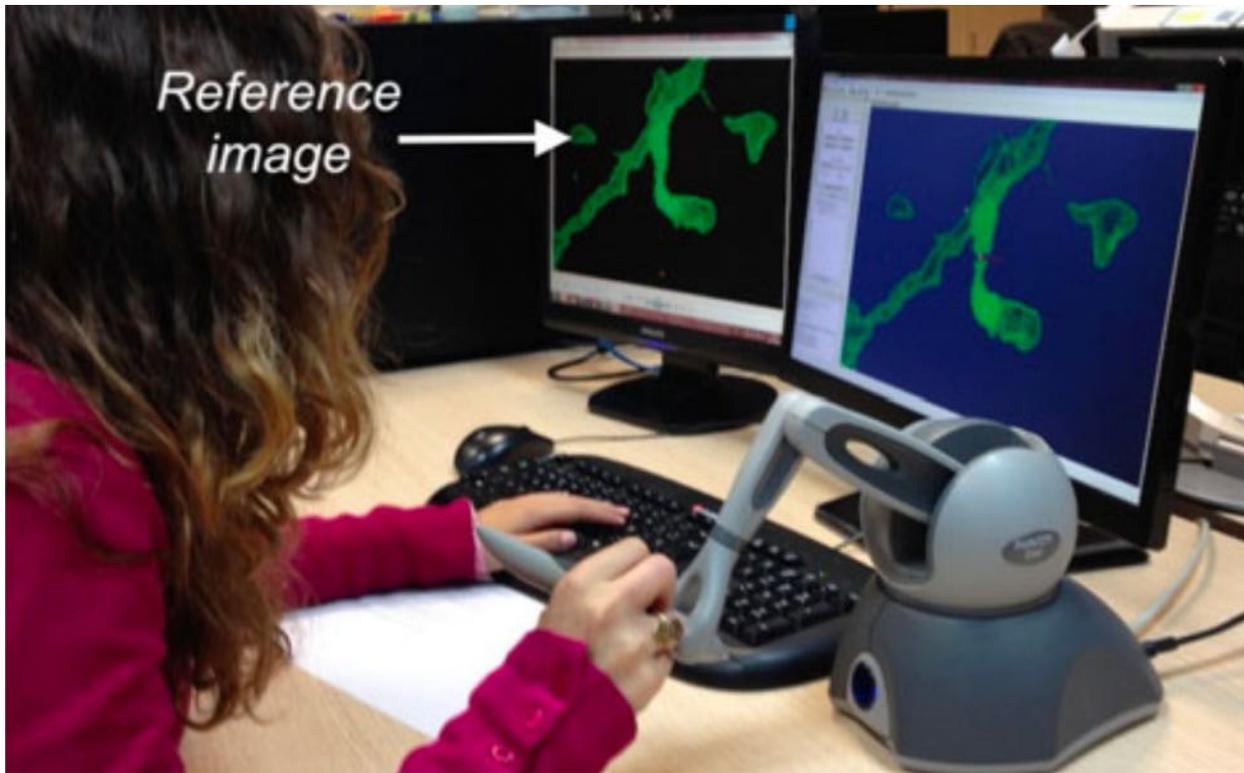


Figure 2.7: Small dendritic splines are recovered by using the haptic device for navigation and volume image editing. The haptic device provides 6 DOF control over the dataset, and the force feedback helps to place target points on the surface (Corenthy et al., 2014). Permission to republish is not required for a thesis, as mentioned in FAQ regarding IEEE Permissions.

In addition to surfaces, force feedback may be used to explore and interact with tubular structures. Hardens and Szekely (2003) provided haptic feedback along the centerline of tubular intestines by precomputing a Euclidean distance map from a voxel in the intestine to a voxel in the surrounding tissue. From this map, the gradient was calculated to direct the probe along the centerline. Using this feedback, a user removed unrelated voxels from the initial segmentation result. Englund et al. (2018) used haptic feedback to enhance visual exploration of 4D MRI — volumetric MRI data at various temporal moments — of the heart that conveyed the path of blood flow as a vector field. To explore this data, three modes of haptic are used: *force mode* to directly map the direction and strength of vector field to the probe; *follow mode* to provide resistance if probe moves perpendicular to the vector field; and *line follow mode* to follow one of the selected pathlines. By following along the flow using

the haptic probe, flow features that are normally hidden from occlusion can be explored and analyzed. Corenthy et al. (2014) used proxy-based haptics to touch or constrain the probe to an isosurface (Figure 2.7). Haptic feedback was used to place seeds for tracing, and to edit the intensity value of voxels around the proxy. Because kinesthetic haptic devices provide a convenient point of interaction, haptic feedback has been used to annotate structures within volumetric data. However, none of the reviewed methods explored haptic interaction with the vascular structure of a plant.

# Chapter 3

## Haptic Assistance

Unlike surgery simulators that aim to provide an immersive virtual environment where surgeons can plan and execute various procedures before performing them in vivo (de Notaris et al., 2014) (Serrano et al., 2020), the goal of haptic assistance<sup>1</sup> is to help explore and interact with the centerline of a network of tubular structures. This requires robust feedback that can interact with various size and shape of flower head, and precise control to avoid non-vascular structures such as bracts and trichomes.

Creating an effective haptic feedback has several challenges. First, rendering detailed haptic feedback has a trade-off with computation speed. In particular, collision detection and resolution becomes increasingly more expensive as the shape of object becomes more complex (Xia, 2018). This causes haptic rate to decrease, which results in instability from energy added into the system due to the probe lagging behind its actual position. (Salisbury et al., 2004). Another challenge is that haptic feedback needs to be insightful while annotating volumetric data. (Preim and Botha, 2014) provides four requirements for an effective segmentation method: robust, accurate, reproducible, and fast. In other words, the method

---

<sup>1</sup>This terminology was chosen because haptic feedback provides a direction towards where the user will likely want to annotate. In contrast, “haptic guidance” suggests that haptics is doing much of the annotation for the user.

must be able to handle a large variety of cases, output results close to the desired ground truth, produce similar results even after several attempts, and be fast enough to keep up with the user’s intent. Note that the importance of each requirement differs based on the goal of application.

### 3.1 Haptic Modes

Proxy-based haptics is used to guide the probe towards the structures of interest. This speeds up the annotation workflow while the user proof-edits complex regions using knowledge about the target structure. To accommodate a variety of potential interactions, the system has three different haptic modes using proxy-based haptics: surface sampling, proximity following, and object following.

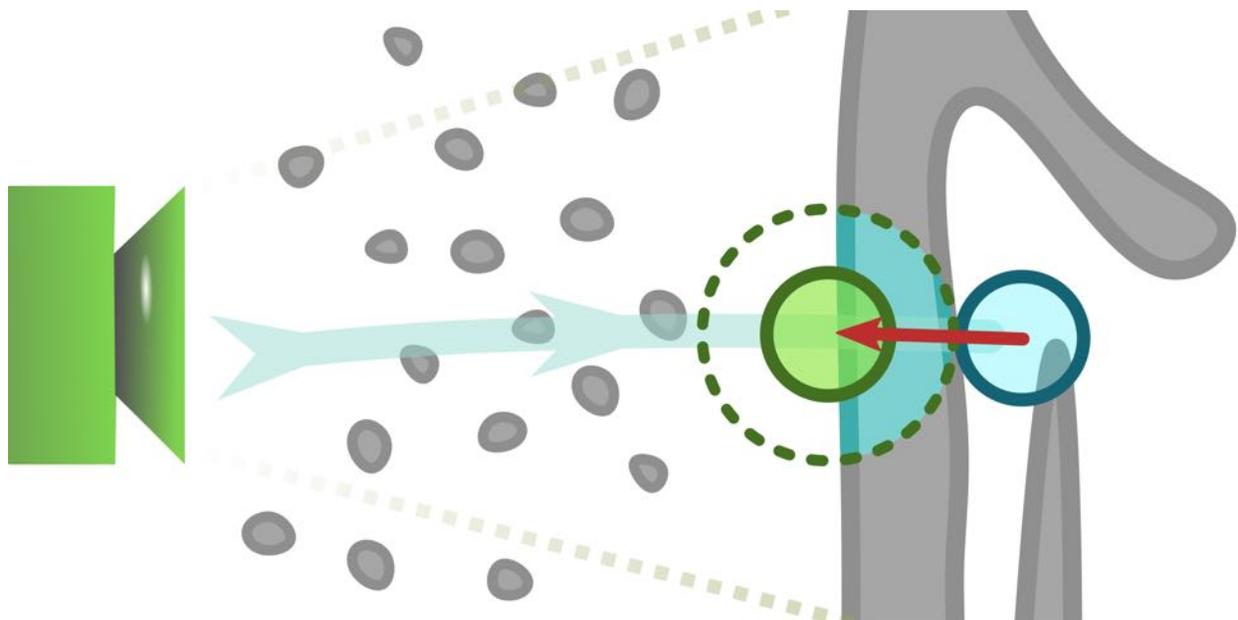


Figure 3.1: Example of surface sampling. Surface sampling ignores relatively small objects as the probe moves away from the camera (light blue arrow). When the probe encounters a large object, the proxy (light green circle) is placed where the percentage of relevant sample points exceeds some threshold rate (dotted green circle). Moving the probe deeper into the object returns a spring force toward the proxy (red arrow). The proxy is released when the probe moves towards the camera.

Surface sampling is used to touch the surface of an object. This is done by preventing the proxy from penetrating through visible objects that occupy more than some threshold amount of region within the probe. This criterion is loosely based on thickness of the object at a local region surrounding the probe. Although this allows the probe to pass through some opaque surfaces, it is useful to ignore small objects that are much smaller than the probe (Figure 3.1). To release the proxy, the probe only has to move toward the camera. This frees the proxy, and allows the probe to move without restraint. Ways to interact using this mode include stepping the probe along the surface of an object, and following along the cross-section of a tubular structure.

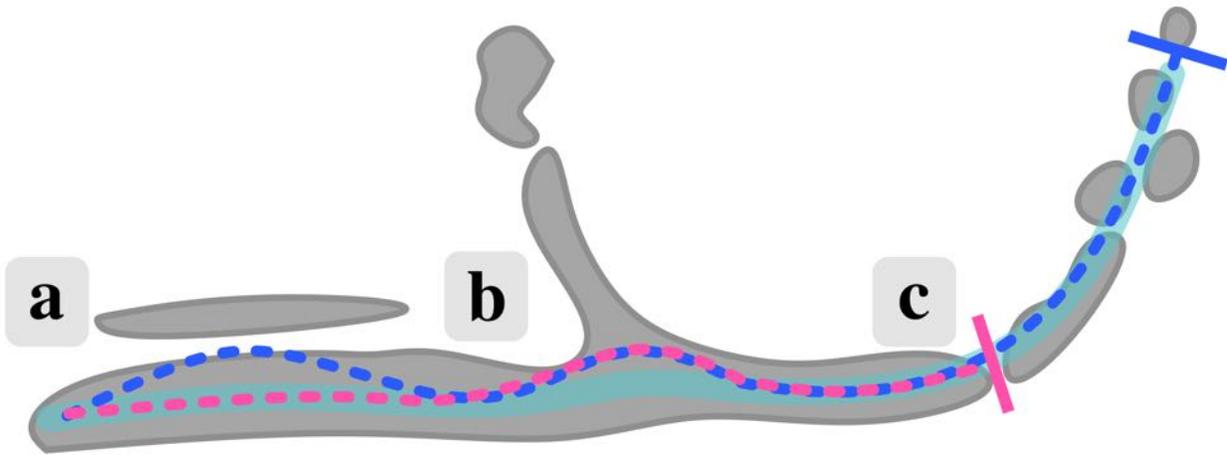


Figure 3.2: Example of proximity and object following modes. The two dotted lines indicate the path of the proxy using object following (magenta), and proximity following (blue). The path of the probe is indicated by light blue line below. The difference in haptic feedback can be seen at (a), where a proximity following moves the proxy toward a nearby object, and at (c), where object following cannot move past the current object. The intersection at (b) moves probe away from the centerline for both object and proximity following.

Proximity and object following are similar in that they can be used to follow close to the centerline within a tubular structure. This is done by placing the proxy in the center of a group of objects in a local region surrounding the probe. The difference is that proximity following considers all objects surrounding the probe, whereas object following considers only one object. In other words, both modes would output the same haptic feedback if used on an

isolated object. But if the probe is also surrounded by small objects, then the feedback from proximity following will differ from object following. Thus, proximity following is useful to follow along a trail of objects, while object following is useful to follow along a single object (Figure 3.2).

## 3.2 Implementation Details

This haptic method uses proxy-based haptics based on points sampled locally around the probe. Information about image features within objects of interest are marked and stored in a boolean volume. Then, the points are used to calculate the proxy position  $\vec{p}_{proxy}$  using the information stored in the boolean volumes. Force feedback is provided using the spatial difference between the probe and the proxy. This section will provide the implementation detail of each step. The visible regions containing objects of interest will be referred to as “relevant regions”. Although haptic feedback can be independent from the visibility of region, “visible region” will also refer to regions that are interact-able using haptic feedback for simplicity sake. In fact, haptic feedback is set to match visible regions by default.

### 3.2.1 Background on Proxy-based Volume Haptics

Within volumetric data, object voxels need to be explicitly differentiated from background voxels. Since each voxel contains a scalar intensity value, the simplest method to define an object is by defining a haptic threshold value  $t_{haptic}$ , which assigns all voxels  $V(x, y, z)$  of a volume  $V$  under a threshold intensity value to be part of the background, and all other voxels to be part of an object. With this classification, a volumetric object would be a connected set of voxels where each voxel has a path to its neighbouring object voxels. In addition, the surface of a volumetric object would be the a region in between background voxels and

object voxels.

To provide haptic feedback, the haptic algorithm needs to detect and resolve collision between the probe and an object. Collision detection assesses when, where, and how large a collision has occurred with the probe. Then, collision resolution returns a force  $\vec{f}$  to provide force feedback to the user (Salisbury et al., 2004). For proxy-based haptics, the movement of the probe is constrained to some object feature, such as a point, line or plane, or to some direction of movement, by finding the proxy position that the probe should be attracted towards (Lundin et al., 2005).

When the probe encounters an object, haptic feedback may push the probe away from the surface by constraining the movement of the proxy to the surface. The surface is defined at each point in between object and background voxels and the direction perpendicular to the surface, known as the surface normal. The surface normal is calculated using the local gradient  $\nabla V(x, y, z)$  at a point on the surface. By iteratively updating the proxy position  $\vec{p}_{proxy}$  to be the nearest surface point to the probe  $\vec{p}_{probe}$ , the haptic feedback is provided in the direction of the surface normal. Haptic feedback may also be combined with material properties such as friction, stiffness, viscosity and penetrability to add additional immersion in the feedback. However, surface normal may be undefined or very weak at some regions within a volume, such as on the surface of small objects, bumpy surfaces or surfaces with cusps. This may add energy to the system and cause instability due to the lack of continuity in the haptic feedback (Lundin et al., 2002). Furthermore, haptic methods for touching the surface of an object is not suitable for thin structures due to the lack of surface area to interact with. The proposed haptic method avoids using surface normals when calculating the proxy position  $\vec{p}_{proxy}$  to provide a more robust haptic feedback that can also be used to interact with thin structures.

### 3.2.2 Marking Image features

Relevant regions contain image features, such as voxel intensity values and connectivity, that can identify objects of interest. These regions are marked within the local region — the region surrounding the probe bounded by minimum and maximum cell indices — as a boolean value to indicate whether the relevant image feature is present or not.

The local region  $L$  maps the position and size of the 3D probe to a region within the volumetric data. It is centered at the cell index  $\vec{c}_{probe}$  that corresponds with the probe position  $\vec{p}_{probe}$  in world space. A position in world space  $\vec{p}$  to a cell index  $\vec{c}$  within volumetric data is converted by performing the following equations:

$$\vec{p}_o = M \cdot \vec{p} \tag{3.1}$$

$$\vec{c} = \left( \frac{\vec{p}_v \cdot x}{w_o} \cdot w_{cell}, \frac{\vec{p}_v \cdot y}{h_o} \cdot h_{cell}, \frac{\vec{p}_v \cdot z}{d_o} \cdot d_{cell} \right) \tag{3.2}$$

where  $M$  is a transformation matrix to go from world space to object space,  $\vec{p}_o$  is the position of the probe in object space,  $w_o$ ,  $h_o$  and  $d_o$  are the width, height and depth of the volume in object space, and  $w_{cell}$ ,  $h_{cell}$  and  $d_{cell}$  are the width, height and depth of the cells in the corresponding volumetric data. The size of the local region is set to one cell larger than the radius of the probe to include all potential sample points. In addition, the local region adjusts dynamically to fit within the bounds of the volume.

Different haptic modes provide different relevant regions. If surface sampling or proximity following are selected, then all object voxels within the local region are considered relevant. If object following is selected, then connectivity is also considered by applying region growth method from a seed point, which iteratively marks all unmarked adjacent object voxels and

then adds the newly marked voxels to be seeds for the following iterations of region growth until no more seed points are in the queue.

---

**Algorithm 1:** Relevance Thread

---

```

Input: Seed point  $s$ , volume data  $V$ , local region  $L$ , haptic threshold value  $t_{haptic}$ ,
          back relevance volume  $V_r$ , and haptic mode  $h$ 
// Clear local region  $L$  in back relevance volume  $V_r$ 
for all voxels  $V(i, j, k)$  in local region  $L$  do
  |  $V_r(i, j, k) = \text{False}$ 
end
// Mark local region  $L$  in back relevance volume  $V_r$ 
if ( $h = \text{"surface sampling"}$  OR  $h = \text{"proximity following"}$ ) then
  | for all voxels  $V(i, j, k)$  in local region  $L$  do
    | | if  $V(i, j, k) \geq t_{haptic}$  then
      | | |  $V_r(i, j, k) = \text{True}$ 
    | | end
  | end
else if ( $h = \text{"object following"}$ ) then
  | Region growth from  $s$ 

```

---

Figure 3.3: Implementation of relevance thread. It updates the local region around the probe to mark relevant regions.

In the implementation, a thread is dedicated to updating the relevant region. This is done to relieve the expensive task of marking relevant thread from the time-sensitive main haptic loop. This thread, which will be referred to as “relevance thread”, uses the information about the local region, the haptic threshold value, the current haptic mode, and a seed point to update the relevant regions (Figure 3.3). To store information about the relevant regions, two boolean volumes are used. These volumes will be referred to as “relevance volumes”  $V_r$  for storing information about relevant regions. These volumes have the same dimensions as the original volumetric data to act as double buffers, where the back relevance volume is updated with information about relevant regions surrounding the probe, while the front relevance volume is used by the main haptic loop to compute haptic feedback. Note the dimensions of the relevance volumes are different from the dimensions of the local region to prevent resizing the relevance volumes when the size of the probe changes.

Because relevance thread needs to keep the parameters consistent while the back relevance

volume is updated, the thread stays idle once complete. When the main haptic loop detects that the thread is idle, the parameters are updated, the front and back relevance volumes are swapped, and the thread starts again. This synchronization limits the amount of excess computation performed if the relevant region is small, while allowing larger relevant regions to be updated without stalling the main haptic loop. Although this may result in the current relevant region lagging behind the actual relevant region, this issue is barely noticeable in the haptic feedback.

Precomputing the connectivity volume for the entire sample can be tempting to reduce real-time computational load. However, the settings to interact with one structure may be different for another. For example, a structure of interest may be connected to another structure at one haptic threshold value, but not at a higher threshold value. However, a different structure of interest may not be perceivable at a high threshold value, which will require lowering the haptic threshold value. By dynamically updating the relevant region, users can choose what structures to interact with.

### **3.2.3 Locating the Proxy**

The proxy is calculated using a collection of sample points, which are uniformly distributed within the local region surrounding the probe (Figure 3.4). The points are created by adding offset positions to the probe position (Figure 3.5). The offset positions are initially created in polar coordinates before converting to Cartesian coordinates (Weisstein, n.d.) and then scaled by the probe size. By having the points sample from the relevance volume, information included in the relevant regions is incorporated into the sample points.

Proxy calculation differs based on the selected haptic mode. Surface sampling checks the percentage of relevant sample points out of the total number of sample points within a local region. If the percentage exceeds some threshold, then the proxy becomes fixed in place.

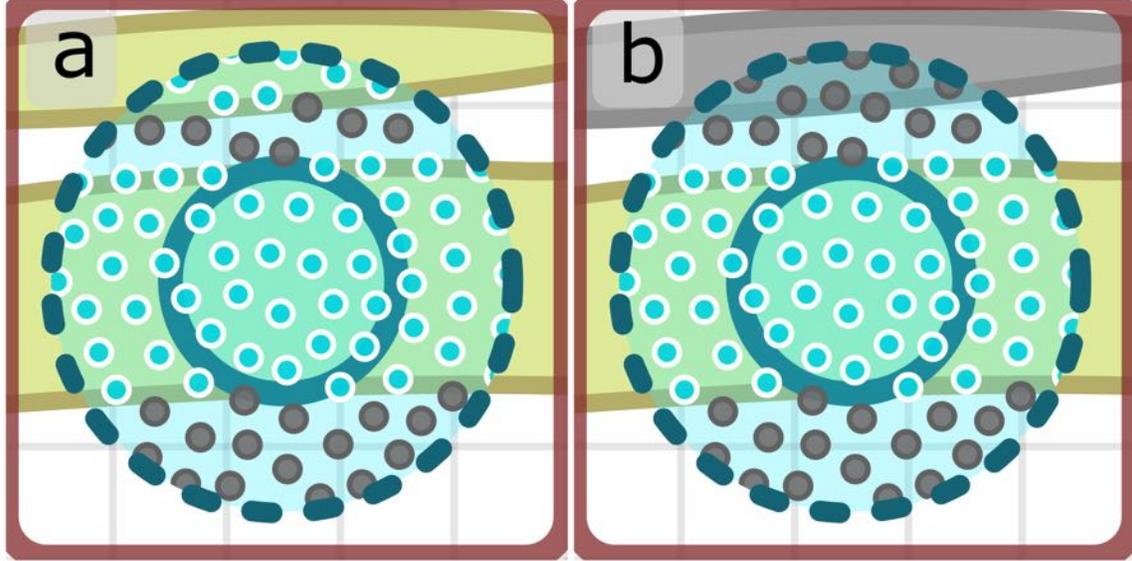


Figure 3.4: For proximity following (a), all sample points that are within relevant regions are relevant. For object following (b), only points within a relevant region connected to the previous relevant region are relevant.

This is represented by the equation:

$$\frac{n}{n_{total}} > ssRate \quad (3.3)$$

where  $n$  is the number of relevant sample points,  $n_{total}$  is the total number of sample points, and  $ssRate$  is a threshold rate for surface sampling. In other words, the movement of the probe will be inhibited when it is surrounded by relevant region. When the probe moves toward the camera, the proxy is released from its position, and follows the probe until a new surface is detected. Unlike methods that use the gradient to find the surface, this method measures the confidence level of whether the region is likely to contain a region of interest or not. As a result, this corresponds more with thickness of object rather than the location of surface. This method has the benefit of outputting a robust feedback that can reliably interact with objects without any stability issues, and avoiding interaction with small objects that are both undesirable to interact with and difficult to perceive while exploring. Note that the user can choose the size of the object to interact with by changing the size of the

---

**Algorithm 2:** Calculation of Sampled Points

---

**Input:** Number of sample points  $N$ , probe  $\vec{p}_{probe}$ , probe size  $r_{probe}$ , and uniform distribution of random numbers between  $[0,1]$   $u_0, u_1, u_2$

**Output:** List of sample points  $S_{sample}$

```
// Precompute randomly distributed offsets in a sphere
for  $N$  count of offsets  $\vec{o}_i$  do
     $r = \sqrt[3]{u_0}$ 
     $\theta = \cos^{-1}(2 \cdot u_1 - 1)$ 
     $\phi = 360 \cdot u_2 - 180$  // Value between  $[-180, 180]$ 
    // Convert polar coordinates to Cartesian coordinates
     $\vec{o}_i = (r \cdot \sin\theta \cdot \cos\phi, r \cdot \sin\theta \cdot \sin\phi, r \cdot \cos\theta)$ 
end
...
// Update sample points around the probe
for each sample point  $\vec{s}_i$  in  $S_{sample}$  do
     $\vec{s}_i = \vec{p}_{probe} + r_{probe} \cdot \vec{o}_i$ 
end
```

---

Figure 3.5: Calculation of sample points  $s_i$ . The offset positions  $o_i$  are precomputed from polar coordinates and stored in a list. Then, the offset positions are scaled by the probe size  $r_{probe}$  and added to the probe position  $\vec{p}_{probe}$  to create sample points.

probe. By default,  $ssRate$  is set to 40%, which places the proxy near the surface of the object.

Both proximity following and object following modes use the arithmetic mean to calculate the position of the proxy  $\vec{p}_{proxy}$ . These two modes resemble “follow mode” from (Englund et al., 2018), where the probe is guided towards the centerline when nearby a tubular structure. It is represented by the formula:

$$\vec{p}_{proxy} = \frac{\sum_{i=0}^n \vec{a}_i}{n} \quad (3.4)$$

where  $n$  is the total number of relevant sampled points, and  $\vec{a}_i$  is the position of each relevant point. This places the proxy at the average position of all relevant points. If there are no relevant points, then the proxy is set to follow the probe. Note that intensity values are not used to weigh each sample point. This is because users cannot see the intensity value of voxels

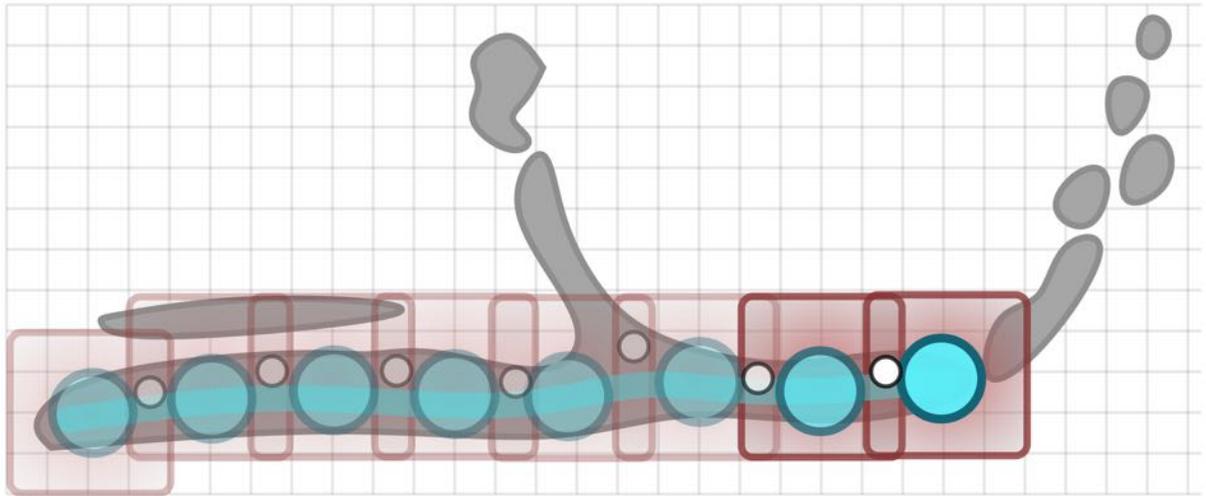


Figure 3.6: Updating relevance volume for object following. Only the relevant region within a local region that is connected to the previous relevant region is updated. A probe (teal circle) is surrounded by a voxel-aligned local region (dark red square). At each time step, a seed (white circle) is produced where the current and previous local relevant region overlaps. This seed is used to apply region growth method in the local region around the probe for haptic assistance.

inside of a structure. If the calculation of proxy was weighted, then the user will become confused when the highest intensity values within a local region is not at the centerline of a tubular structure.

The difference in implementation between the two modes lies in the relevance volume. For proximity following, all voxels surrounding the probe that are above the haptic threshold are considered. Object following mode further limits the inclusion of voxels by also requiring voxels to be connected to the initial point of contact between the probe and the relevant region. This is done by applying region growth method at the point of contact within a local relevant region. When the probe moves, a seed for region growth method is obtained inside of an overlapping region between the previous and current local relevant region. By ensuring connectivity between previous and current local relevant region. the continuity from the initial point of contact is maintained (Figure 3.6).

For all haptic modes, increasing the number of sample points can produce smoother, more

consistent feedback by updating the proxy position  $\vec{p}_{proxy}$  more gradually as the probe moves around at the cost of performance. In the implementation, a fixed count of 3000 sample points is used.

### 3.2.4 Returning Force Feedback

In proxy-based haptics, the probe moves toward the proxy to provide force feedback. As the distance between the probe and the proxy increases, the force becomes stronger towards the proxy. As the probe moves faster, the force is dampened to reduce instability from collision with an object. Two components of force, the spring force  $\vec{f}_s$  and the damping force  $\vec{f}_d$ , are represented by the formulas:

$$\vec{f}_d = k_{damping}(\vec{p}_{probe\_prev} - \vec{p}_{probe}) \quad (3.5)$$

$$\vec{f}_s = k_{spring}(\vec{p}_{proxy} - \vec{p}_{probe}) \quad (3.6)$$

where  $k_{spring}$  is the spring coefficient,  $k_{damping}$  is the damping coefficient,  $\vec{p}_{probe}$  is the current position of probe,  $\vec{p}_{probe\_prev}$  is the position of the probe from the previous time step, and  $\vec{p}_{proxy}$  is the position of the proxy.

One potential source of instability is when the size of the probe changes. As the probe becomes larger, the proxy will also be located further away, which will increase the spring force  $\vec{f}_s$ . However, since the magnitude of the spring force is based on the distance between the probe and the proxy, the output force can become too strong to resist assistance. Similarly, when the probe becomes smaller, the output force becomes too weak to perceive assistance. Another potential source is when the the probe interacts with distant structures. Because haptic devices have finite resolution in their positional sensors, determining when and where

contact occurs always results in a quantization error, where the probe position will always lag behind the actual position from the haptic device. This introduces additional energy into the system, which can cause instability. (Salisbury et al., 2004). When interacting with distant objects, the smaller movement on a haptic device increases the quantization error to the point where damping force  $\vec{f}_d$  is not sufficient. To address these issues, the output force  $\vec{f}$  is inversely scaled by the size of the probe  $s_{probe}$ , and the scale of movement  $s_{movement}$  such that:

$$\vec{f} = \frac{\vec{f}_s + \vec{f}_d}{s_{probe} * s_{movement}} \quad (3.7)$$

---

**Algorithm 3:** Haptic Loop

---

```

while (Simulation is running) do
  Read haptic device
  Update probe  $\vec{p}_{probe}$ 
  ...
  if (Haptics is enabled) then
    if (Relevance thread is idle) then
      Update parameters in relevance thread
      Swap front and back relevance volumes
      Start updating back relevance volume  $V_r$  in relevance thread
    end
    Update set of sample points
    Update proxy position  $\vec{p}_{proxy}$ 
  else
     $\vec{p}_{proxy} = \vec{p}_{probe};$ 
  end
  Update force  $\vec{f}$ 
end

```

---

Figure 3.7: Implementation of Haptic Loop. The haptic loop calculates the position of proxy based on the current relevance volume, and returns force to the haptic device. Parameters for relevance thread includes the haptic mode, the haptic threshold value, the local region, and a seed point.

This calculated force  $\vec{f}$  is updated to the haptic device, which is perceived by the user (Figure 3.7).

# Chapter 4

## System Design and Implementation

Haptic assistance provides force feedback to both the surface of volumetric objects, and towards the centerline of tubular structures. This feedback increases spatial awareness within volume to allow the user to interact more accurately and confidently. However, several challenges exist for annotating. First, the region of interest may be obscured by unimportant regions. Second, annotating structures of interest may require excessive amount of interaction. Lastly, figuring out what to annotate and what to avoid may not be a trivial task. To address these issues, tools are provided to explore the contents within volumetric data, to iteratively adjust the visualization, and to quickly annotate structures of interest. These tools are broadly divided into four sections: annotation, volumetric rendering, user interface, and system organization.

### 4.1 Annotation

Within a sample, a variety of anatomical structures may be of interest to annotate. For example, florets, vascular bundles, and bracts are separate structures that provide different

insight about a flower head. Although these structures can be extracted through segmentation, additional information may be obtained through tracing. By simplifying the image data into a graph representation, connectivity between vascular bundles can be extracted without the ambiguity from noise and artifacts. In addition, the output from tracing can provide useful metrics to quantify a network of tubular structures, such as branching angles, number of branching nodes, and total length of network (Acciai et al., 2016).

In SHVR, annotation is performed interactively by placing the probe inside of a structure of interest. Because the haptic device provides information bidirectionally, much of the guidance is provided by haptic feedback rather than annotation algorithms tailored for specific applications.

### 4.1.1 Voxel Painting

Segmentation is performed using voxel painting, which annotates regions of interest by marking the voxels within the local region on a segmentation mask. This mask data has the same dimensions as the image data, but contains indices that refer to a colour and an opacity value. This reference from an index will be called “colour group”. By storing colour groups in each voxel, several different anatomical structures can be segmented within a single sample (Figure 4.1). When the “paint” button is pressed, the existing colour group within the local region is overwritten with the selected colour group based on the chosen paint brushes. This process resembles painting using a brush, where the user strokes along a region of interest using the stylus. This method is similar to (Berger et al., 2018), where they paint around a mouse pointer on a 2D image slice.

Segmentation mask efficiently stores and modifies the colour and opacity for each voxel using colour groups. For example, the index “1” may refer to opaque red, and the index “2” may refer to translucent green. If the colour group of a voxel is set to “0”, the colour and opacity

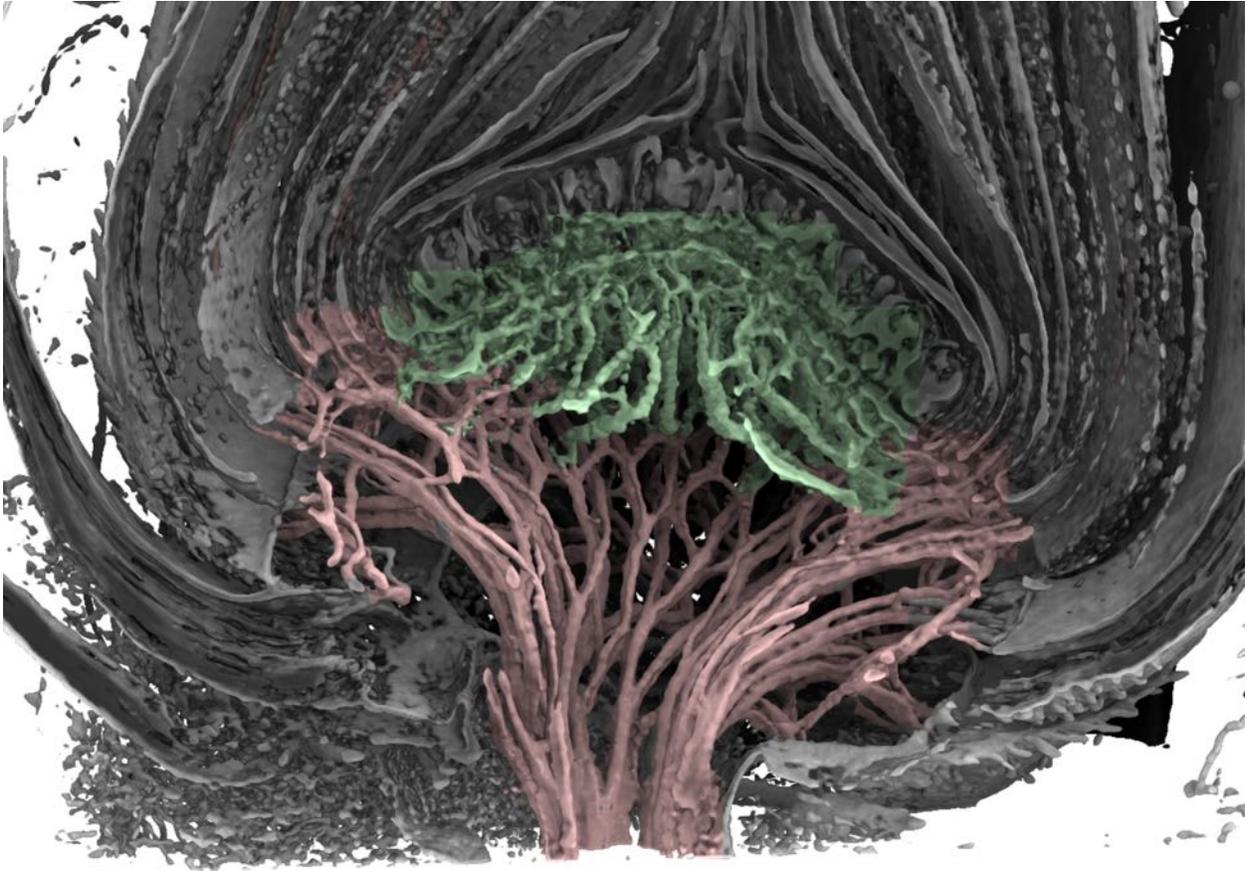


Figure 4.1: Segmentation of the vascular structure within a thistle flower head. The upper vascular bundles (green) are labeled with a different colour compared to the lower vascular bundles (red). In addition, the segmented vascular structure is contrasted from rest of the flower head.

are obtained from a transfer function. This reduces the data stored per voxel from a 32-bit colour and opacity value to an 8-bit index, which reduces the size of the segmentation mask to a quarter. In addition, when the colour and opacity is modified within a colour group, all voxels in that group change together. This provides a quick way to edit the appearance of a large group of voxels.

Three different brushes are available: fill, flood fill, and speckle fill. Fill brush segments all voxels with intensity value above the paint threshold value within the local region. This is useful if the user wants to paint a large region quickly, or paint all voxels within a region. Flood fill brush applies region growth algorithm from a seed located at the center of the

probe. This segments an object without bleeding paint into surrounding objects. Speckle fill brush paints only objects completely encased within the local region. This is used to segment small objects while leaving the large objects untouched (Figure 4.2).

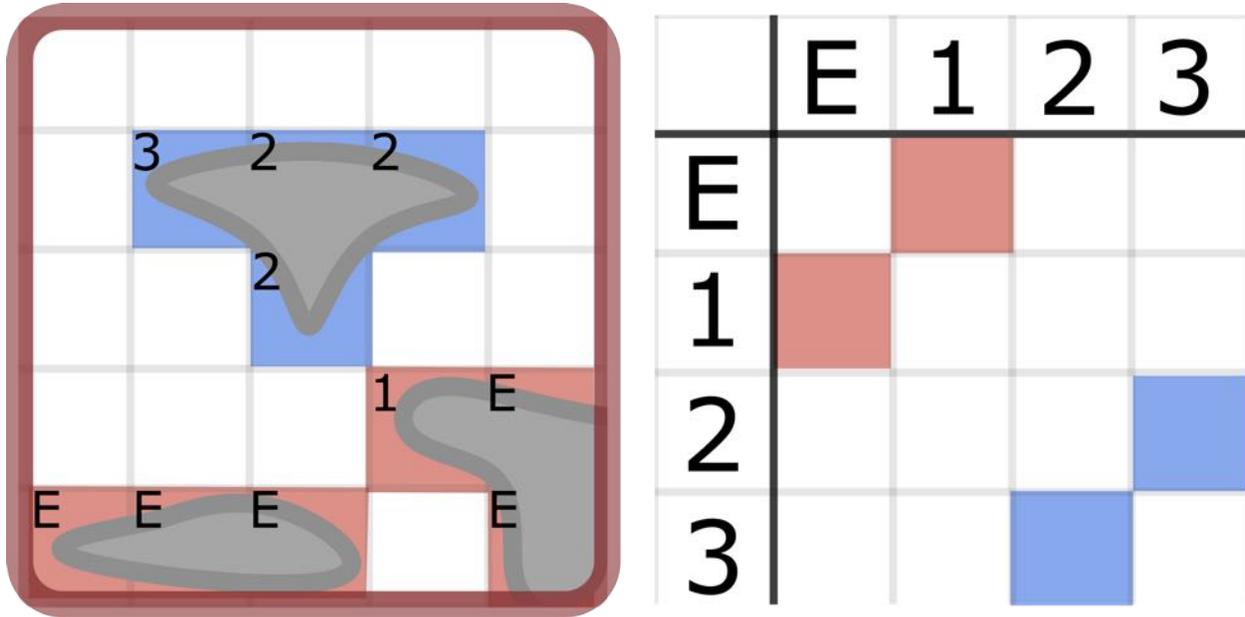


Figure 4.2: Example of speckle fill. (left) Objects that are not touching the edge of the local region are segmented. Each non-background voxel is marked with a seed index. If a voxel is at the edge of the local region, then it is marked with “E”. If there is no neighbouring seed index to the left, bottom, and back of the voxel, then a new seed index is created. (right) An adjacency matrix keeps track of connection between seed indices.

Speckle fill is useful for scenarios where a structure of interest is obstructed by objects in front of it. This distracts the viewer from the main structure. By removing those objects from view, the resulting visualization can be more pleasant to look at. This is done by painting with the opacity of the selected paint group set to transparent. Large objects are easy to remove because they require only a few seeds to segment the entire structure. However, small objects need to be individually painted, or risk accidentally painting on nearby structures when flood fill or fill brush is used. To address this problem, speckle fill was developed such that large number of small objects can be removed quickly within a local region.

The implementation of speckle fill has three steps: identify all objects contained within the local region, output a seed point from each of those objects, and then applying region growth

---

**Algorithm 4:** Speckle Fill

---

**Input:** Adjacency matrix  $[S]$ , volume data  $V$ , local region  $L$ , seed indices within local region  $I$ , and paint threshold value  $t_{paint}$

**Output:** List of seeds  $S$

```
for (All voxels  $V(i, j, k)$  in local region  $L$ ) do
  if ( $V(i, j, k) < t_{paint}$ ) then
    |  $I(i, j, k) = \text{"invalid"}$ 
  else if ( $(i, j, k)$  is located at the edge of local region  $L$ ) then
    | Update  $[S]$  such that  $I(i, j, k) \leftrightarrow \text{"edge"}$ 
    |  $I(i, j, k) = \text{"edge"}$ 
  else
    | // Adopt neighbouring seed index
    | if  $I(i - 1, j, k) \geq 0$  then
    | |  $I(i, j, k) = I(i - 1, j, k)$ 
    | if  $I(i, j - 1, k) \geq 0$  then
    | | Update  $[S]$  such that  $I(i, j - 1, k) \leftrightarrow I(i, j, k)$ 
    | |  $I(i, j, k) = I(i, j - 1, k)$ 
    | if  $I(i, j, k - 1) \geq 0$  then
    | | Update  $[S]$  such that  $I(i, j, k - 1) \leftrightarrow I(i, j, k)$ 
    | |  $I(i, j, k) = I(i - 1, j, k - 1)$ 
    | | // Create new seed index if no neighbour exists
    | | if  $I(i, j, k) < 0$  then
    | | | Create a new seed  $s_i$  at  $(i, j, k)$ 
    | | | Update  $[S]$  to include new seed  $s_i$ 
    | | |  $I(i, j, k) = \text{index}(s_i)$ 
  end
end
Output seeds  $S$  that are not connected to "edge"
```

---

Figure 4.3: Pseudo-code for speckle fill brush. This returns a set of seeds separated from the edge of the local region. These seeds are used for region growth segmentation.

method using these seed points. This is done by assigning a seed index for each voxel. Each seed index refers to a seed position inside of an object. If two voxels with different indices are neighbouring each other in the same object, then this connectivity is marked in an adjacency matrix<sup>1</sup> using the seed indices. If there is no seed index within a voxel's neighbourhood, then a new seed is created along with a new seed index. If there is a seed index, then the seed index of the neighbour is used instead. Because the voxels are iterated only once, the

---

<sup>1</sup>Adjacency matrix is a graph represented as a matrix of booleans. This is used to keep track of connection between seed points, and to prevent creating redundant connections during marking.

neighbourhood check is only done to voxels that were previously visited (Figure 4.2). By finding voxels that are connected to the edge of the local region, objects that extend beyond the the local region can be avoided. After all the voxels are checked, the remaining seeds are used as input for region growth segmentation (Figure 4.3).

### 4.1.2 Tracing

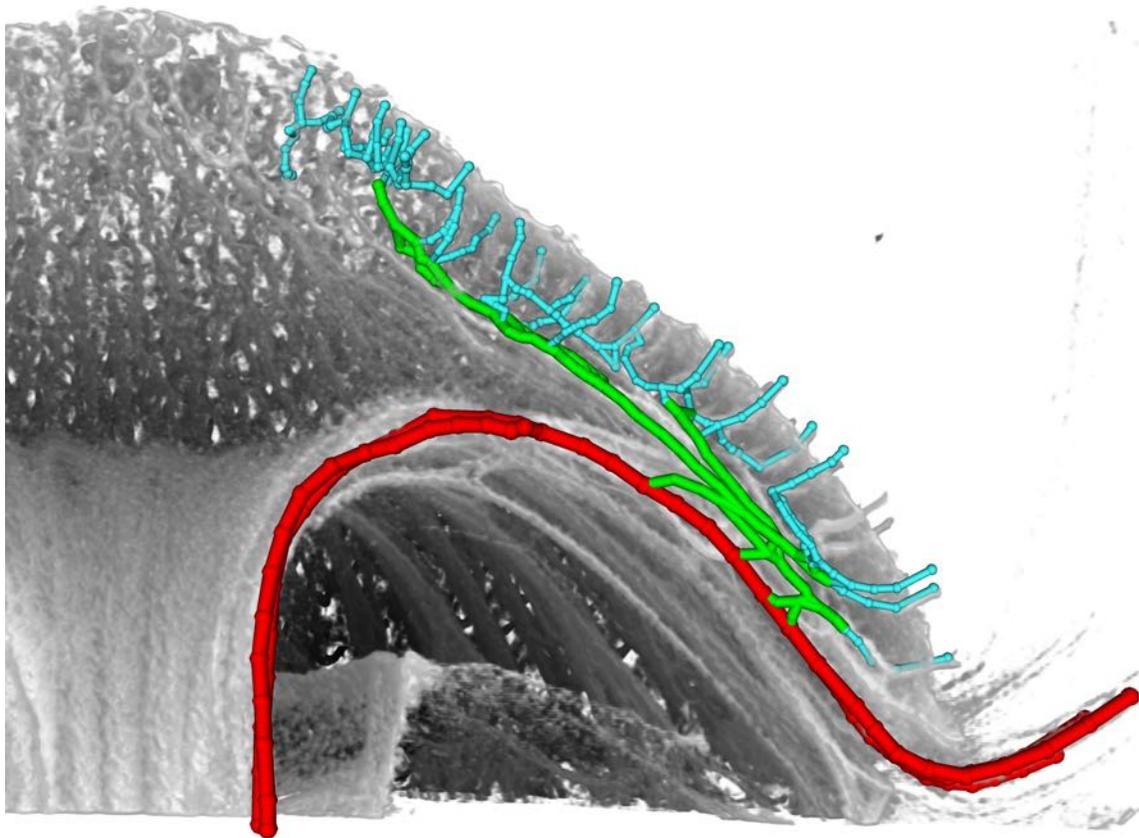


Figure 4.4: Example of tracing. Vascular bundles within a *Gerbera hybrida* flower head are traced along the centerline. The vascular structure is classified into three sections. Connectivity between vascular bundles is easily perceived.

Tracing is manually created as a network of nodes and internodes. By using the guidance of haptic feedback, nodes and internodes can be placed along the centerline of tubular structures (Figure 4.4). This is done by using a combination of the haptic device and the keyboard.

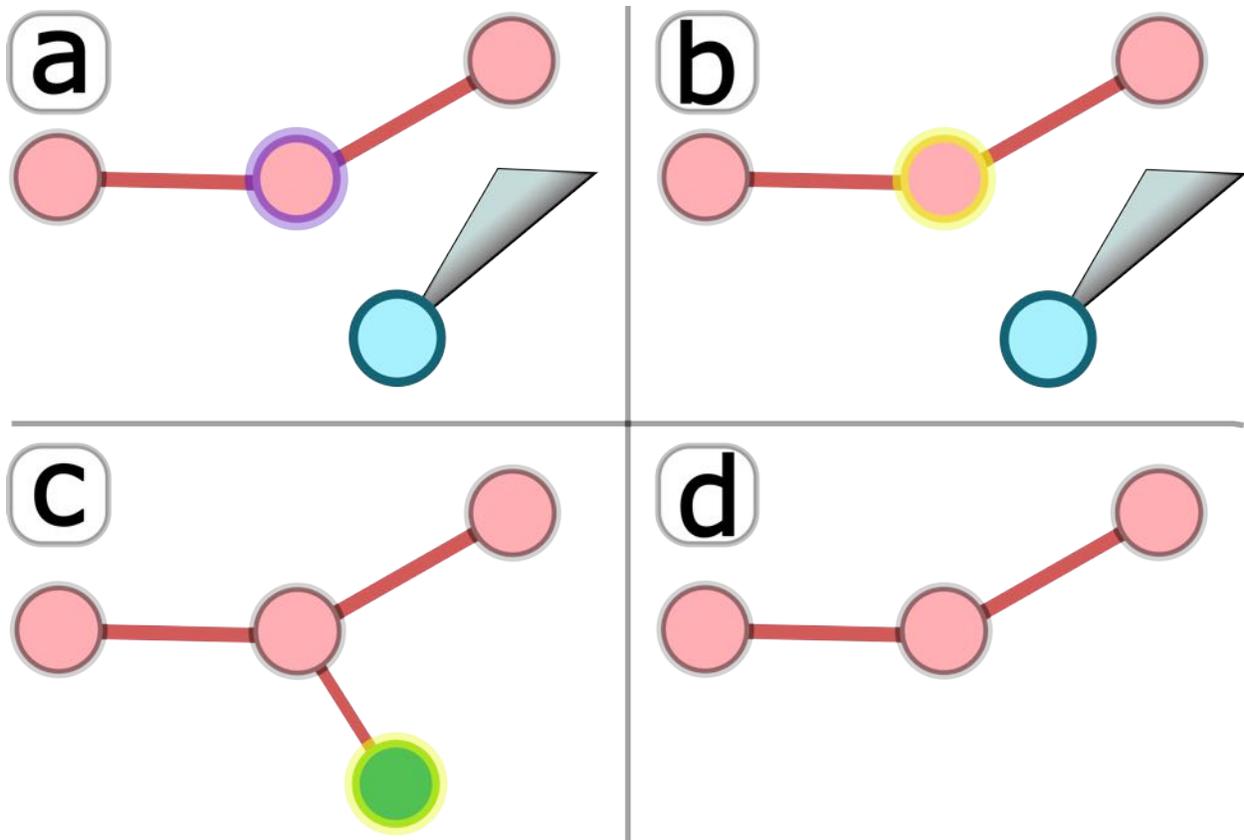


Figure 4.5: Example sequence of tracing. (a) A node (red) nearby the probe (blue) is highlighted with purple. (b) Selecting the nearby node will change the colour of the highlight to yellow. (c) A new node is created in the same place and with the same size as the probe. The colour group of the new node is from the selected colour group. An internode will connect the selected node to the new node. (d) Deleting the selected node will remove it and all the internodes connected to it.

A network starts by pressing the “add node” key. This creates a node with the same size as the probe, in the same place as the probe, and the same colour as the selected colour group. In addition, the node becomes selected if no other node or internode is selected. If another node is created while a node is selected, then an internode will connect the previously selected node and the new node. This new internode takes the colour and size from the selected colour group. This can create a skeleton quickly when paired with haptic assistance.

A node can also be selected by positioning the probe nearby, and then pressing the “select” button. A nearby node is indicated by a purple colour. When selected, the colour turns to yellow. A selected node can be moved to a new position, linked with a nearby node,

or deleted. When a selected node is deleted, then it is deleted with all of its connected internodes (Figure 4.5). Because internodes are selected much less often compared to nodes, an internode requires one of its nodes to be selected before it can be selected. Note that only one node or internode can be selected at a time. Selection of a node or internode is removed by pressing the “deselect” key.

### 4.1.3 Editing Annotations

A mistake is likely to occur while interactively annotating volumetric data. One option is to either edit or remove the existing annotation. For example, if a node is placed in the wrong location, then that node can be removed before creating a new one, or moved to the desired location. If a region is painted with the wrong colour, then the region can be painted over with the correct colour group. If the annotation was recently done, then it can be reverted quickly using the undo and redo functionalities. Undo reverses the last annotation made. For example, performing undo on a recently created node will remove it from scene. If a region was painted, then undo will revert the colour group of voxels to what it was before. If redo is performed, then the last undone action will be reverted.

Undo and redo functionalities are implemented using a command structure. A command structure consists of two stacks, undo and redo, that keep track of recent commands — performed actions, such as painting and tracing. Each command consists of instructions to execute, undo, and redo an action. When an action is executed, this is stored as a command in the undo stack. When undo is performed, undo instructions are executed from the most recently added command to the undo stack, and then this command is moved to the redo stack. Likewise, when redo is performed, then redo instructions are executed from the most recently added command to the redo stack, and then this command is moved back to the undo stack. If a new action is executed, then the redo stack is cleared.

<b>Execute</b>	<b>Undo</b>	<b>Redo</b>
Add Node	Set “deleted” flag for node	Clear “deleted” flag for node Select the node
Add Connected Node	Set “deleted” flag for node	Clear “deleted” flag for node and internode Select the node
Delete Node	Clear “deleted” flag for node Select the node	Set “deleted” flag for node
Move Node	Move to previous position	Move to back to new position
Add Internode	Set “deleted” flag for internode	Clear “deleted” flag for internode Select the internode
Delete internode	Clear “deleted” flag for internode Select the internode	Set “deleted” flag for internode
Change Group	Set colour of node or internode to the previous colour group	Set colour of node or internode to the new colour group

Table 4.1: List of commands for tracing. All commands have an execute, undo, and redo component.

Each command can be implemented such that it either loads a copy of the previous state, or performs the opposite of the previous action. For tracing, the previous state is stored in each command, and reverted when undo is performed (Table 4.1). With segmentation, however, swapping values between current and previous state can be too memory intensive to quickly perform undos and redos. Instead, the difference between the current and previous state is stored in a sub-volume. While a region is being painted, all the newly painted voxels and the difference between the current and previous colour group are stored in a list. When the user releases the “paint” button, a sub-volume is created to store all the information above. The dimension of this sub-volume corresponds to the largest cell coordinates of a painted voxel minus the smallest cell coordinates of a painted voxel. When undo is performed, the values within the sub-volume are added to the segmentation mask. When redo is performed, the values are subtracted instead.

## 4.2 Volumetric Rendering

Volumetric data contains a wealth of image features that may indicate the presence of a structure of interest. By assigning material properties to voxels, and rendering the scene with

light, the boundary between various structures can be perceived within volumetric data. This visibility of structures is largely dependent on the selection of material properties, which is guided by the user. In addition, rendering is a computationally expensive process. To provide an insightful and interactive system, quality of rendering needs to be balanced with performance. To address these issues, direct volumetric rendering is implemented, based on the implementation in (Hadwiger et al., 2009)<sup>2</sup>. It consists of three parts: material properties, lighting, and viewing.

### 4.2.1 Material Properties

Unlike real objects that already have well-defined boundaries, virtual objects are composed of data values in an image. These data values need to be mapped to some material properties to differentiate one object from another based on some image feature derived from data values. This is done using a transfer function.

A transfer function has an enormous number of potential material properties. If the user interface does not help guide the selection of control points on the transfer function, then the user has to rely on trial and error to find a viable transfer function to view structures of interest. Even if a viable transfer function is found, undesirable objects may be in the way of viewing a structure of interest (Kniss et al., 2002).

In SHVR, a transfer function is used to map intensity values within voxels to colour and opacity (Figure 4.6). The appearance of the volume changes immediately after the transfer function is modified, which helps to remove noise and artifacts, assign a colour and opacity based on the intensity value of objects, and accentuate the differences between structures within volumetric data. It is drawn on a 2D window, where the x-axis corresponds to

---

<sup>2</sup>This implementation of direct volumetric rendering also contains these features: local ambient occlusion, distance colour blending, viewing modes, and density painting.



Figure 4.6: (left) The volume coloured by a transfer function. (right) The transfer function widget consists of a histogram of all intensity values within volumetric data, editable control points, and lines that interpolate the colour and opacity in between each successive control point.

intensity value of voxels, and the y-axis corresponds to opacity. It consists of a histogram, and control points.

The histogram shows the distribution of intensity values within the volumetric data. When a peak is seen in the histogram, it shows the most common intensity value. This value most likely corresponds to background voxels, which should be avoided while defining the boundaries of an object. Control points define the colour and opacity at a given intensity value. The colour and opacity of an intensity value are linearly interpolated based on the distance from the two control points nearby. If an intensity value does not have two control points nearby, that value does not become rendered. Each control point can be added, deleted, edited with a new colour, and dragged to a new position on the transfer function widget. In addition, all control points can be shifted left or right using the mouse wheel. For precise adjustment, a modifier key can be held down to slow the rate of change while shifting control points.

## Density Painting

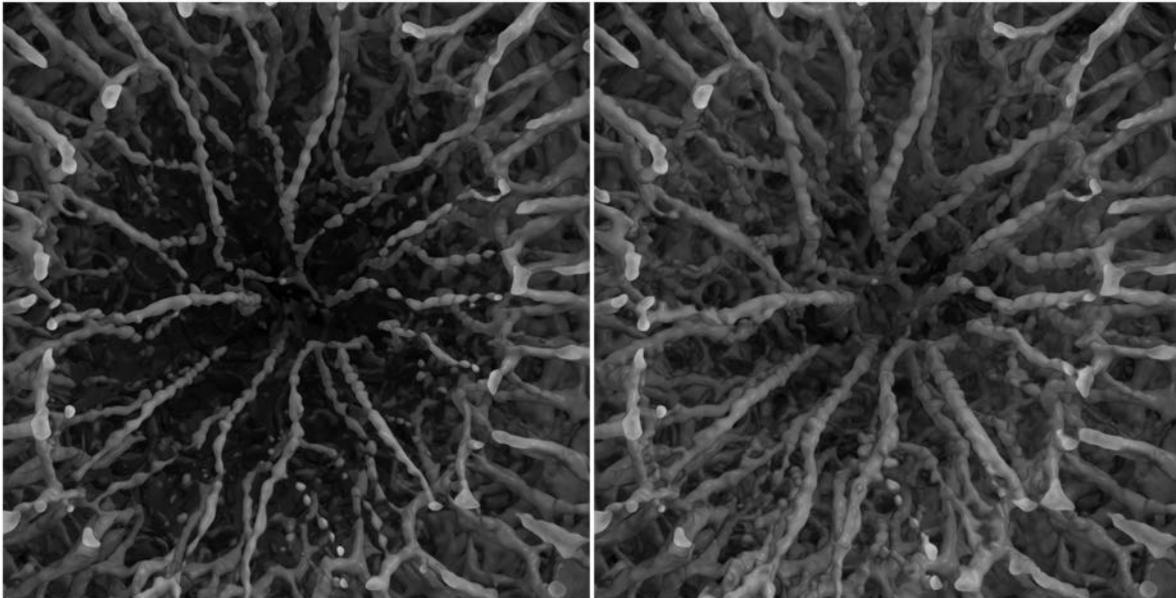


Figure 4.7: Example of density painting. It increases the intensity values within a local region to view structures that may not be visible, or difficult to perceive with the selected transfer function.

Within a sample of a developing flower head, its structures in different regions may be at different stages of growth. For example, the florets at the edge of flower head are more mature than the florets at the center. As a floret matures, its vascular bundles mature as well. When a transfer function is tuned to visualize the vascular bundles at the edge of flower head, it cannot visualize the vascular bundles at the center of flower head due to the low intensity value of the vascular bundles. But if the transfer function is shifted to visualize the vascular bundles at the center, then the vascular bundles at the edge become too cluttered from the surrounding ground tissue, which makes the connections between neighbouring vascular bundles challenging to see. To address this issue, the lower intensity value within a structure is raised to be visible with higher intensity structures using density painting (Figure 4.7). This was inspired by a method that enhances contrast of inhomogeneous tubular structures by locally shifting the transfer function for selected voxels (Lathen et al., 2012).

Density painting adds a non-destructive offset value to voxels within a local region. This is executed at the same time as voxel painting, and reuses information about which voxels are to be labeled. Information about how much offset is applied to each voxel is maintained as separate offset data. This offset data has the same dimensions, and stores the same amount of data per voxel as the image data<sup>3</sup>. Overflows are prevented by limiting the painted offset value to the minimum and maximum possible intensity value within the image data. When a region is painted, an offset value is added to both the image data and the offset data. This is in contrast to voxel painting which only paints on a segmentation mask.

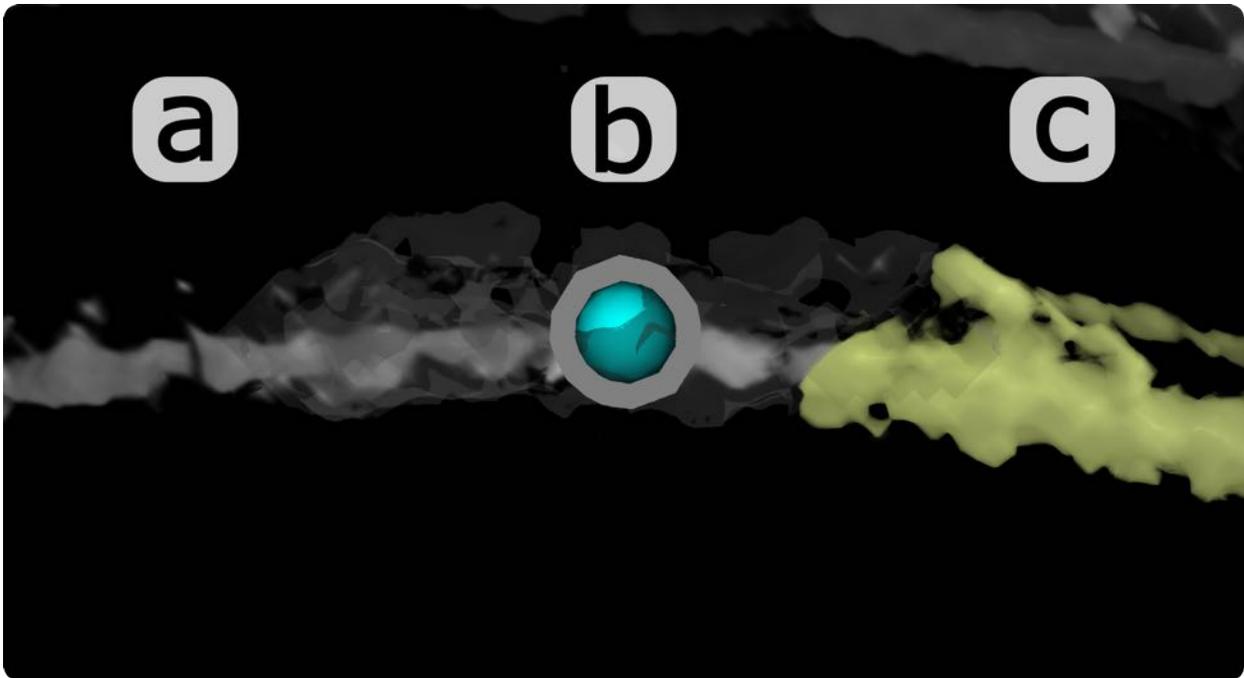


Figure 4.8: Example of density painting around the probe. (a) Standard volumetric rendering; (b) Thin shell iso-surface is formed within the local region surrounding the probe based on the paint threshold value. (c) When the region is painted, the intensity value of voxels within the region also increases.

The painted offset value can be set independently, or automatically set as the difference between the minimum transfer function value and the paint threshold value. This ensures that whatever voxel that is painted becomes visible with the current transfer function. This

<sup>3</sup>In current implementation, the offset volume actually stores more data per voxel than the image data. This was done to apply negative offsets, but positive offsets were enough to do all of the desired tasks.

difference in intensity value can be visualized using thin-shell iso-surfaces — surfaces of volumetric objects where all voxels within the object are assumed to have homogeneous intensity. This forms a translucent surface around visible objects to show how a region would change after density painting (Figure 4.8). In addition, the painted offset value can be set to a flat value, or a value that is based on the distance from the probe.

Similar to voxel painting, density painting can be removed, edited, or undone. Removing offset value from the volume is done by painting the region with the colour group “0”. This will reset the offset values within a region by reversing the amount of offset in the offset data from the image data. When a new offset value is painted on top of a region with existing offset values, then the highest offset value that is less than the painted offset value will be applied. This means that if a region has higher offset value than the painted offset value, then the offset values within the region will be lowered to match the painted offset value. But if a region has lower offset value, then the higher offset value will be selected. This is useful when painting the region with an offset value based on the distance from the probe.

To undo or redo density painting within a region, a command structure is used similar to voxel painting. Every time a region is painted, the difference between the new offset value and the current image intensity value is stored in the offset data, and in a new sub-volume. When this difference is added to the existing offset value in the offset data, the total offset value equals the amount of offset in the image data. The new sub-volume has the same size as the sub-volume used for voxel painting. When either undo or redo is called, both voxel painting and density painting are edited at the same time.

## 4.2.2 Lighting

Virtual objects need to be rendered with light to make their form believable, and their appearance visually pleasing. In addition, the shading model needs to be efficient enough to

render in real time. This process involves finding which parts of the volume are visible, and how much light is visible on each surface point. These are achieved through a combination of ray-casting and the Blinn-Phong shading model (Blinn, 1977).

## Ray-casting

Ray-casting determines what is visible in a virtual scene by tracing the path of light entering the camera. It casts a ray through the center of each pixel in the direction of the camera. These rays travel in equidistance steps to sample optical properties, such as colour and opacity, at a sampled position in the volume. The optical property is obtained from either the transfer function, or the colour group. As the ray travels through the scene, it accumulates the optical properties of sample positions within the volume. When the ray reaches maximum opacity, it is then output as a colour for a pixel in the screen. This is represented by the formula:

$$c = \sum_{i=0}^n c_i \prod_{j=0}^{i-1} a_j \quad (4.1)$$

where  $c$  is the colour of the output pixel,  $n$  is the total number of non-transparent sampled positions,  $c_i$  is the colour at the sampled position, and  $a_i$  is the opacity of the sampled position. Opacity correlates to how much light is reflected at a sampled location <sup>4</sup>. If the opacity of a sampled position is opaque, then the output pixel colour will only consist of the colour at that sampled position. On the other hand, if the opacity is transparent, then light will be transmissive, and the output colour will not be impacted by that sampled location. It is used to approximate optical depth such that the colour of sampled positions closer to

---

<sup>4</sup>(Hadwiger et al., 2009) correlates opacity to absorption. However, the term “opacity” refers to how transmissive the light is through an object. Light has more difficulty passing through an object as it becomes more opaque. Since light in SHVR does not become absorbed or emitted in this shader model, the amount of light reflected from a sampled position is the most accurate description.

the camera are prioritized higher than positions further away. This process is similar to alpha-blending — combining one image with another using the alpha channel — to weigh output colour based on distance from camera.

By adjusting how the rays are sampled, internal structures within the volume can be explored. This is done by using clipping planes and cutting planes. Clipping planes are the planes in front of the camera that define where the rays starts and ends. All parts of the volume before the near clipping plane, and beyond the far clipping plane are not rendered. Cutting planes, on the other hand, are planes that are aligned with faces of the volume. All parts of the volume that are outside of the cutting planes are not sampled for optical properties.

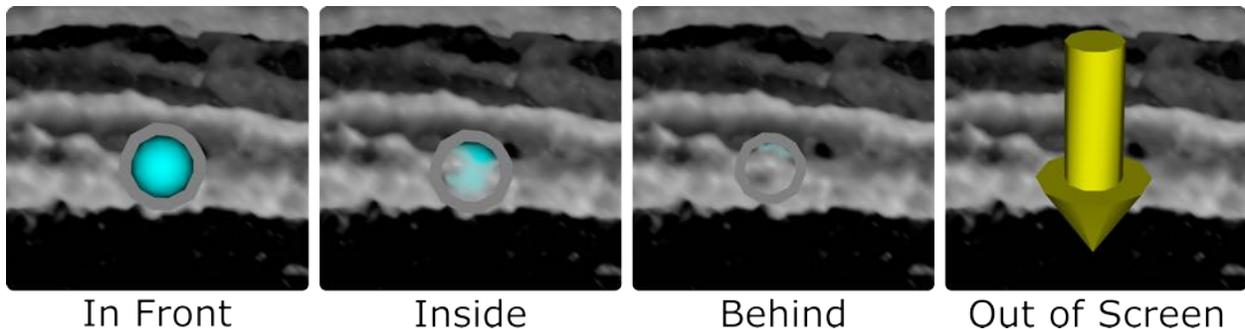


Figure 4.9: Cases of the probe interacting with volumetric objects. A persistent silhouette surrounds the probe to help find when it is occluded by other objects. In addition, an arrow appears in the middle of the screen when the probe is out of screen.

Ray-casting is also used to render mesh objects alongside volumetric objects. Mesh objects are objects made from vertices and triangles. These include nodes, internodes, and the probe. Rays do not travel beyond the front face of a mesh object to give the appearance of an opaque surface. By showing whether a mesh object is in front of a volumetric object or not, a depth cue is reinforced. This provides an important visual point of interaction to know where the probe is inside of the volume (Figure 4.9).

Some mesh objects need to be overlaid on top of volumetric data. This is necessary when the probe becomes lost when it is occluded by volumetric objects in front of it, or goes

outside of the screen. To address these problems, the probe is surrounded by a persistent silhouette, and an arrow pointing to the probe appears in the center of the screen when the probe moves outside of the screen. To always view these mesh objects, these objects are rendered on top of existing render.

Traversing each ray can be expensive in computation cost due to the high number of pixels. To reduce the amount of computation performed, the number of steps traversed by a ray is reduced. The most natural way to reduce the number of steps is to terminate the ray if its accumulated opacity reaches the maximum value ( $\prod_{i=0}^{n_{curr}} a_i \geq a_{max}$ ). Another way is to terminate when the a ray reaches one of the stopping points along a ray: the front face of mesh object, the back face of the volume, or the far clipping plane. In addition, a constant maximum number of steps is set to limit the amount of calculation per ray, which may be necessary if the volume is much larger than the step size. In implementation, the maximum number of steps is set to 4000 steps. Furthermore, the distance from the camera to the volume is skipped by starting each ray from either the front-face of the volume, or the the near clipping plane. This is chosen based on whichever starting position is further from the camera. process is accelerated using the GPU by computing the casted rays in parallel.

To further enhance performance, the rendering process is accelerated by computing the casted rays in parallel in the GPU (Figure 4.10). The rendering is performed on a lower resolution texture that has the same aspect ratio as the screen to reduce the number of pixels rendered. This texture is then stretched to fit the screen. The benefit of this method is that an image with half of the width and height of the original dimension has only around a quarter of the original pixel count. For interactive systems, maintaining a smooth frame rate takes higher priority compared to image quality. The idea is to use the graphics window as a viewfinder, and then take a higher resolution screenshot separately. In addition, the update rate of rendering is capped to a constant frequency to avoid a jarring difference in performance when moving from one region to another. By default, 70% of original width

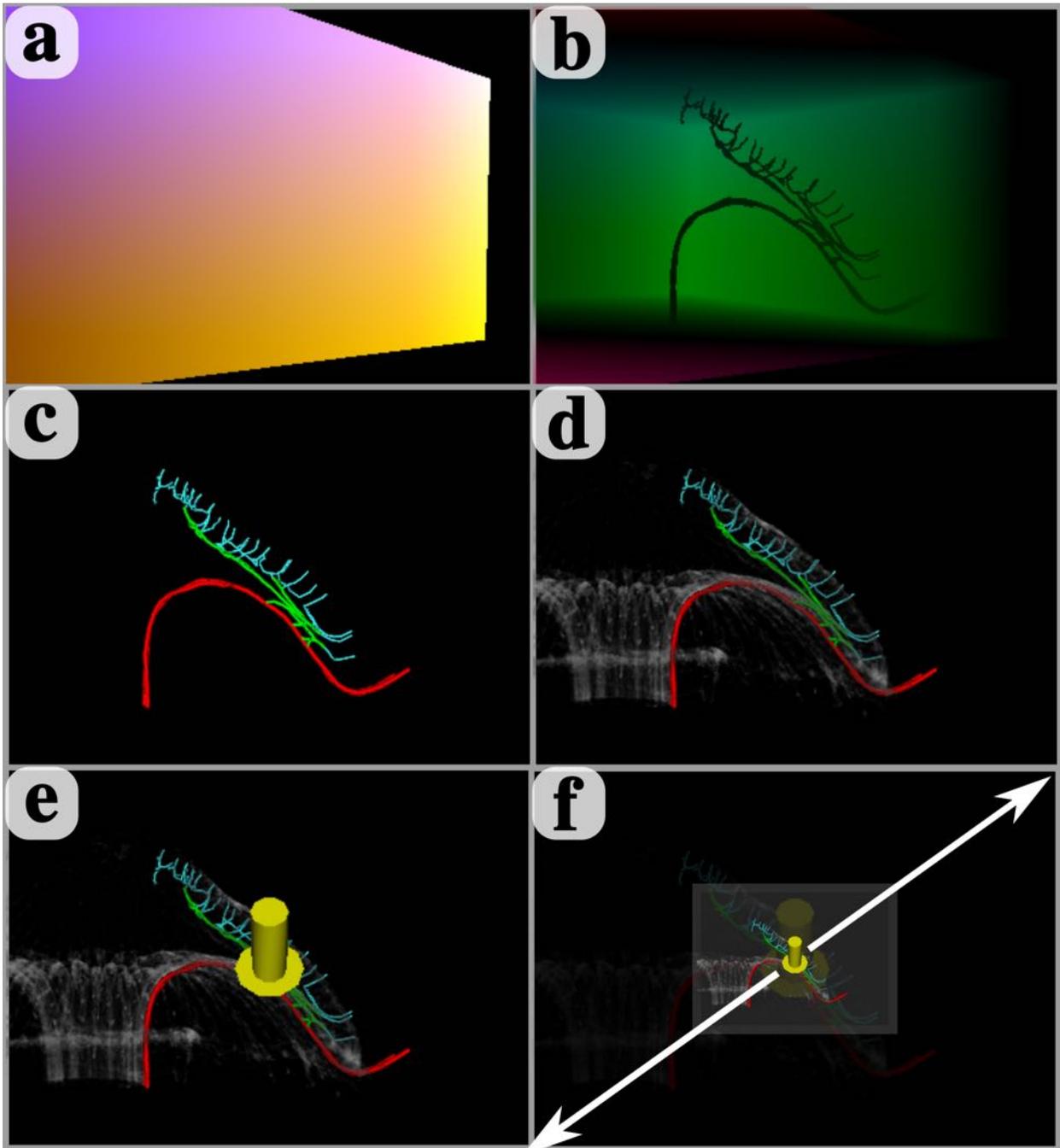


Figure 4.10: Rendering process in the GPU. (a) The starting position, (b) the direction and length of rays are pre-computed for volumetric rendering. The colours on the textures correspond to the starting position and direction of rays, respectively. (c) Mesh objects are rendered first, followed by (d) volumetric rendering, and then (e) mesh objects that are on top of existing rendering, such as the arrow to find the probe. Finally, (f) the output image is stretched to fit the screen.

and height is rendered, and the update rate is capped at 30Hz.

## Blinn-Phong Shading

With ray-casting, objects in the path of the camera can be detected. Blinn-Phong shading model (Blinn, 1977) simulates light on the virtual objects to perceive its shape and form. It is an improvement to Phong shading model (Phong, 1975) to render curved surfaces. To calculate the amount of light on each surface point, this model calculates and sums these three components of light: diffuse, specular, and ambient light.

Diffuse light is the light that is scattered evenly on surfaces in all directions. This is modelled after Lambert's law, which states that the intensity of reflected light is relative to the angle of incidence — the angle between the surface normal, the direction perpendicular to a surface, and the direction toward the light source from a surface point. For the current position of a viewing ray  $\vec{x}$  with the surface normal  $|\nabla f(\vec{x})|$  which is calculated as the negative direction of the gradient, and normalized light vector  $\vec{L}$ , diffuse light  $l_d$  is calculated as:

$$l_d = \max(|\nabla f(\vec{x})| \cdot \vec{L}, 0) \quad (4.2)$$

Specular light is the light that reflects back to the camera when light hits a surface, which appears as a bright spot. The position of this light depends on view direction, where the viewer is looking at the surface point from. It incorporates a half-way vector  $\vec{H}$  between the view direction  $\vec{V}$ , and light direction  $\vec{L}$  and a constant value  $\alpha$  to influence the shape of the highlight. As  $\alpha$  becomes larger, the highlight transitions from a smooth gradient over a large area to a smaller sharper highlight. Specular light  $l_s$  is written as:

$$\vec{H} = \frac{\vec{V} + \vec{L}}{2} \quad (4.3)$$

$$l_s = \max(|\nabla f(\vec{x})| \cdot \vec{H}, 0)^\alpha \quad (4.4)$$

Finally, ambient light is the indirect light that reaches the surface of objects. Without any ambient lighting, surfaces facing away from the light source will receive no light, and appear as black. In Blinn-Phong shading, this is approximated as a constant value. This assumes that all surfaces are illuminated uniformly in all directions regardless of objects nearby. Putting all the terms together, the colour of ray casted pixel  $\vec{c}_v$  is:

$$\vec{c}_v = l_a \cdot \vec{c}_m + l_d \cdot \vec{c}_m + l_s \cdot \vec{c}_l \quad (4.5)$$

where  $l_a$  is intensity of ambient light,  $l_d$  is intensity of diffuse reflection,  $l_s$  specular highlight,  $\vec{c}_m$  is material colour, and  $\vec{c}_l$  is light colour. Note that the segmentation mask is also sampled at a sampled position. If the sampled voxel is assigned a colour group, then it will refer to the material property from the colour group rather than from the transfer function.

### Local Ambient Occlusion

When the ambient light is approximated as a constant value, structures lose visual cues to perceive spatial information. In particular, features within a structure, such as bumps, nooks, and crannies, appear flat, and the distance between structures is not conveyed (Figure 4.13a). By calculating how much light reaches each surface point, structures can be perceived more vividly and realistically. To efficiently provide ambient light, local ambient occlusion is implemented based on (Hernell et al., 2007) and (Hernell et al., 2010).

Local ambient occlusion estimates the amount of light reachable at each voxel position by averaging the number of amount of light that is reachable within a local spherical neighbourhood. This darkens regions based on opacity of nearby objects and proximity to nearby

structures. The intensity of incident light  $I_k(x)$  arriving at a voxel location  $x$  from one of the ray directions  $k$  within a spherical neighbourhood is described as:

$$I_k(x) = \sum_{m=0}^M \frac{1}{M} \prod_{i=0}^{m-1} (1 - \alpha_i) \quad (4.6)$$

where  $M$  is the number of samples taken along the ray, and  $\alpha_i$  is the opacity at the sampled position according to the current transfer function. The direction of incident light ray goes from a voxel position to the light source. This means that the intensity of a light ray decreases as the light ray passes through translucent objects. If a light ray encounters an opaque object, then the intensity of light will stop increasing from taking more samples. Note that no surface point is completely darkened, even if it is inside of a fully enclosed space. This tends to be a desirable feature to perceive structures within enclosed spaces, even if the lighting condition isn't realistic. In addition, only the incident rays  $\vec{k}$  that are in the direction of the surface normal  $\vec{n}$  such that  $\vec{k} \cdot \vec{n} > 0$  are used within a spherical neighbourhood. This reduces the number of rays by half, and excludes light rays that would go towards the surface. To obtain the intensity of local ambient occlusion,  $I(x)$ , at voxel position  $x$ , all of the incident light rays  $I_k(x)$  are averaged such that:

$$I(x) = \frac{\sum_{k=0}^K I_k(x)}{K} \quad (4.7)$$

where  $K$  is the total number of incident light rays around a spherical neighbourhood. Then, this value is mapped to ambient occlusion mapping to intensify bright and dark regions from ambient occlusion. The final output value replaces intensity of ambient light  $l_a$  in direct volumetric rendering. To use local ambient occlusion at interactive frame rates, it is calculated and stored in GPU. This is done by calculating local ambient occlusion in parallel for each voxel location. This reuses the image data that is already in the GPU. When the

calculation is finished, the result is stored in the GPU, where it will be used to provide ambient light to the volume (Figure 4.11). Since ambient occlusion depends on the opacity and shape of structures, ambient occlusion is updated when the transfer function is modified. Updates can be done manually, or automatically whenever the transfer function is changed.

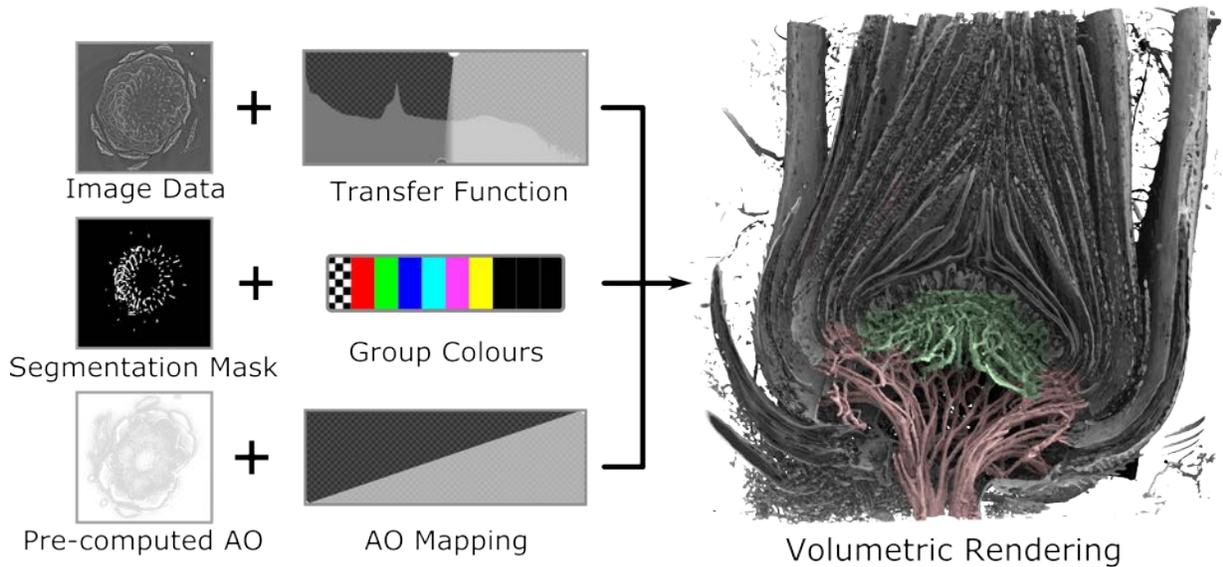


Figure 4.11: Volume rendering uses a combination of 3D image data, segmentation mask data, and ambient occlusion data. These datasets are stored in the GPU, and mapped to colour and opacity based on a transfer function, group colour settings, and ambient occlusion mapping respectively.

### Distance Colour Blending

Exploring dense structures rely on depth cues to identify which structure is in front of another. This cue may be given by the ordering of objects such that the closer object is in front of the distant object, or from perspective lines where several straight lines converge over distance. A clear depth cue helps understand the shape of individual structures, and the spatial relationship between structures. However, these methods are not useful to perceive depth cues in dense tubular structures due to numerous overlaps and a lack of straight lines. To address this problem, Distance colour blending is based on the implementation in

(Rheingans and Ebert, 2001).

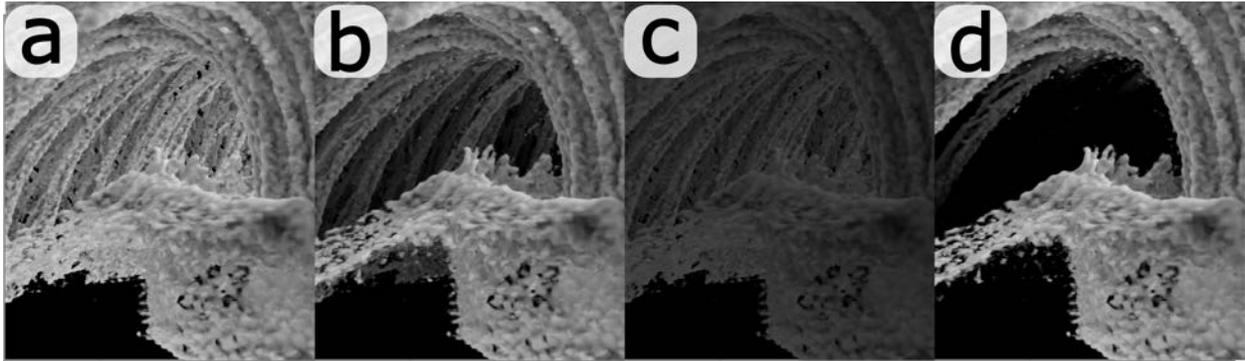


Figure 4.12: Results from distance colour blending. (a) No blending. (b) Smooth transition from volume colour to background colour. (c) Constant level of blending. (d) Sudden transition from volume colour to background colour.

Distance colour blending gradually decreases the intensity of the volume colour while increasing the intensity of background colour as the depth into the volume increases. This appears similar to viewing a scene through a haze. Two controls are given to the user: blending size, and blending rate. Blending size controls the space where blending occurs. When the blending size is small, then blending starts close to the viewer. If the blending size is large, then blending starts further away from the viewer. Blending rate controls the progression from volume colour to background colour. It can change the progression to logarithmic, linear, or exponential. If the blending rate is zero, then the volume colour never becomes replaced by the background colour. But if the blending rate is high, then volume colour switches to background colour almost immediately. Note that blending size can perform similar functionality as blending rate. But by adjusting both parameters, the starting location and the rate of progression can be tuned to visualize a structure of interest (Figure 4.12). To disable distance colour blending, the blend size can be set to zero. This is represented by the formula:

$$\vec{c}_d = (1 - k_{ds}d_v^{k_{de}})\vec{c}_v + k_{ds}d_v^{k_{de}}\vec{c}_b \quad (4.8)$$

where  $k_{ds}$  controls the blending size,  $k_{de}$  controls the blending rate,  $d_v$  is the normalized distance through the volume,  $c_v$  is the volume colour, and  $c_b$  is the background colour. By combining local ambient occlusion with depth colour blending, the dense tubular structures can be seen with more vividly within image (Figure 4.13d).

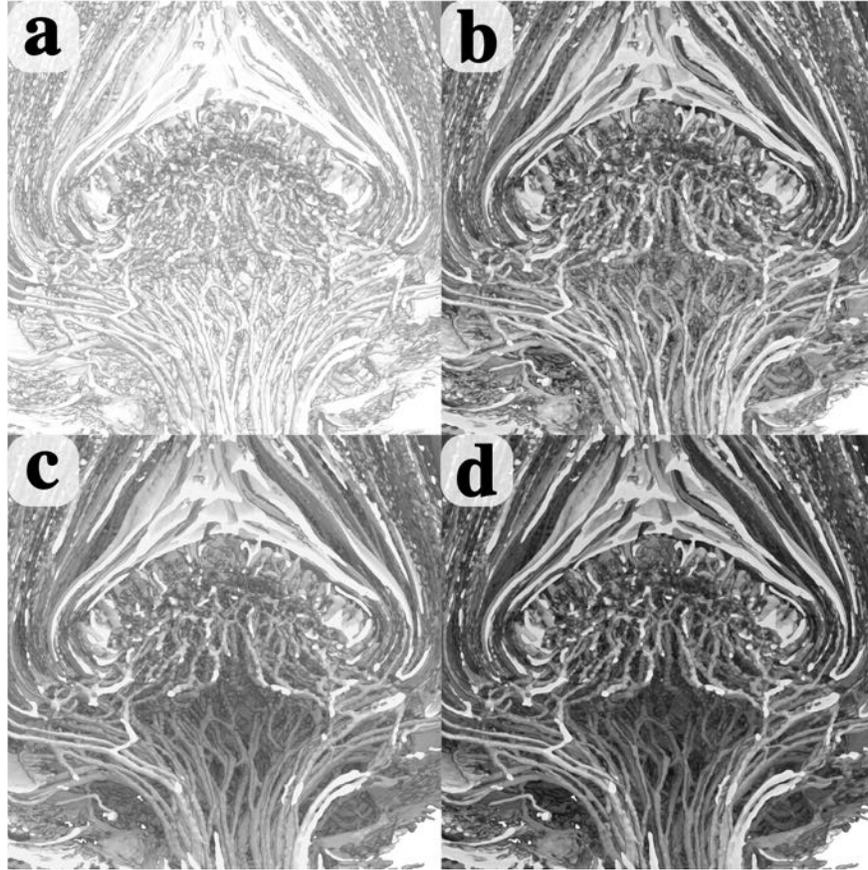


Figure 4.13: Comparisons of rendering with visual enhancements. (a) Without visual enhancements, the structures within volumetric data can be hard to perceive. (b) Ambient occlusion darkens objects that are in close proximity to each other to differentiate between tight gaps and open spaces. (c) Distance colour blending helps differentiate nearby objects from distant objects. (d) Combining ambient occlusion and distance colour blending provides depth cues and contact shadows to help identify structures within volumetric data.

### 4.2.3 Viewing

To become familiar with the contents of a sample, the user has to explore the volume to see structure of interests from various viewpoints, and to avoid objects that are obstructing the

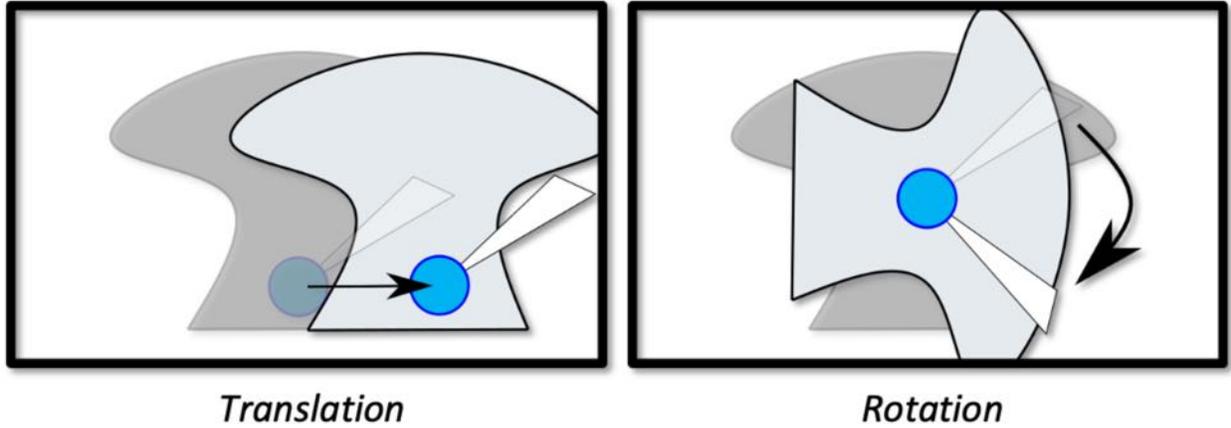


Figure 4.14: Moving the volume using the probe. When the “move” button is pressed, the movement of the volume matches the movement of the probe. The movement of the volume can be scaled by scaling the movement of the probe.

view. In SHVR, the view is updated by moving the volume in front of the camera, while the camera remains stationary. The volume is arranged in the virtual world using a transformation matrix, which consists of translation, rotation, and scaling. When the “move” button from the haptic device is pressed, the starting transformation matrix for the probe and the volume is stored. Then, the difference in movement between the current and the starting transformation matrix of the probe is extracted to be applied to the starting transformation matrix of the volume. Finally, this transformation matrix updates the position and orientation of the volume. This is represented by the formula:

$$M_{volume} = M_{probe}M_{probe\_start}^{-1}M_{volume\_start} \quad (4.9)$$

where  $M_{volume}$  is the current transformation matrix of the volume,  $M_{volume\_start}$  is the start transformation matrix of the volume,  $M_{probe}$  is the current transformation matrix of the probe, and  $M_{probe\_start}$  is the start transformation matrix of the probe. This matches the movement of the volume to the movement of the probe (Figure 4.14).

Interaction with volumetric data may occur at different scales. For example, exploring

vascular bundles requires moving the volume less than exploring the entire sample. However, the workspace — the limited region within a haptic device where the stylus can move around — is not large enough to accommodate interaction at different scales. By digitally scaling the movement of the probe relative to the movement of the stylus, the probe can move further within the virtual world. This can be used to accurately interact with nearby objects, quickly interact with distant objects, and move the volume over a longer distance. This workspace scale can be controlled by a keyboard shortcut, or through a settings widget.

In addition to the haptic device, the position and orientation of the camera can be adjusted using the keyboard. This moves the volume with fixed step sizes to precise adjustment. This is useful when framing structure of interest before taking an image, avoiding occluding objects, or changing viewpoints. Six keys (W,S,A,D,Q,E) correspond to forward, and backward, left, right, up, and down movement of the volume respectively. To rotate the volume around the camera, “Alt” modifier is held down while pressing one of the above six keys. This rotates the volume around one of three orthogonal axis of the camera. Both the “Shift” and “Alt” modifiers can be pressed together to slow down the rate of rotation of the volume.

## **Viewing Modes**

Various viewing modes help explore the data by changing what is shown on the screen. There are three viewing modes: full screen mode, stereoscopic mode, and overview mode (Figure 4.15). The full screen mode shows the view from the main camera. The stereoscopic mode shows an offset image separately to each eye. This is implemented by positioning the left-eye camera apart from the right eye camera. Then, each camera is tilted slightly towards each other such that the view directions converge at a single point. This informs the viewer about how far away objects are while wearing a pair of stereoscopic glasses and looking at a stereoscopic display .

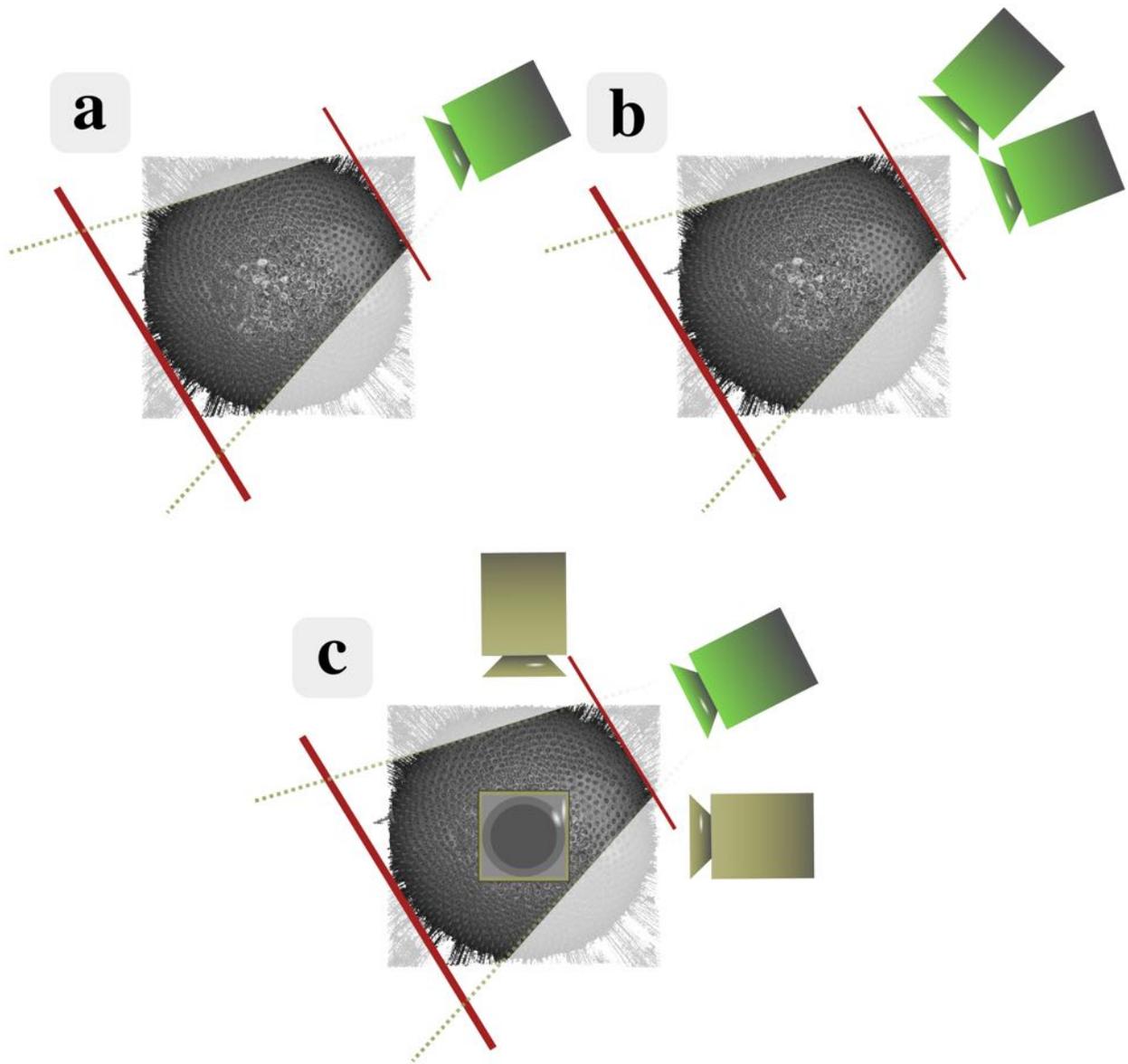


Figure 4.15: Camera position for each viewing modes. There are three viewing modes: (a) full screen, (b) stereoscopic, and (c) overview mode. The visible region is contained within the near and far clipping plane (red lines) in front of the main camera (green). Overview mode has three orthographic cameras (yellow) located at the top, side, and front of the volume.

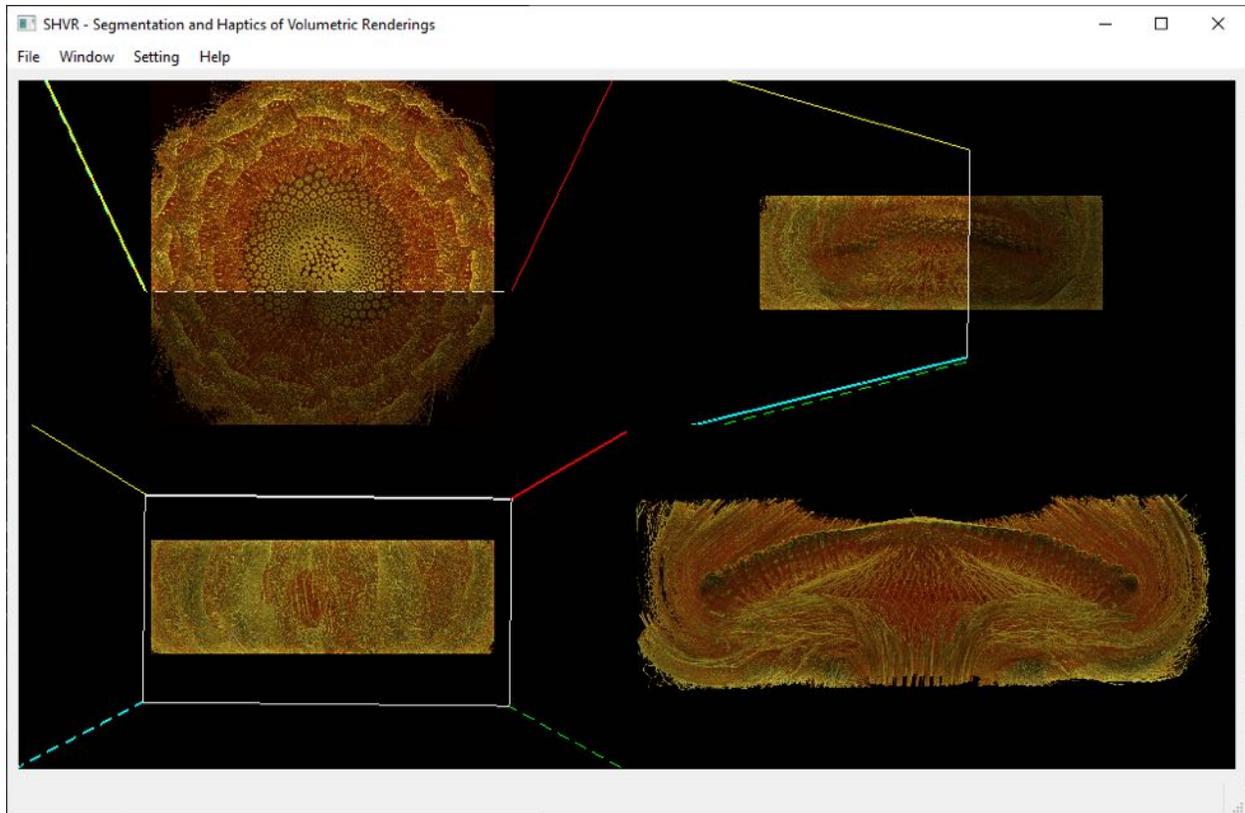


Figure 4.16: Overview Mode. This provides a view from the top, side, bottom, and main camera. The colour of the lines label each edge of the viewing frustum. Edges that are closer to the camera are drawn solid, while edges further from the camera are drawn stippled.

The overview mode presents four separate views: top, front, side, and camera views. These are shown by dividing the screen into four equally sized quadrants. As their names suggest, the top, front, and side views are positioned at the top, front, and side of the volume. These three views use orthographic projection to see the entire volume within each view. This is unlike the main camera view, which uses perspective projection, to see a region within the volume with an arbitrary camera position and orientation (Figure 4.16).

Within these views, the position and orientation of where the main camera is viewing is represented as coloured frustums in the top, front, and side orthographic views. The frustum is colour coded to identify individual edges. In addition, the edges are textured to know which edge is closer to the viewing camera. The frustum can be moved orthogonal to the view direction of the viewing camera by dragging the mouse within one of the views while

holding down the left mouse button, or along the view direction by scrolling with the mouse wheel. This allows the main camera to quickly view a region of interest. Note that this method of movement has limited amount of accuracy and cannot be used to orient the volume<sup>5</sup>.

Overview mode allows the user to quickly see which portion of the volume is in view of the main camera. Scenes can change abruptly while moving within a sample, which can be disorienting to the user. By providing a map of where the user is viewing, the user can quickly go back to exploring without feeling lost. A similar idea is used in (Usher et al., 2018), where a mini-map is placed at the corner of the world in virtual reality to help users orient themselves in the dataset. This map keeps track of existing tracing annotation, and locates the subregion within the volume as the user explores around.

### 4.3 User Interface

The user interface is where the user and computer interact and communicate with each other. This is done through a device, such as a mouse, display, keyboard, and haptic device (Figure 4.17), or through an application. This is used to effectively communicate the content of an application with the user (Fred Churchville, n.d.).

In SHVR, the application consists of a menu bar at the top, a large graphics window in the center, and a status bar at the bottom of the application (Figure 4.18). The menu bar contains menu items to save and load settings, and setting widgets to adjust settings within SHVR. The setting widgets display the current setting values, which may be changed from user input. These settings include visualization, haptics, and performance settings. The graphic window renders the virtual world, which consists of the volume, the probe,

---

<sup>5</sup>Orientation of volume was implemented, but understanding how the frustums were rotating based on the movement of the mouse was too challenging. As a result, this feature was removed.

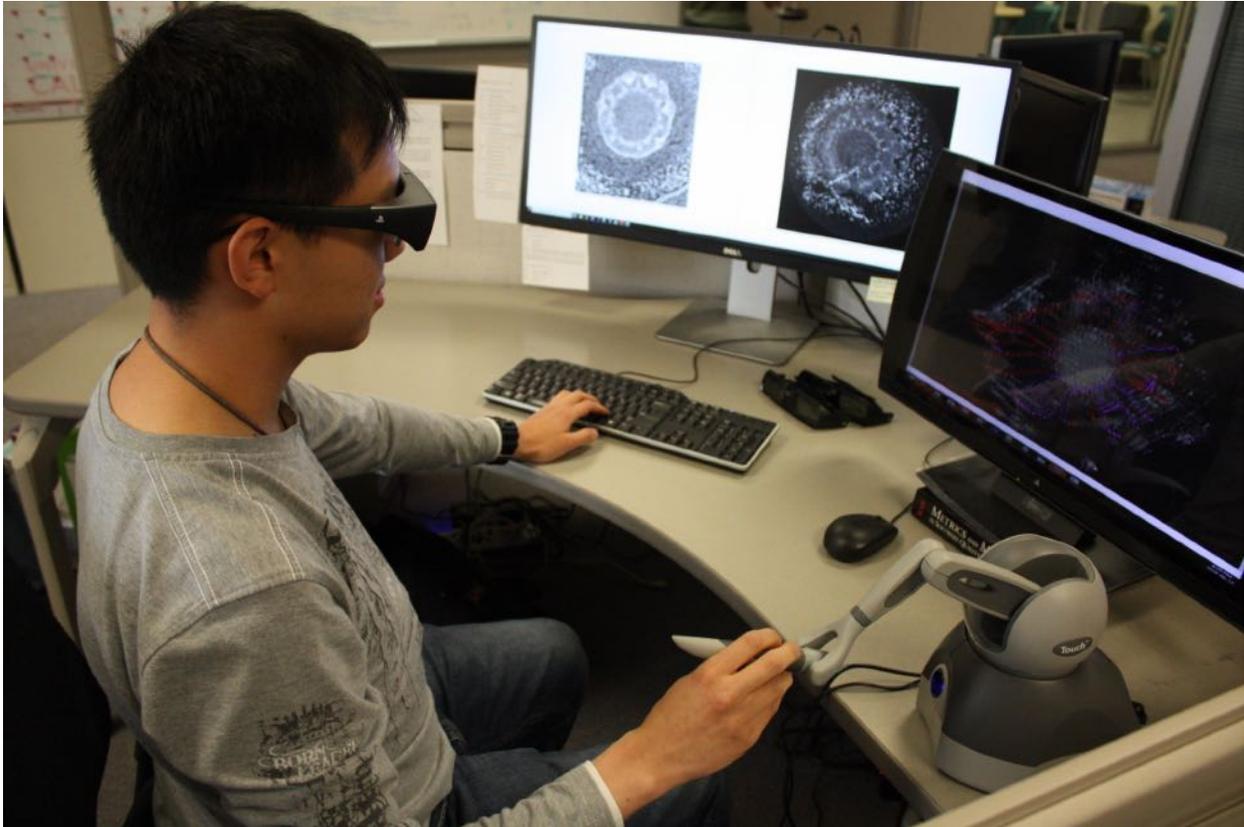


Figure 4.17: Setup for SHVR. Users use a combination of Touch<sup>TM</sup> haptic device (3D Systems, n.d.), a keyboard, and a mouse to interact with SHVR. They also have the option to use the Playstation 3D Display for stereoscopic viewing.

and annotations. This is where the user looks most of the time to annotate and visualize structures of interest. The status bar records the most recent action performed on the system. For example, when the size of the probe is adjusted, or the haptic becomes enabled, a line of text is displayed in the status bar to restate the performed action. See Appendix A for full list setting widgets.

In addition to using the menu items, image data files and saved setting files can be loaded by dragging them on top of the application window, or by loading a configuration file that contains the location of all the desired files to load. This allows the user to quickly change multiple settings and image data files at once instead of adding them one at a time.

SHVR is designed to be used with a 6 DOF kinesthetic haptic device with the right hand,

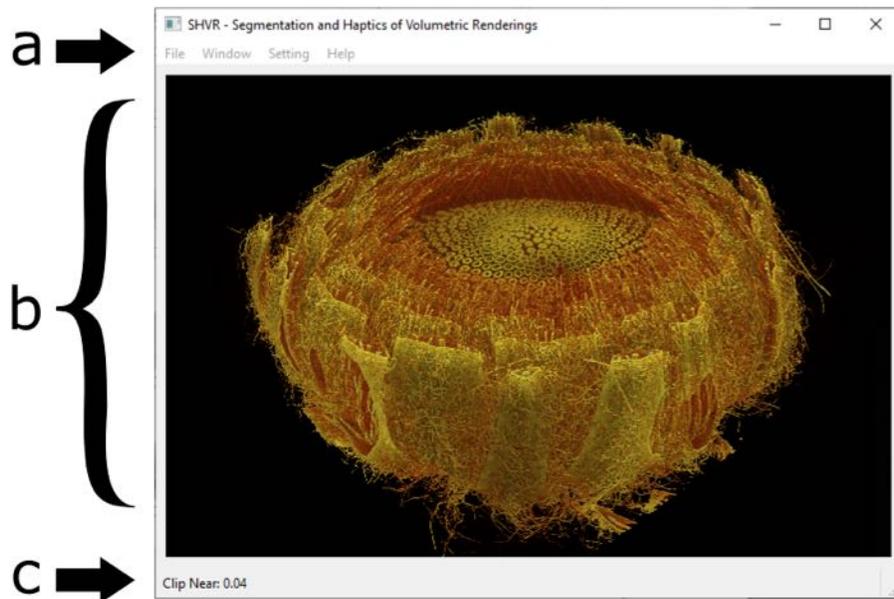


Figure 4.18: The main window consists of (a) a menu bar, (b) a graphics window, and (c) a status bar.

and a keyboard placed on the left hand side. A haptic device is operated using a stylus with two buttons. This stylus controls the position and rotation of the probe by moving within a limited workspace. The two buttons are assigned to moving the volume, and painting a region of interest. On the keyboard, many of the keys are assigned specific functions to quickly perform common operations, such as changing the selected colour group, moving the clipping planes, or toggling haptics. Because remembering the functionality of each key can be overwhelming, all operations, except for changing the view and annotating structures of interest, can be done from the menu. (Figure 4.19) contains the map of functions assigned to each key. Since both devices can be used with either hand, their positions can be swapped if desired. See Appendix C for complete list of keyboard shortcuts.

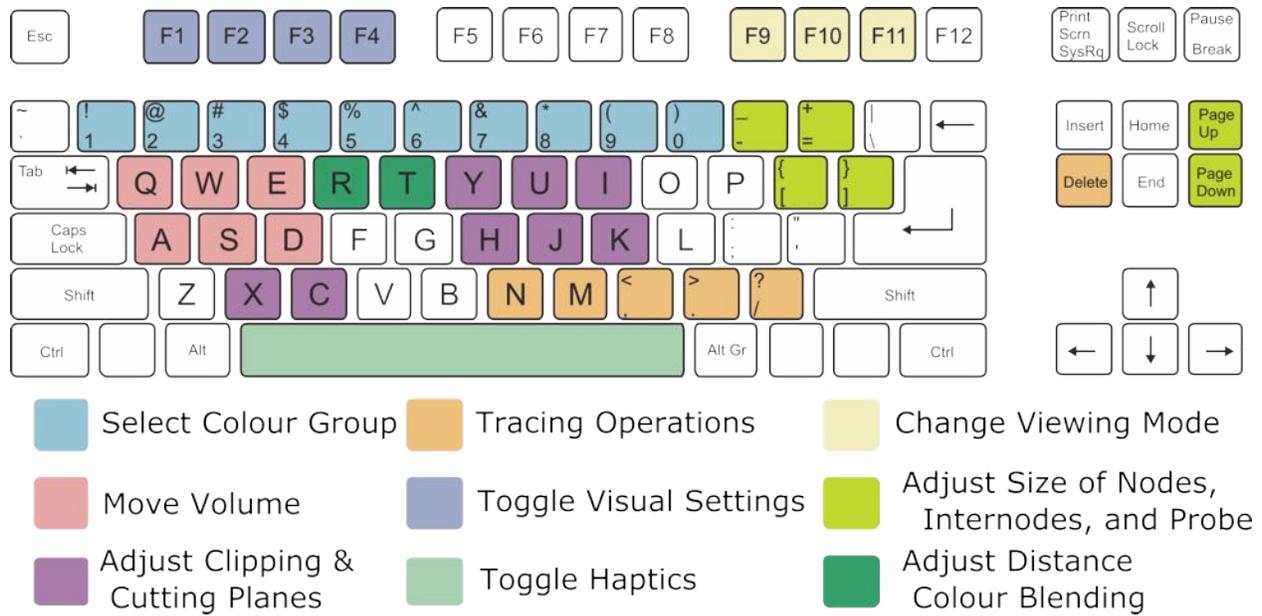


Figure 4.19: Commonly used keyboard shortcuts. The keys are grouped by similarity in functionality. Permission to use the base keyboard layout is given by CC BY-SA 3.0 license.

The mouse is used to perform operations that are less commonly performed. This involves using the menu bar, and adjusting the parameters from the setting widgets. This may be used to change parameters more precisely when making small changes, or more quickly when making large changes. This is also a useful alternative to keyboards, which do not have labels directly on the keys to inform what each key does.

## 4.4 System Implementation

This application is implemented using OpenGL (Khronos Group, n.d.) for graphics rendering, Chai3D (Barbagli et al., n.d.) for haptics rendering, and Qt (The Qt Company, n.d.) for GUI library. Because the user interacts with the system in real time, graphics rendering and haptics rendering need to run fast at a consistent rate for smooth and uninterrupted performance. In addition, the haptics rendering needs to update significantly faster compared to graphics rendering (Salisbury et al., 2004).

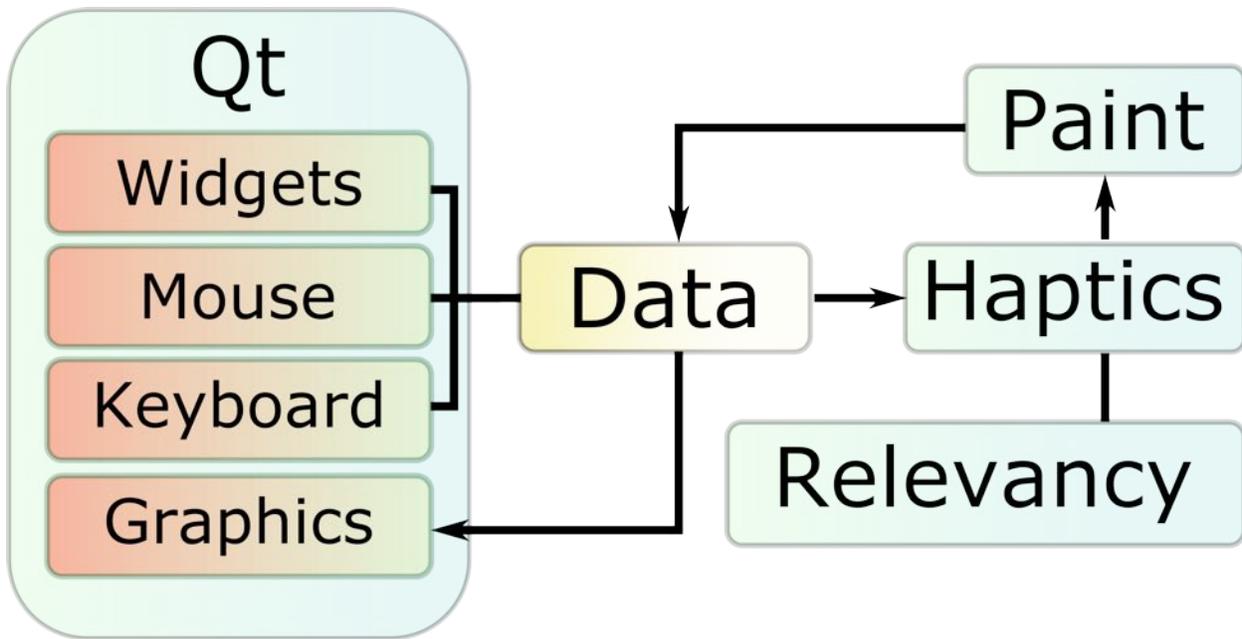


Figure 4.20: Organization of threads. The arrows show the flow of data from one thread to another. If a line does not have an arrow, that indicates information flows bidirectionally.

To resolve this issue, the system is organized to run time-sensitive tasks separately from each other. The two main threads execute the graphics loop and the haptics loop, respectively. The graphics loop displays graphics on the screen at a frame rate around 30Hz, while the the haptics loop controls the probe, obtains input from buttons on the haptic device, and provides haptic feedback at a rate of around 1000Hz. However, the rates may differ based on how intensive the processes are within their respective loops. For example, the haptics loop would run slower if the number of sample points around the probe increased, and the graphics loop would run slower if the resolution of the graphics window increased. Note that the keyboard and mouse events share the same thread as the graphics loop due to Qt’s system design (Figure 4.19). This means that intensive operations using the keyboard and mouse will also slow down the graphics loop.

In addition, processes that require marking or labeling large number of voxels are performed on a separate thread. The output from these processes is much less time-sensitive compared to graphics and haptics rendering. The processes includes painting to label a region of inter-

est, and haptic feedback to mark regions within volumetric data to interact with. Before a region is marked or labeled, input parameters, such as the size of the local region surrounding probe, are synchronized to match with the thread. Then other threads cannot modify the set input parameters while the thread is being updated. See Appendix D for additional information on system requirements and file formats.

# Chapter 5

## Case Studies

Analysis of flower heads scanned using micro-computed tomography (micro-CT) is the main motivation behind SHVR. Flower heads contain numerous florets at the top of the head that are arranged into parastichies — oblique spiral rows — to create a phyllotactic pattern. These florets connect to a vast network of vascular bundles underneath the receptacle — tissue near the end of reproductive stems — to receive nutrients from the stem. The appearance of the vascular structure can be mesmerizing due to the highly regular, yet varied spatial arrangement of vascular bundles. To understand the relationship between the vascular pattern and the phyllotactic pattern of florets, several different flower heads were annotated using SHVR.

### 5.1 Annotation Workflow

The annotation workflow consists of four steps: preprocessing, exploring, annotating, and visualizing. In the preprocessing step, the size of the image data is reduced to fit into GPU memory by cropping a region of interest, and lowering the resolution of the dataset. In

addition, noise and inhomogeneities within structures are reduced by blurring the image data using a Gaussian convolution filter. A convolution filter modifies the image data by applying a kernel within a local region around each voxel. A kernel is a predefined matrix that weighs neighbouring voxels to achieve different effects, such as blurring, sharpening, embossing, detecting edges, and more (Ormesher, 2020). This is mathematically described as:

$$g(x, y) = w * f(x, y) = \sum_{dx=-a}^a \sum_{dy=-b}^b w(dx, dy) f(x + dx, y + dy) \quad (5.1)$$

where  $g(x, y)$  is the output filtered value,  $f(x, y)$  is the original value in the image data at cell index  $(x, y)$ , and  $w$  is the filter kernel. The kernel for Gaussian convolution filter is derived from Gaussian function — a function with a symmetric bell curve shape — to provide more weight to neighbouring voxels that are closer to the cell index  $(x, y)$ . Blurring also helps smoothen the surface of objects for more aesthetically pleasing visualization.

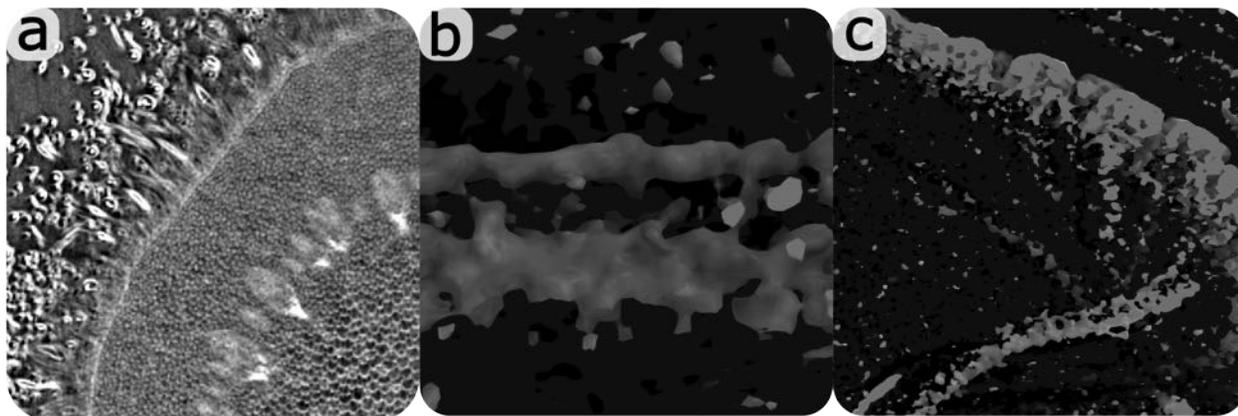


Figure 5.1: Contents of a CT-scanned flower head. (a) A flower head shows visible epidermal, vascular, and ground tissue. (b) A vascular bundle is comprised of xylem and phloem tissues that appear separated as two separate tubular structures. (c) Underdeveloped vascular bundles appear fragmented and noisy.

Exploration is the process of becoming familiar with the dataset (Kniss et al., 2002). By exploring, the user can understand what to expect from the dataset, and plan how to annotate

effectively. In particular, the vascular structure within flower heads is explored. Vascular bundles are characterized as tubular structures that have higher intensity relative to the surrounding tissues, such as dermal tissue — the outer protective layer of a plant—, and ground tissue — tissue that is not dermal, or vascular tissue (Figure 5.1a). Vascular bundles within a flower head can be found just underneath the receptacle, and are likely to follow along the epidermis — the layer of cells that form a boundary between the plant and the external environment. Within developing flower heads, vascular bundles may be difficult to perceive due to low contrast to the surrounding tissues, which may appear as fragmented trail of dense tissue (Figure 5.1c). By applying the knowledge about the location and shape of vascular bundles from mature flower heads, the underdeveloped vascular bundles from younger flower heads can be identified.

With interactive annotation, users have much flexibility in what structures to annotate, and how to annotate them to help extract a variety of information from a sample. However, users may also need to spend additional time to amend their existing annotation if they accidentally annotate structures that are not of interest. By planning interactive annotation, an effective strategy to structures of interest to annotate quickly and effectively can be identified, and avoid potential problems while annotating. For example, a vascular bundle within a developing *Gerbera hybrida* may appear as two separate tubular structure consisting of xylem and phloem tissues (Figure 5.1b). With segmentation, both tubular structures can be painted. However, with tracing, the centerline in between two tubular structures can be difficult to perceive. Because vascular bundles connect with other bundles on the inner side, the inner tubular structure is traced.

The process of annotating differs between painting and tracing. For painting, the general technique is to paint large structures first, and then work down to regions that are prone to painting outside of the regions of interest, such as the boundary between structures. If a structure is difficult to perceive due to low intensity values, then density painting (Section

4.2.1) is used to increase its visibility. For tracing, nodes are added within the centerline of a tubular structure. A node at the branching point of a tubular structure can be selected to start creating a new branch. Nodes can be selected, moved, and deleted, and be linked with other nodes. In both cases, the probe is adjusted to loosely match the diameter of the veins. This provides an optimal haptic feedback using object and proximity following haptic modes, and acts as a visual guide for painting and placing nodes during tracing. In addition, annotation is saved frequently to avoid losing progress, and to return to an earlier state if a large mistake was made.

Finally, the annotated structures of interest are visualized to be aesthetically pleasing, and be conveyed clearly. This may involve adjusting the view, clipping planes, cutting planes (Section 4.2.2), transfer function, distance colour blending (Section 4.2.2), colour and opacity of annotation, and light settings.

The time taken for annotation depends heavily on the size, complexity, and quality of a given flower head. If the flower head is small, does not contain much noise, and has well defined structures, then the annotation can be completed within a short period of time. Otherwise, the user will need to take the time to carefully annotate the boundary between structures, verify the accuracy of annotation, and clean up mistakes. A fully developed flower head takes around 15 minutes to explore the contents of the flower head, 10 minutes to 1 hour to annotate the vascular structure within a flower head, and 30 minutes to visualize the results. Although the process of segmenting and tracing differ, the time taken is often similar. Tracing takes time to place nodes and internodes in the centerline of each tubular structure, while segmentation takes time to clean up the boundary between different structures. Thus, a typical session takes about 1 to 2 hours.



Figure 5.2: Different views of *Gerbera hybrida*. (a) A photograph of a mature flower head. (b) Top view shows the florets arranged with visible parastichies. (c) Vascular structure is divided into three sections: green represents the bract, abaxial and stem veins, red represents the adaxial veins and orange represents the florets and floret veins.

## 5.2 Example of an Analysis

Among different species of flower heads, florets in the Asteraceae family produce a phyllotactic pattern with much higher parastichy numbers than flowering shoots such as *Arabidopsis* (Zhang et al., 2021). In particular, *Gerbera hybrida* was selected due to its susceptibility to genetic transformation — introduction of foreign DNA to plant cells, tissues or organs (Keshavareddy et al., 2018). By modifying the genes, various development factors can be tested to find a relationship between vascular and phyllotactic patterns (Figure 5.2). *Gerbera*

*hybrida* was analyzed by comparing the annotating the vascular structure with the vascular structure at different stages of development, and the vascular structure of different species of plants.

Three new terminologies have been created to describe the different sections of the vascular structure within a *Gerbera hybrida* flower head<sup>12</sup>: First, abaxial veins continue from the stem veins to go into the bract veins. Second, floret veins extend down from the florets through the receptacle. Finally, adaxial veins form an intermediary connection between the floret veins and the abaxial veins. For simplicity, annotation is divided into three section. The first section is made up of bract, abaxial, and stem veins. The second section is made up of adaxial veins. Finally, the last section is made up of florets and floret veins (Figure 5.2c). Although florets are not a part of the vascular structure, they are included to show the connection between the vascular pattern and the phyllotactic pattern of florets.

By arranging various annotations of *Gerbera hybrida* flower heads by their stage of development, the developmental sequence of plant organs can be understood (Figure 5.3). A flower head starts as inflorescence meristems — undifferentiated plant tissue giving rise to flower heads —, where primordia — plant organs early in the development stage — start developing from the outer rims toward the inner head. These primordia mature into florets, and organize into a phyllotactic pattern consisting of parastichies.

In the early stages, only the bract and stem veins are visible in the vascular structure. Within the head, abaxial, adaxial, and floret veins all grow separately from each other. Abaxial veins grow outward with the shape of the flower head. Adaxial veins start to form below the receptacle separated from the abaxial veins. These veins grow in two directions: towards the abaxial veins at the outer rim, and towards the inner head. However, they

---

<sup>1</sup>The researched literatures do not have terminologies to describe different sections of the vascular structure within a flower head due to being previously undiscovered.

<sup>2</sup>These new terminologies are useful to describe the vascular structure within a *Gerbera hybrida* flower head. However, they may not properly describe other flower heads.

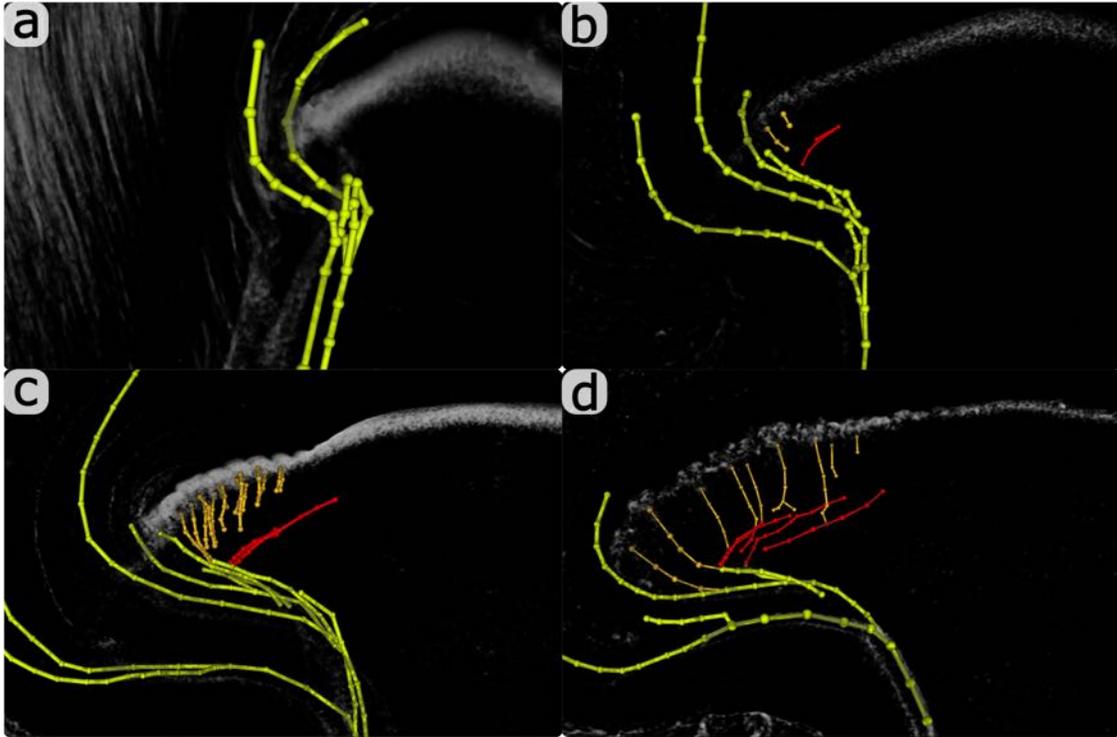


Figure 5.3: Tracings of *Gerbera hybrida*. (a) Early stage shows only the abaxial and bract veins. (b) Floret veins start emerging from florets at the outer rims of a flower head. Adaxial veins emerge separately from all other veins. (c) Floret veins connect with nearby abaxial veins. More florets emerge toward the center of a flower head. Abaxial veins extend towards both the outer rims and the abaxial veins. (d) Floret veins closer to the center connect with nearby adaxial veins.

stop before reaching the center of the flower head. Floret veins start developing from the most mature florets at the outer rim towards the inner head. These veins grow down from the receptacle, and connect with nearby abaxial veins at the outer rims, floret veins at the center, and adaxial veins in between the outer rim and the center.

The relationship between vascular pattern and phyllotactic pattern of florets differs between species of flower head (Figure 5.4). For example, *Gerbera hybrida* has straight adaxial veins running radially from the center of the head, but does not develop near the center of the head. These veins are spaced apart in roughly equal sectors. *Helianthus annuus* has drooping adaxial veins that develop up to the center of the head. These veins appear to run radially, but not as much as *Gerbera hybrida*. *Bellis perennis* has adaxial veins that run alongside

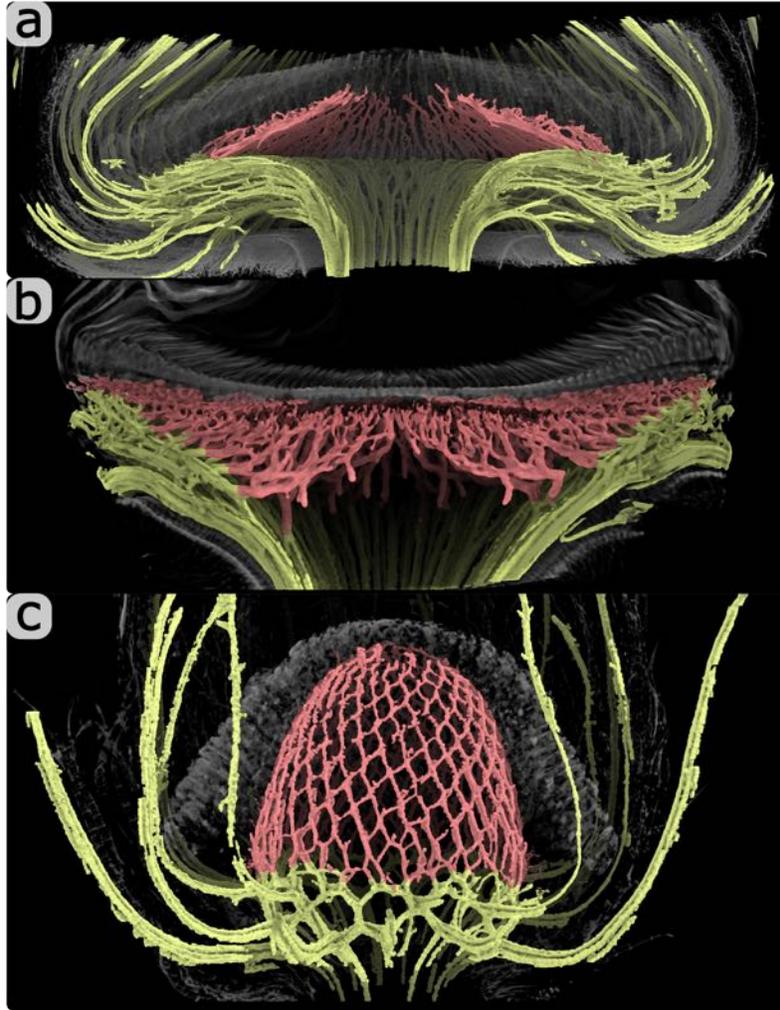


Figure 5.4: Comparison of adaxial veins (red) between different flower heads. (a) *Gerbera hybrida* has straight adaxial veins running radially from the center. (b) *Helianthus annuus* has drooping adaxial veins that connects with other nearby adaxial veins. (c) *Bellis perennis* has adaxial veins arranged in a regular reticulate pattern.

its parastichies. These form a regular reticulate pattern. All of the heads in the examples have identifiable parastichies in the arrangement of florets, but have vastly different vascular structures. This suggests that vascular patterns may range from sectorial to those aligned with parastichies. Thus, phyllotactic pattern of florets grow separately from the vascular pattern, but the vascular structure likely depends on the phyllotactic pattern for its shape.

However, the relationship between vascular pattern and phyllotactic pattern of florets requires further study to understand the complex mechanism behind their development. Al-

though differences in the vascular structure can be between the different developmental stages and species, several other factors may need to be tested. For example, The shape and size of the flower heads differ vastly between the compared species. *Gerbera hybrida* appears as a flat sphere, *Helianthus annuus* as a depressed sphere, and *Bellis perennis* as an elongated sphere. This may impact where the adaxial veins start forming, or how the floret veins connect to nearby veins. In addition, the phyllotactic pattern of florets may influence the underlying vascular structure based on factors such as the size, shape, and arrangement of florets.

### 5.3 Additional Examples

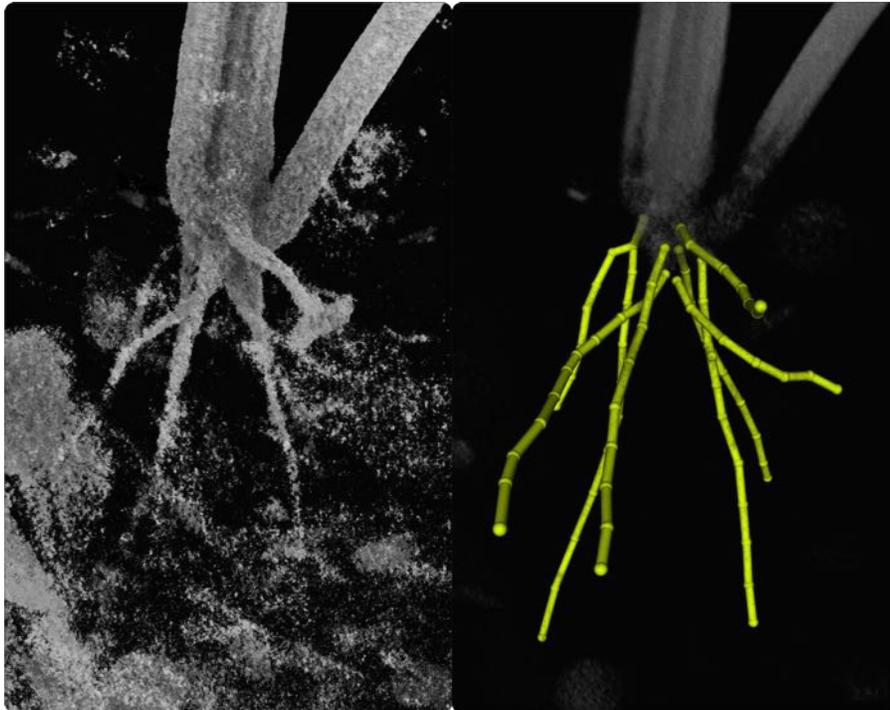


Figure 5.5: Tracing of roots in a heterogenous mixture of soil, rocks and other organic matter.

SHVR is also capable of annotating the vascular structure in other parts of a plant, such as roots embedded in soil (Figure 5.5). Roots appear as thick and long tubular structures that become less visible as other objects, such as rocks and other organic materials block

their view. In addition, the roots may be touching other objects in the soil. To trace the roots, proximity following haptic mode (Section 3.1) was used to follow along the trail of fragmented signals connected to the known roots above the soil. Objects within the soil were avoided by visually confirming that the traced structure was tubular, and connected to the roots above.



Figure 5.6: Segmentation of the vascular structure in *Arabidopsis thaliana*.

The vascular structure within *Arabidopsis thaliana* was also annotated (Figure 5.6). Unlike flower heads where vascular bundles are connected in a vast network, stem veins are long, tubular structures that do not form complex connections. However, these veins were thin, and had little distance from the neighbouring dermal tissues. To segment the stem veins, object following haptic mode (Section 3.1) was used to follow the centerline of the stem veins. The painting threshold was set higher than the transfer function threshold to prevent bleeding into the neighbouring dermal tissues.

## 5.4 Annotation Experience

About 47 different plant samples were annotated using SHVR, in which 11 of them were mature flower heads. As mentioned in annotation workflow (Section 5.1), the time spent and the ease of annotating depended on the size, complexity, and quality of the scanned samples: as the size and complexity increased and the quality decreased, the time spent increased, and the ease of annotating decreased. This was because the complexity of the vascular structure generally increased as the size of a sample increased. The difference in complexity can be seen between early (Figure 5.3a) and late (Figure 5.3d) developmental stages of *Gerbera hybrida*. The vascular structure in the late stage has more connections between vascular bundles, more regions to annotate, and more anatomical structures to label. Several factors can impact quality, including the imaging technique used to capture thin structures deep within a sample, the preparation of the sample, and the size of the structures of interest (Kuan et al., 2020). The difference in quality can also be seen in the developmental stages, where the vascular bundles in the late stage is much easier to perceive due to being thicker than the earlier stages.

Haptic assistance has been indispensable during the annotation process. By keeping the probe either on the surface of an object, or within the centerline of a target tubular structure, the cognitive load to annotate was greatly reduced. In addition, haptic feedback gave the freedom to either accept or deviate from the given assistance based on knowledge about the target structure, and the visual feedback. However, haptic feedback has sometimes been deceiving. For example, following a structure using only haptic assistance could lead the probe into a different structure unknowingly. Another issue is that the haptic device may not provide sufficient granularity in feedback to find the centerline of a tubular structure due to the limited resolution of the positional sensors. To avoid these problems, the annotation should be visually checked from time to time for any deviation from the desired output.

# Chapter 6

## User Study

To measure the usefulness of SHVR, an evaluation was performed. (Olabarriaga and Smeulders, 2001) provides three evaluation criteria to assess the capabilities of interactive annotation method: accuracy, repeatability, and efficiency. Accuracy is the degree to which the user annotation corresponds to the ground truth annotation. A ground truth annotation or a phantom — artificial data with similar size, location and overall shape of target structure — may be used as a gold standard — reference annotation produced and verified by several experts (Ritter et al., 2011) (Preim and Botha, 2014). An advantage of interactive annotation method is that users can always improve the accuracy of annotation until they are satisfied (Olabarriaga and Smeulders, 2001). This is particularly important for flower heads, which can vary greatly in shape and size based on its stage of growth, and species of plant. In addition, they contain a large number of anatomical structures, such as florets, bracts, vascular bundles and stems, which may be of interest to annotate. Repeatability is the similarity between several annotation results when each result is created with the same intention. Annotation results may be created by a single person to observe intra-operator variability, or several people to observe inter-operator variability. Because the speed of interactive method depends heavily on the task, efficiency is divided into a computational and

interactive part. For computational part, efficiency is the length of time needed to generate the result. For interactive part, evaluation is measured by the amount of interaction, such as the number of mouse clicks, and complexity of task, such as carefully drawing around a boundary, or knowing what sequence of operations can be used to complete a task. In other words, interactive annotation methods should “minimize user interaction, minimize annotation variability among users, and be computationally fast to allow quick user editing” (Zhao and Xie, 2013).

A user study obtains qualitative and quantitative feedback from several participants to see if the system satisfies its design goals. With SHVR, the goal is to quickly and robustly produce accurate annotation of vascular structure within scans of flower head. Thus, the study was designed around measuring the accuracy of annotation, ease of annotating, and the usefulness of haptic feedback.

## 6.1 Study Setup

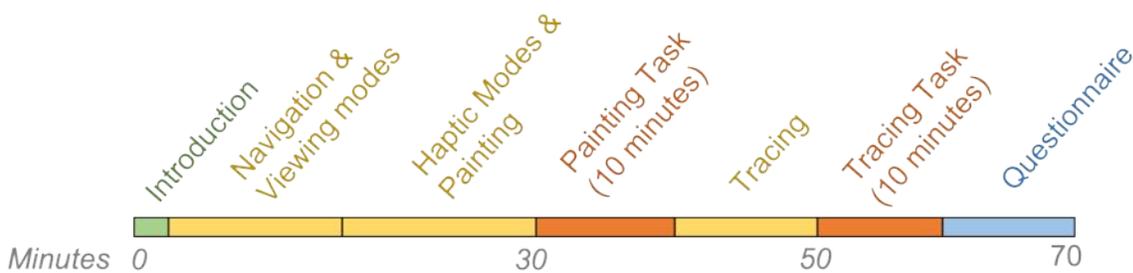


Figure 6.1: User study timeline. Participants were given instructions on how to use SHVR (yellow) before being asked to annotate a small section of a flower head (orange). The study took roughly 70 minutes per participant.

Each participant was instructed on how to use SHVR before being tasked with annotation. They were instructed on how to navigate using the haptic device, use various haptic modes,

and annotate a structure of interest using painting and tracing. After the instructions on painting, participants were free to use SHVR until they felt comfortable with the controls. This included viewing various parts of a sample, painting a structure of interest, and following a tubular structure using haptic assistance. Then, they were tasked with painting a small section of a scanned flower head for 10 minutes. For tracing, they were given time to practice using SHVR, before being tasked to trace the same section of a flower head. At the end of each task, their annotation result, as well as the time spent on annotating was recorded. At the end of the user study, the participants were given a questionnaire to describe their experience regarding navigation, painting, and tracing. Each question was rated between 1 and 5, where 5 indicated that they strongly agreed with the statement. This study took around 70 minutes per participant (Figure 6.1). See Appendix B for the complete questionnaire.

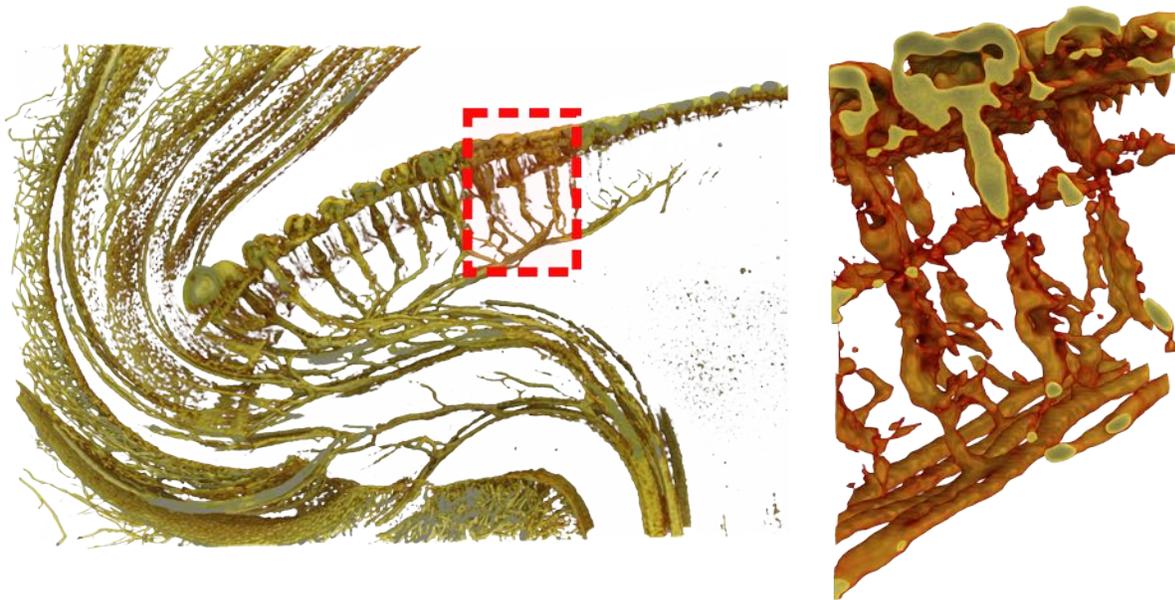


Figure 6.2: Section of *Gerbera hybrida* flower head. (left) Approximate location of where the section was cropped from. (right) Rendered section of *Gerbera hybrida* flower head used during annotation tasks.

The selected dataset for the annotation task had to be simple enough to complete within the allocated time, while being complex enough to evaluate the usefulness of the system in a more complex environment. In addition, the structure of interest had to be unambiguous for the

users to explore and annotate without much confusion. This resulted in selecting a section in between the center and the outer rim of a *Gerbera hybrida* flower head, where adaxial veins connect with floret veins (Figure 6.2). Even though this dataset is much simpler than a typical annotation task, the task of touching the object and following along the centerline of tubular structure remain the same. The main difference is that the user explores less structures, and proof-edits less annotation results.

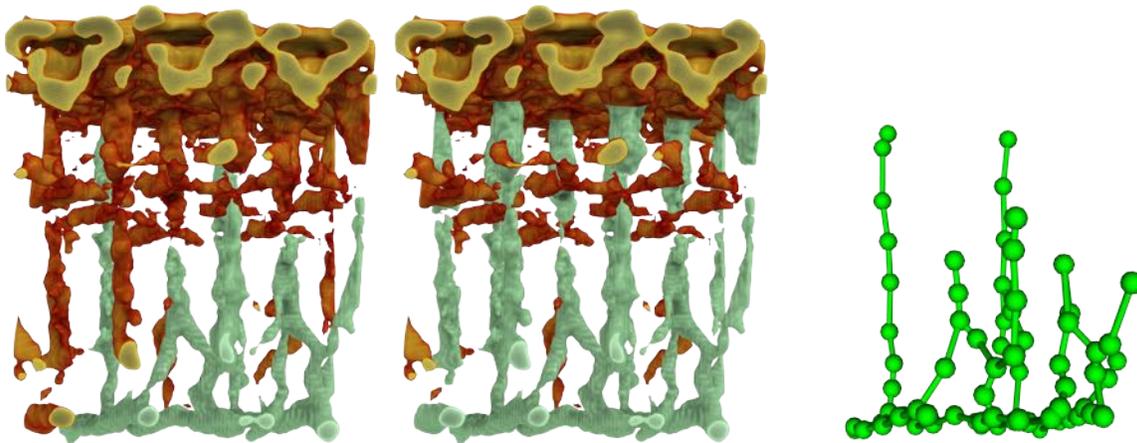


Figure 6.3: Ground truth annotations. (left) Partial segmentation. (middle) Full segmentation. (right) Partial tracing.

Ground truth annotation of the vascular bundles was produced by segmenting the vascular bundles while avoiding florets and receptacle tissue nearby based on the shape, size, and location of tubular structures. This was possible because the anatomical structures within the chosen section of *Gerbera hybrida* flower head is easily identifiable. In early trials of the user study, participants had difficulty identifying all of the vascular bundles within the sample. As a result, the instructions were simplified by asking the participants to annotate vascular bundles, starting from the ones that are connected to adaxial veins. This resulted in annotation close to the simplified ground truth segmentation. Since all participants understood the vascular structure within the simplified ground truth segmentation, the ground truth tracing was produced based on it (Figure 6.3).

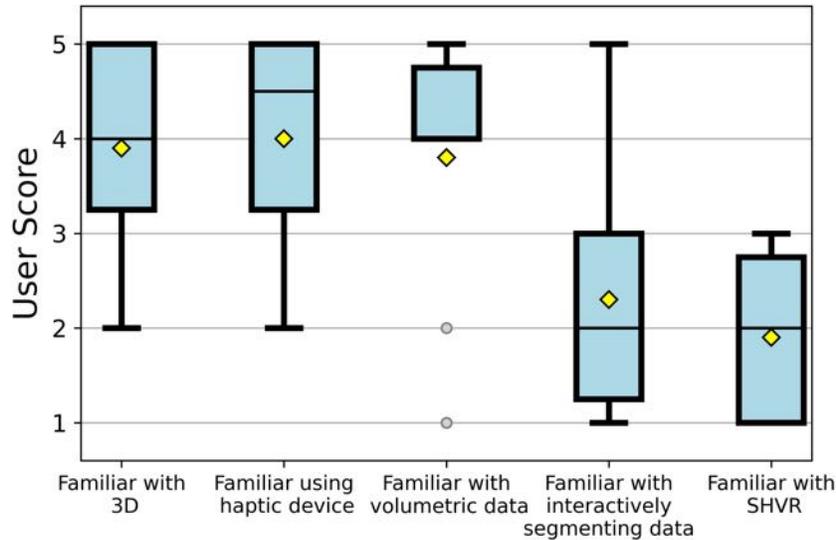


Figure 6.4: Information about participants. Most participants were familiar with haptics and 3D graphics, but not with image annotation. Box plots show how the responses are distributed, divided into quartiles. Outliers (gray circle) indicates that a data point differs significantly from the rest of the dataset. Mean value (yellow diamond) is the average value of all the data points for a response.

This study had ten participants recruited from within the author’s lab and from neighbouring labs<sup>1</sup>. Eight participants were familiar with haptics and 3D graphics, but did not have much experience with image segmentation. One participant was not familiar with haptics or 3D graphics, but was experienced with volumetric data and interactive segmentation from performing segmentation on 2D image slices using a mouse and keyboard. One participant did not have any knowledge of volumetric data or interactive segmentation (Figure 6.4).

## 6.2 Study Results

Navigation using the haptic device was received positively by the participants (AVG: 4.0). Translating the volume was unanimously agreed to be intuitive. However, rotating the volume was disorienting to some participants. This was likely due to losing visual landmarks,

<sup>1</sup>Three participants were removed due to their close involvement with this project.

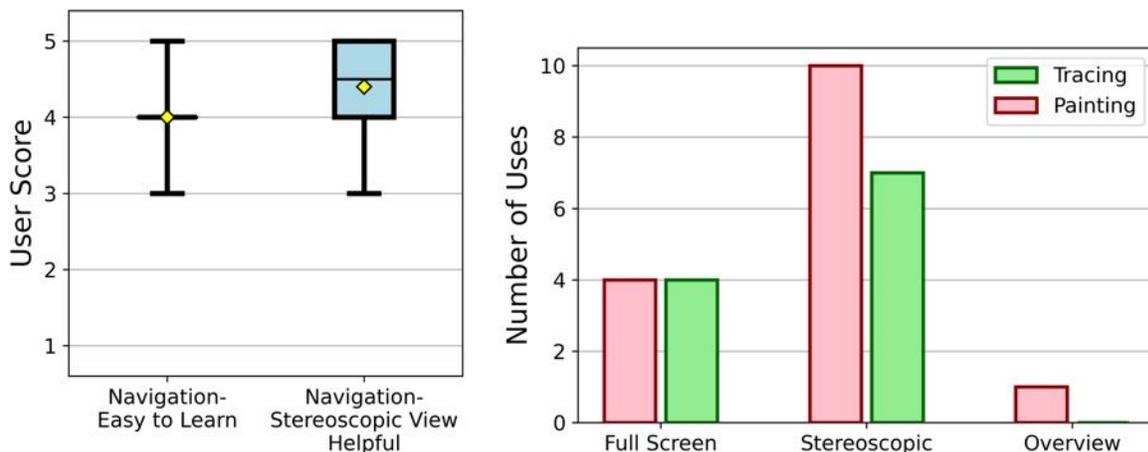


Figure 6.5: (left) Participant feedback on navigation. Navigation was generally considered to be intuitive. (right) Distribution of used viewing modes while annotating. Red represents painting, while green represents tracing. Users preferred to use stereoscopic viewing mode over full screen and overview viewing modes.

such as a recently annotated region, as the view changed. Out of the three viewing modes, stereoscopic view was the most used (Figure 6.5). Participants found it to be more useful for navigating compared to using a 2D display (AVG: 4.4). Only one participant used the overview mode during a task. Because the entirety of the sample was visible in the stereoscopic and full screen mode, participants commented that they had no need to view the sample in overview mode.

The accuracy of annotation was measured by comparing user annotation with ground truth annotation. For segmentation, the accuracy is calculated using a Dice coefficient  $D$ , which measures the amount of intersection between a ground truth segmentation  $A$  and a user segmentation  $B$  (Dice, 1945) such that:

$$D = \frac{2|A \cap B|}{|A| + |B|} \quad (6.1)$$

where  $|A|$  and  $|B|$  refers to the number of voxels for ground truth segmentation and user segmentation respectively. This means that higher score implies a greater overlap between

ID	Partial Seg.	Full Seg.	N. Score	N. Out	I. Score	I. Out	Total	T. Error
G. Truth	1.000	0.824	0.971	0/70	1.000	0/71	26/26	0
2	0.883	0.770	0.969	0/32	0.914	2/29	11/26	0
3	0.862	0.819	0.916	3/63	0.967	1/63	26/26	0
5	0.907	0.841	0.938	3/57	1.000	0/54	17/26	0
6	0.966	0.841	0.974	1/73	1.000	0/72	23/26	0
7	0.988	0.814	0.977	0/46	0.995	0/46	14/26	0
8	0.991	0.817	0.959	1/43	0.987	0/38	11/26	0
9	0.970	0.803	0.918	2/46	0.968	2/48	26/26	1
10	0.881	0.847	0.977	0/53	0.941	3/47	19/26	4
11	0.975	0.806	0.964	0/35	0.966	1/35	14/26	1
12	0.986	0.812	0.871	7/67	0.909	6/63	24/26	1
Average	0.941	0.817	0.946	0.029	0.965	0.031	18.5/26	0.7

Table 6.1: Accuracy of annotation results. From left to right, the columns stands for: participant ID, Dice score compared with partial ground truth segmentation, Dice score compared with full ground truth segmentation, tracing score for nodes, number of nodes placed outside of the region of interest out of the total number of nodes, tracing score for internodes, number of internodes placed outside of the region of interest out of the total number of internodes, number of segments traced, and number of topological error. The outputs from the ground truth annotations are included to compare with the participant scores.

the ground truth segmentation and user segmentation. The results show that user segmentation closely matches with the partial ground truth segmentation, where six participants had a score greater than 95% (Table 6.1). Participants lost score due to rough delineation at the boundary between two structures, misunderstanding about what region is part of the vascular structure, and being unable to see their mistakes in view (Figure 6.6). Some participants commented that checking the back side of vascular bundles was challenging as it required multiple steps to move using the haptic device. One participant wanted to see the exact region that would be painted instead of approximating the size of the painted region using the probe.

For tracing, the accuracy is measured using a combination of various metrics. Some potential metrics include distance from the centerline, number of branching points, total length of the trace, and similarity in topology (Acciai et al., 2016). For SHVR, a combination of the distance from the centerline, the amount of the sample traced, and the number of topological

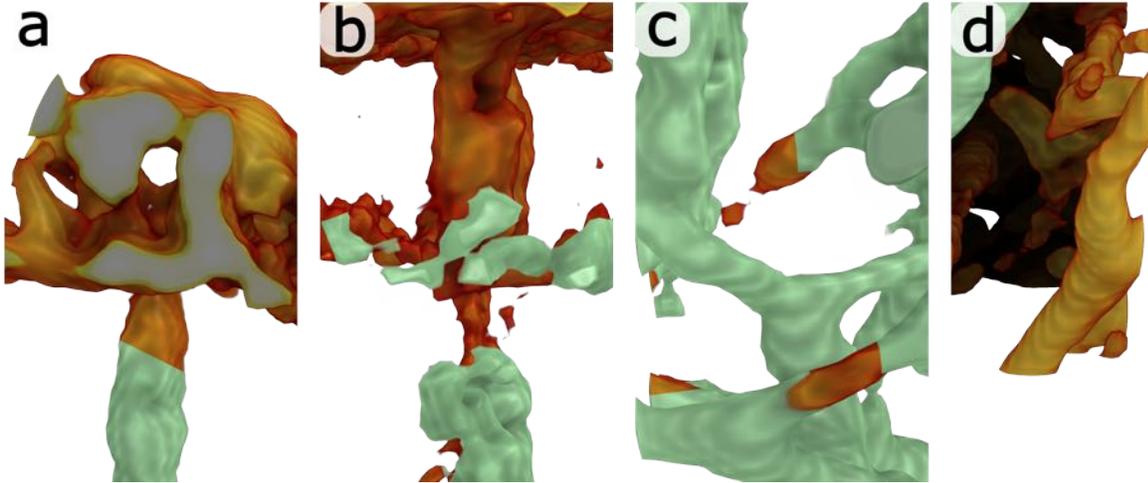


Figure 6.6: Mistakes during painting. (a) Painting stops early before the end of the vascular bundle. (b) Receptacle tissue is painted, which is not a part of the vascular structure. (c) Parts of a vascular bundle are not painted. (d) Adaxial vein is not painted.

error are used to evaluate accuracy.

(Acciai et al., 2016) proposes several measures to calculate the distance from the centerline by comparing the distance between a point in the user tracing and the nearest point in the ground truth tracing. However, the centerline within the vascular structure of a flower head can be subjective due to how the connection between veins can appear in the volumetric data. For example, separate structures may appear connected to each other due to their proximity, or appear separated if the structure is thin. In addition, other nearby structures, such as dermal, receptacle, and ground tissue, may appear connected to the structure of interest. To accommodate potential variations in connectivity, the amount of region within a node overlapping with the vascular structure is used to measure the distance from the centerline. This works by placing nodes inside of tubular structures that roughly have the same diameter as the veins. If the node is at the centerline of a tubular structure, then the node will return a high score. If the node is outside of a tubular structure, then the score for the node will be zero. Since the diameter of veins change, nodes at the centerline may not return a full score if the tubular structure is smaller than the node, and nodes outside of the centerline may not return a partial score if the tubular structure is larger than the

node. The region with vascular structure is known from the full ground truth segmentation. The final tracing score for nodes is the average of the score from individual nodes. This is calculated as:

$$D_{trace} = \sum_{i=1}^n \frac{|A \cap B_i|}{|B_i|} \quad (6.2)$$

where  $A$  is the ground truth segmentation,  $n$  is the total number of nodes, and  $B_i$  is the region occupied by the  $i^{th}$  node. Nodes that intersect less than 50% with the vascular structure are counted separately. The above calculation is also done for internodes, where an internode is approximated as a node located at its midpoint.

The amount of tracing task completed within the allocated time is measured using the total number of segments within the user tracing. Each segment is a link between two branching nodes, two terminal nodes, or a branching and terminal nodes. Partial segments are excluded from the count. When the arrangement of segments in the user tracing differs from the ground truth tracing, then the count of topological error is incremented. The number of segments and topological errors were counted manually.

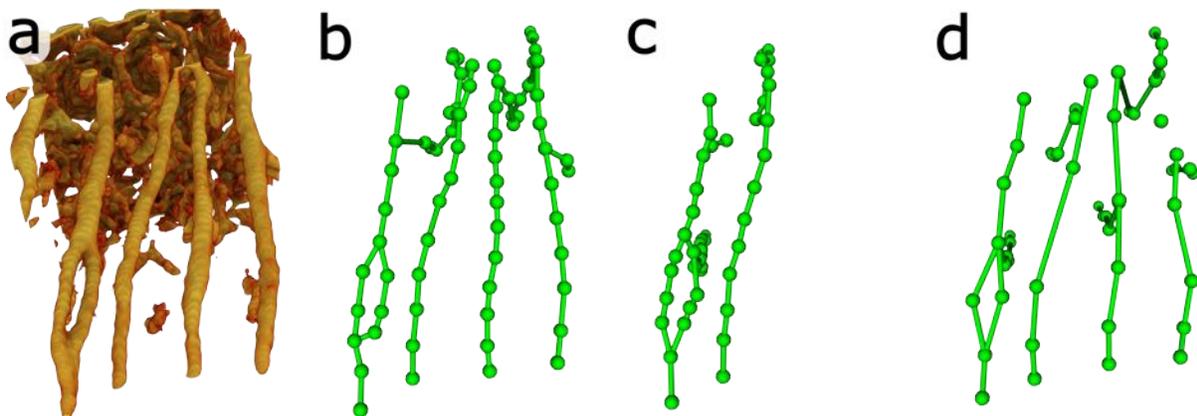


Figure 6.7: Comparison of tracing results between participants. (a) Section of flower head viewed from the bottom. (b) Result from Participant 6. (c) Result from Participant 8. (d) Result from Participant 10. The output tracing closely resemble the vascular structure within the sample.

The tracing scores for nodes and internodes were very high, where the average scores were 95% and 97% respectively (Table 6.1) (Figure 6.7). These scores are close to the tracing scores from the ground truth tracing, which indicates that the output is both accurate and repeatable with many users. Nodes and internodes that are clearly outside of the region of interest is only 3% of the total nodes and internodes placed during tracing. Some mistakes include connecting two separate vascular bundles that were in close proximity to each other, connecting two distant nodes without noticing their mistake, and connecting nodes within a vascular bundle.

Participants were much faster with painting compared to tracing (Table 6.1). With painting, three participants used the full 10 minutes for tracing, and two participants finished the task within less than 5 minutes. However, with tracing, 8 participants used the full 10 minutes, where the minimum time spent was 7 minutes and 30 seconds. In addition, two participants were less than 50% complete with their tracing when the time limit was reached. An interesting observation is that tracing scored more than painting despite it being more complicated. This may be due to participants becoming familiar with the task, dataset, and application as the user study progressed.

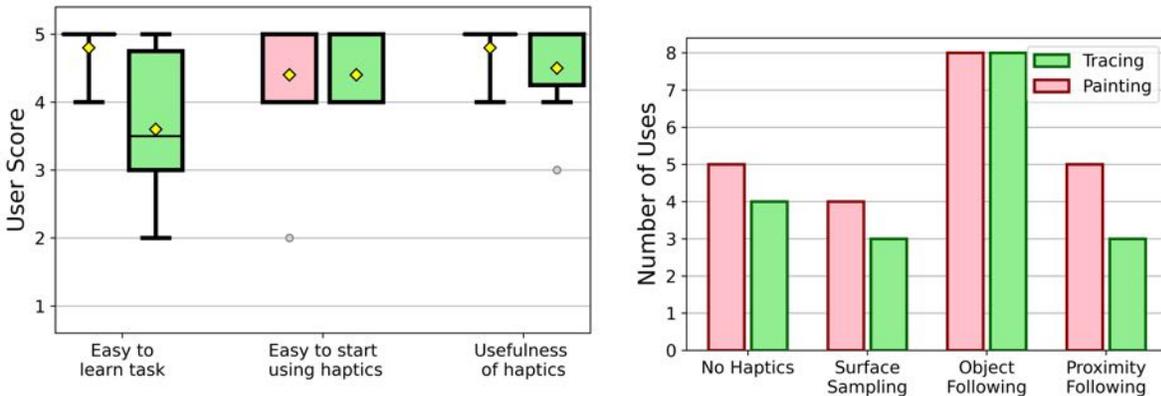


Figure 6.8: Responses for haptic feedback. (left) Users found that painting was much easier to learn compared to tracing. Haptic feedback was useful for both annotation tasks. (right) Users preferred “Object Following” haptic mode over all other haptic modes.

Among the haptic modes, participants preferred using object following mode, followed by

no haptics, proximity following, and then surface sampling (Figure 6.8). Haptics was often paused to explore and navigate around the volume. Once a satisfactory view was found, participants re-enabled haptic assistance to annotate. All participants found painting to be easy to learn (AVG: 4.8). Most participants found the haptic assistance for painting to be easy to use (AVG: 4.4), and useful (AVG: 4.8). On the other hand, participants found tracing to be more challenging to learn (AVG:3.6). This decrease in score is from the increase in complexity of the task. They had to use more keys to create and edit tracing, which is more mentally taxing. They still found haptic assistance to be easy to use (AVG:4.4), and useful (AVG:4.5). Two participants rated the usefulness to be only somewhat useful. This is surprising as other participants rated the usefulness of haptic assistance for tracing equally or higher than painting. Participants commented that haptic assistance provided suggestions on where to place new nodes, and kept the probe steady from the involuntary movement of the hand while operating the haptic device. One participant commented that they would be lost where to place the next node without haptic feedback. Although the last comment may have been intended as a compliment, it also indicates the desire for more guidance on where to place new nodes.

Overall, the comments about SHVR were positive. Many participants desired to adjust some parts of the setting, such as the painting radius, visibility of the volume, and distance from the clipping planes. However, these settings were not provided to the participants due to time constraints.

### **6.3 Expert Evaluation**

An additional user study was conducted by visiting experts. One of them was a biologist who has some experience in annotating scanned plant samples. He commented that this system would be helpful for annotating 3D structures of interest within scanned samples. In

particular, he liked the flexibility of navigation using the haptic device, and how SHVR can be used for a long period of time<sup>2</sup>.

Another was an expert in segmenting biological scans. He evaluated SHVR by annotating roots embedded in soil. This scan had much noise and artifacts from scanning which made automatic segmentation not feasible using Avizo, a commercial visualization and analysis software for volumetric data (Thermo Fisher Scientific). After using SHVR for a week, he made several observations. He commented that the haptic device gives more dexterous control over the view and interaction compared to using a mouse and keyboard. The view can be easily adjusted to view from arbitrary angle, and haptic feedback allows following along a tubular structure. Compared to Avizo, which heavily relies on pre-processing for segmentation, such as filtering, smoothing, and removing voxels outside of the region of interest, the use of haptic device and haptic feedback made annotating more intuitive and convenient. In particular, he commented that the learning curve of SHVR is much shorter and easier compared to Avizo. He felt that he understood how to annotate within less than a day using SHVR, but spent weeks to understand how the segmentation editor works before he could confidently segment on Avizo. To measure the difference in performance, he segmented the vascular structure within a full *Gerbera hybrida* flower head. It took him 8 to 10 hours to figure out a segmentation scheme, and a further 14 to 16 hours to complete the project. Much of the time was spent manually cleaning up segmented regions outside of the region of interest. In comparison, the same flower head was segmented using SHVR in less than 2 hours.

---

<sup>2</sup>This was in comparison to VR systems, similar to (Usher et al., 2018)

# Chapter 7

## Discussion

SHVR combines haptic assistance, interactive annotation, and volumetric rendering to learn and incorporate knowledge about the target structures during annotation and visualize the final output. The annotation workflow involves preprocessing the image data, exploring various structures within a sample, annotating structures of interest, and visualizing the result. In my experience, this process takes roughly 30 minutes to 2 hours per head depending on the size, complexity, and quality of dataset. An analysis of the vascular pattern of *Gerbera hybrida* compared the vascular structure of several stages of development and different species of flower heads. Results showed that vascular patterns likely depend on the phyllotactic pattern of florets, while phyllotactic patterns are independent from the vascular pattern.

The usefulness of SHVR was evaluated through a user study, which analyzed accuracy, repeatability, and efficiency of annotation in relation to its speed. Regarding accuracy and repeatability, participants were able to reliably produce annotations close to the ground truth annotations. Two key features that users found helpful for this task were haptic assistance and stereoscopic viewing. Participants agreed that haptic assistance was easy to use and useful, and they preferred to keep it enabled during annotation. They also found stereoscopic

viewing to be useful. However, even though all participants used stereoscopic viewing during the painting task, two participants did not use it again for the tracing task. In contrast, all participants used haptic assistance for both painting and tracing. This suggested that haptic assistance was perceived as more useful for annotating than stereoscopic viewing. Still, using the two features together provides the best outcome by delivering spatial information through both the haptic sensory channel and visual sensory channel. This result is consistent with (Ernst, 2006), which showed that combining haptic and visual sensory channels improve accuracy of perception and the reliability of information perceived.

In terms of interactivity, participants found navigation and painting easy to perform. However, they did not feel the same for tracing. Even though the participants were already exposed to the sample before the tracing task, most of them could not complete it within the time allocated. In contrast, most participants finished painting early. This difference may have been due to increased complexity of the task, where participants were expected to carefully place nodes at the centerline of tubular structures, compared to broadly painting a region of interest. Another issue is that annotations become more challenging to correct as the complexity of the sample increases. Although painting over a region or amending the connection between nodes is simple, it can become tedious if many annotations need to be modified. This may occur when regions outside the structure of interest are erroneously painted, or when the colour group of nodes in a tracing needs to be converted into another group.

Various methods have been implemented to improve computational efficiency. By reducing the resolution of the screen, capping the frame rate, and precomputing ambient occlusion, a user can interact with the volume in real time. However, rendering can become computationally expensive as the size of the volumetric data increases. In particular, the GPU has to store the volumetric data, segmentation mask data, and ambient occlusion data in its limited memory. Since volumetric data above the GPU memory limit are not uncommon, datasets

have to be cropped and scaled down to a smaller size. This results in loss of information that could be useful for analysis.

Some features were removed during the development. One of them was mask layering, which assigned multiple colour groups to one voxel. This allowed a part of a sample to belong to multiple colour groups. However, since only one colour group was visible at a time and switching between mask groups was computationally expensive, the additional memory used to keep this feature was not justified. Another feature was voice commands, which allowed the user to speak commands into a microphone. It was implemented to avoid using the mouse to adjust settings in the menus. However, using voice commands was awkward since learning the commands was more difficult compared to adjusting the desired setting in a setting widget. In addition, frequent commands were much faster to execute using keyboard shortcuts. Although stereoscopic view is still implemented, stereoscopic displays have become obsolete in the consumer market. A viable alternative is to use VR headset (Wang et al., 2019) (Usher et al., 2018). However, this requires a higher refresh rate of at least 90Hz compared to the current default refresh rate of 30Hz for SHVR.

For future work, several improvement can be made to SHVR. Haptic assistance provides a framework for interacting with volumetric data. Currently, SHVR considers only intensity values and connectivity to provide haptic feedback. But by incorporating other image features, such as the size, shape, texture, boundary of objects, and distance from nearby landmarks, haptic feedback may help interact with structures of interest with more accuracy, precision, robustness, and efficiency. In addition, haptic feedback could be limited to an annotated regions to quickly edit existing annotations. Furthermore, the smoothness of haptic feedback can be improved by increasing the number of sample points when updating the location of the proxy. This may be done by simultaneously checking sample points within relevance volume using multiple threads.

Tracing may become more manageable by incorporating additional visual feedback. An

overlay could be placed on top of the probe to indicate the amount of the probe intersecting with a tubular structure, or the depth of the probe inside of a structure. Tracing may also become faster by selecting all nodes and internodes in between two branching nodes, or between a branching node and a terminal node. This can be used to move, delete or change the colour group of multiple nodes and internodes at once. Another method to speed up editing may be to arrange colour groups in a hierarchical tree such that a child group can inherit the properties of the parent, such as the colour and opacity (Yong Wan et al., 2009).

Annotations of samples can be understood with more depth by including external image analysis programs in the annotation workflow. Applications, such as MorphoGraphX (Barbier de Reuille et al., 2015) and Fiji (Schindelin et al., 2012), have a wide variety of tools for analyzing annotations. Combining the analysis from SHVR with external applications can help increase confidence in the interpreting the contents of a sample.

Larger volumetric data can be loaded by using a multi-resolution hierarchy to restrict computation to only the visible volumetric data (Hadwiger et al., 2012) (Bria et al., 2016). This method has been shown to render and annotate petabytes of volumetric data using standard GPUs.

Combining interactive annotation methods with learning-based approaches has been suggested to minimize user interaction, while providing computationally fast output (Zhao and Xie, 2013) (Acciai et al., 2016). However, the viability of these methods will need to be tested for annotating the vascular structure of plants.

Although the combination of case studies and a user study has shown that SHVR is useful, its features should be further evaluated. For example, the user study did not evaluate the efficiency of navigation and exploration of structures of interest due to time constraints, which may involve finding a structure of interest, identifying various structures within a volume and following the path of a tubular structure. In addition, gathering a more diverse

group of participants, such as participants who are less familiar with 3D navigation or more familiar with image annotation, would provide additional insight into the usefulness of SHVR. Furthermore, SHVR should be validated against other interactive image annotation methods for accuracy, repeatability, and efficiency. This could involve exploring a set of samples, annotating a target structure, and comparing the results.

In summary, I introduced a method to visualize and annotate volumetric data using haptic feedback. A 6 DOF haptic device provides feedback on surface of volumetric objects, and along the centerline of tubular structures, which helps annotate target structures, and navigate through the sample intuitively. The system, SHVR, runs in real-time with the volumetric rendering handled by the GPU, and less demanding sequential tasks are handled using the CPU. This mixed approach makes the tool fast and interactive. After a review of literatures, the thesis provides details on the features of SHVR, including haptic assistance, annotation and visualization. Then, the case study showed annotation of several scans of flower heads. Finally, the user study showed that participants can reliably produces accurate annotations using haptic assistance.

# Bibliography

- 3D Systems. Specifications, n.d. URL [www.3dsystems.com/haptics-devices/touch/specifications](http://www.3dsystems.com/haptics-devices/touch/specifications).
- Sasakthi S. Abeysinghe and Tao Ju. Interactive skeletonization of intensity volumes. *The Visual Computer*, 25(5-7):627–635, May 2009. ISSN 0178-2789, 1432-2315. doi: 10.1007/s00371-009-0325-5. URL <http://link.springer.com/10.1007/s00371-009-0325-5>.
- Ludovica Acciai, Paolo Soda, and Giulio Iannello. Automated Neuron Tracing Methods: An Updated Account. *Neuroinformatics*, 14(4):353–367, October 2016. ISSN 1539-2791, 1559-0089. doi: 10.1007/s12021-016-9310-0. URL <http://link.springer.com/10.1007/s12021-016-9310-0>.
- M. Alkio and E. Grimm. Vascular connections between the receptacle and empty achenes in sunflower (*Helianthus annuus* L.). *Journal of Experimental Botany*, 54(381):345–348, January 2003. ISSN 0022-0957, 1460-2431. doi: 10.1093/jxb/erg019. URL <https://academic.oup.com/jxb/article-lookup/doi/10.1093/jxb/erg019>.
- R.S. Avila and L.M. Sobierajski. A haptic interaction method for volume visualization. In *Proceedings of Seventh Annual IEEE Visualization '96*, pages 197–204,, San Francisco, CA, USA, 1996. ACM. ISBN 978-0-89791-864-0. doi: 10.1109/VISUAL.1996.568108. URL <http://ieeexplore.ieee.org/document/568108/>.

Federico Barbagli, Sonny Chan, François Conti, Sébastien Grange, Dan Morris, Diego Ruspini, and Chris Sewell. Chai3D, n.d. URL <https://www.chai3d.org/>.

Pierre Barbier de Reuille, Anne-Lise Routier-Kierzkowska, Daniel Kierzkowski, George W Bassel, Thierry Schüpbach, Gerardo Tauriello, Namrata Bajpai, Sören Strauss, Alain Weber, Annamaria Kiss, Agata Burian, Hugo Hofhuis, Aleksandra Sapala, Marcin Lipowczan, Maria B Heimlicher, Sarah Robinson, Emmanuelle M Bayer, Konrad Basler, Petros Koumoutsakos, Adrienne HK Roeder, Tinri Aegerter-Wilmsen, Naomi Nakayama, Miltos Tsiantis, Angela Hay, Dorota Kwiatkowska, Ioannis Xenarios, Cris Kuhlemeier, and Richard S Smith. MorphoGraphX: A platform for quantifying morphogenesis in 4D. *eLife*, 4:e05864, May 2015. ISSN 2050-084X. doi: 10.7554/eLife.05864. URL <https://elifesciences.org/articles/05864>.

William A. Barrett and Eric N. Mortensen. Interactive live-wire boundary extraction. *Medical Image Analysis*, 1(4):331–341, September 1997. ISSN 13618415. doi: 10.1016/S1361-8415(97)85005-0. URL <https://linkinghub.elsevier.com/retrieve/pii/S1361841597850050>.

Daniel R. Berger, H. Sebastian Seung, and Jeff W. Lichtman. VAST (Volume Annotation and Segmentation Tool): Efficient Manual and Semi-Automatic Labeling of Large 3D Image Stacks. *Frontiers in Neural Circuits*, 12:88, October 2018. ISSN 1662-5110. doi: 10.3389/fncir.2018.00088. URL <https://www.frontiersin.org/article/10.3389/fncir.2018.00088/full>.

Antonio Bernardo and Alexander I. Evins. Neurosurgical Anatomy and Approaches to Simulation in Neurosurgical Training. In Ali Alaraj, editor, *Comprehensive Healthcare Simulation: Neurosurgery*, pages 219–244. Springer International Publishing, Cham, 2018. ISBN 978-3-319-75582-3 978-3-319-75583-0. doi: 10.1007/978-3-319-75583-0\_17. URL [http://link.springer.com/10.1007/978-3-319-75583-0\\_17](http://link.springer.com/10.1007/978-3-319-75583-0_17).

- Isabel Betegón-Putze, Alejandro González, Xavier Sevillano, David Blasco-Escámez, and Ana I. Caño-Delgado. MyROOT: A novel method and software for the semi-automatic measurement of plant root length. Preprint, Plant Biology, May 2018. URL <http://biorxiv.org/lookup/doi/10.1101/309773>.
- James F. Blinn. Models of light reflection for computer synthesized pictures. *ACM SIGGRAPH Computer Graphics*, 11(2):192–198, August 1977. ISSN 0097-8930. doi: 10.1145/965141.563893. URL <https://dl.acm.org/doi/10.1145/965141.563893>.
- Yuri Boykov and Marie-Pierre Jolly. Interactive Organ Segmentation Using Graph Cuts. In Scott L. Delp, Anthony M. DiGoia, Branislav Jaramaz, Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2000*, volume 1935, pages 276–286. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000. ISBN 978-3-540-41189-5 978-3-540-40899-4. doi: 10.1007/978-3-540-40899-4\_28. URL [http://link.springer.com/10.1007/978-3-540-40899-4\\_28](http://link.springer.com/10.1007/978-3-540-40899-4_28).
- Xavier Bresson, Selim Esedoğlu, Pierre Vanderghenst, Jean-Philippe Thiran, and Stanley Osher. Fast Global Minimization of the Active Contour/Snake Model. *Journal of Mathematical Imaging and Vision*, 28(2):151–167, August 2007. ISSN 0924-9907, 1573-7683. doi: 10.1007/s10851-007-0002-0. URL <http://link.springer.com/10.1007/s10851-007-0002-0>.
- Alessandro Bria, Giulio Iannello, Leonardo Onofri, and Hanchuan Peng. TeraFly: Real-time three-dimensional visualization and annotation of terabytes of multidimensional volumetric images. *Nature Methods*, 13(3):192–194, March 2016. ISSN 1548-7091, 1548-7105. doi: 10.1038/nmeth.3767. URL <http://www.nature.com/articles/nmeth.3767>.
- C. R. Brodersen and A. B. Roddy. New frontiers in the three-dimensional visualization of plant structure and function. *American Journal of Botany*, 103(2):184–188, February

2016. ISSN 0002-9122, 1537-2197. doi: 10.3732/ajb.1500532. URL <http://doi.wiley.com/10.3732/ajb.1500532>.

Craig R. Brodersen, Eric F. Lee, Brendan Choat, Steven Jansen, Ronald J. Phillips, Kenneth A. Shackel, Andrew J. McElrone, and Mark A. Matthews. Automated analysis of three-dimensional xylem networks using high-resolution computed tomography. *New Phytologist*, 191(4):1168–1179, September 2011. ISSN 0028-646X, 1469-8137. doi: 10.1111/j.1469-8137.2011.03754.x. URL <https://onlinelibrary.wiley.com/doi/10.1111/j.1469-8137.2011.03754.x>.

Craig R. Brodersen, Lindsey C. Roark, and Jarmila Pittermann. The physiological implications of primary xylem organization in two ferns: Fern xylem and physiology. *Plant, Cell & Environment*, 35(11):1898–1911, November 2012. ISSN 01407791. doi: 10.1111/j.1365-3040.2012.02524.x. URL <https://onlinelibrary.wiley.com/doi/10.1111/j.1365-3040.2012.02524.x>.

Jonas Bühler, Louai Rishmawi, Daniel Pflugfelder, Gregor Huber, Hanno Scharr, Martin Hülskamp, Maarten Koornneef, Ulrich Schurr, and Siegfried Jahnke. phenoVein - A tool for leaf vein segmentation and analysis. *Plant Physiology*, page pp.00974.2015, October 2015. ISSN 0032-0889, 1532-2548. doi: 10.1104/pp.15.00974. URL <https://academic.oup.com/plphys/article/169/4/2359-2370/6114117>.

Michael Burns, Janek Klawe, Szymon Rusinkiewicz, Adam Finkelstein, and Doug DeCarlo. Line drawings from volume data. *ACM Transactions on Graphics*, 24(3):512–518, July 2005. ISSN 0730-0301, 1557-7368. doi: 10.1145/1073204.1073222. URL <https://dl.acm.org/doi/10.1145/1073204.1073222>.

Paul S. Calhoun, Brian S. Kuszyk, David G. Heath, Jennifer C. Carley, and Elliot K. Fishman. Three-dimensional Volume Rendering of Spiral CT Data: Theory and Method. *RadioGraphics*, 19(3):745–764, May 1999. ISSN 0271-5333, 1527-1323. doi:

10.1148/radiographics.19.3.g99ma14745. URL <http://pubs.rsna.org/doi/10.1148/radiographics.19.3.g99ma14745>.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, volume 9901, pages 424–432. Springer International Publishing, Cham, 2016. ISBN 978-3-319-46722-1 978-3-319-46723-8. doi: 10.1007/978-3-319-46723-8\_49. URL [http://link.springer.com/10.1007/978-3-319-46723-8\\_49](http://link.springer.com/10.1007/978-3-319-46723-8_49).

Loic Corenthy, Miguel A. Otaduy, Luis Pastor, and Marcos Garcia. Volume Haptics with Topology-Consistent Isosurfaces. *IEEE Transactions on Haptics*, 8(4):480–491, October 2015. ISSN 1939-1412. doi: 10.1109/TOH.2015.2466239. URL <http://ieeexplore.ieee.org/document/7182778/>.

Loic Corenthy, Marcos Garcia, Sofia Bayona, Andrea Santuy, Jose San Martin, Ruth Benavides-Piccione, Javier DeFelipe, and Luis Pastor. Haptically Assisted Connection Procedure for the Reconstruction of Dendritic Spines. *IEEE Transactions on Haptics*, 7(4):486–498, October 2014. ISSN 1939-1412. doi: 10.1109/TOH.2014.2354041. URL <http://ieeexplore.ieee.org/document/6891304/>.

Heather Culbertson, Samuel B. Schorr, and Allison M. Okamura. Haptics: The Present and Future of Artificial Touch Sensation. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):385–409, May 2018. ISSN 2573-5144. doi: 10.1146/annurev-control-060117-105043. URL <https://www.annualreviews.org/doi/10.1146/annurev-control-060117-105043>.

Matteo de Notaris, Kenneth Palma, Luis Serra, Joaquim Enseñat, Isam Alobid, José Poblete, Joan Berenguer Gonzalez, Domenico Solari, Enrique Ferrer, and Alberto Prats-Galino. A

Three-Dimensional Computer-Based Perspective of the Skull Base. *World Neurosurgery*, 82(6):S41–S48, December 2014. ISSN 18788750. doi: 10.1016/j.wneu.2014.07.024. URL <https://linkinghub.elsevier.com/retrieve/pii/S1878875014006846>.

S Deng, E Singh, G Wheeler, K Pushparajah, J A Schnabel, J M Simpson, and A Gomez Herero. Evaluation of haptic feedback for interaction with volumetric image data in virtual reality. *European Heart Journal - Cardiovascular Imaging*, 21(Supplement\_1):jez319.986, January 2020. ISSN 2047-2404, 2047-2412. doi: 10.1093/ehjci/jez319.986. URL <https://academic.oup.com/ehjcmimaging/article/doi/10.1093/ehjci/jez319.986/5708809>.

Nancy Dengler and Julie Kang. Vascular patterning and leaf shape. *Current Opinion in Plant Biology*, 4(1):50–56, February 2001. ISSN 13695266. doi: 10.1016/S1369-5266(00)00135-7. URL <https://linkinghub.elsevier.com/retrieve/pii/S1369526600001357>.

Stijn Dhondt, Hannes Vanhaeren, Denis Van Loo, Veerle Cnudde, and Dirk Inzé. Plant structure visualization by high-resolution X-ray computed tomography. *Trends in Plant Science*, 15(8):419–422, August 2010. ISSN 13601385. doi: 10.1016/j.tplants.2010.05.002. URL <https://linkinghub.elsevier.com/retrieve/pii/S1360138510000956>.

Lee R. Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302, July 1945. ISSN 00129658. doi: 10.2307/1932409. URL <http://doi.wiley.com/10.2307/1932409>.

Keith E Duncan, Kirk J Czymbek, Ni Jiang, August C Thies, and Christopher N Topp. X-ray microscopy enables multiscale high-resolution 3D imaging of plant cells, tissues, and organs. *Plant Physiology*, 188(2):831–845, February 2022. ISSN 0032-0889, 1532-2548. doi: 10.1093/plphys/kiab405. URL <https://academic.oup.com/plphys/article/188/2/831/6375950>.

Kevin W Eliceiri, Michael R Berthold, Ilya G Goldberg, Luis Ibáñez, B S Manjunath,

- Maryann E Martone, Robert F Murphy, Hanchuan Peng, Anne L Plant, Badrinath Roysam, Nico Stuurman, Jason R Swedlow, Pavel Tomancak, and Anne E Carpenter. Biological imaging software tools. *Nature Methods*, 9(7):697–710, July 2012. ISSN 1548-7091, 1548-7105. doi: 10.1038/nmeth.2084. URL <http://www.nature.com/articles/nmeth.2084>.
- Nima Enayati, Elena De Momi, and Giancarlo Ferrigno. Haptics in Robot-Assisted Surgery: Challenges and Benefits. *IEEE Reviews in Biomedical Engineering*, 9:49–65, 2016. ISSN 1937-3333, 1941-1189. doi: 10.1109/RBME.2016.2538080. URL <http://ieeexplore.ieee.org/document/7425205/>.
- Rickard Englund, Karljohan Lundin Palmerius, Ingrid Hotz, and Anders Ynnerman. Touching Data: Enhancing Visual Exploration of Flow Data with Haptics. *Computing in Science & Engineering*, 20(3):89–100, May 2018. ISSN 1521-9615. doi: 10.1109/MCSE.2018.03221931. URL <https://ieeexplore.ieee.org/document/8358030/>.
- Marc O. Ernst. A Bayesian view on multimodal cue integration. In Günther Knoblich, Ian M. Thornton, Marc Grosjean, and Maggie Shiffrar, editors, *Human Body Perception From the Inside Out*, pages 105–131. Oxford University Press, 2006.
- Josep M. Espadaler and Gérardo Conesa. Navigated repetitive Transcranial Magnetic Stimulation (TMS) for language mapping: A new tool for surgical planning. In Hugues Duffau, editor, *Brain Mapping*, pages 253–261. Springer Vienna, Vienna, 2011. ISBN 978-3-7091-0722-5 978-3-7091-0723-2. doi: 10.1007/978-3-7091-0723-2\_19. URL [http://link.springer.com/10.1007/978-3-7091-0723-2\\_19](http://link.springer.com/10.1007/978-3-7091-0723-2_19).
- Balázs Faludi, Esther I. Zoller, Nicolas Gerig, Azhar Zam, Georg Rauter, and Philippe C. Cattin. Direct Visual and Haptic Volume Rendering of Medical Data Sets for an Immersive Exploration in Virtual Reality. In Dinggang Shen, Tianming Liu, Terry M. Peters, Lawrence H. Staib, Caroline Essert, Sean Zhou, Pew-Thian Yap, and Ali Khan, edi-

tors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, volume 11768, pages 29–37. Springer International Publishing, Cham, 2019. ISBN 978-3-030-32253-3 978-3-030-32254-0. doi: 10.1007/978-3-030-32254-0\_4. URL [http://link.springer.com/10.1007/978-3-030-32254-0\\_4](http://link.springer.com/10.1007/978-3-030-32254-0_4).

Te-Yung Fang, Pa-Chun Wang, Chih-Hsien Liu, Mu-Chun Su, and Shih-Ching Yeh. Evaluation of a haptics-based virtual reality temporal bone simulator for anatomy and surgery training. *Computer Methods and Programs in Biomedicine*, 113(2):674–681, February 2014. ISSN 01692607. doi: 10.1016/j.cmpb.2013.11.005. URL <https://linkinghub.elsevier.com/retrieve/pii/S0169260713003775>.

Alejandro F. Frangi, Wiro J. Niessen, Koen L. Vincken, and Max A. Viergever. Multiscale vessel enhancement filtering. In William M. Wells, Alan Colchester, and Scott Delp, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI’98*, volume 1496, pages 130–137. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. ISBN 978-3-540-65136-9 978-3-540-49563-5. doi: 10.1007/BFb0056195. URL <http://link.springer.com/10.1007/BFb0056195>.

Fred Churchville. What is user interface, n.d. URL [www.techtarget.com/searchapparchitecture/definition/user-interface-UI](http://www.techtarget.com/searchapparchitecture/definition/user-interface-UI).

Fundamental Surgery. HapticVR, n.d. URL <https://fundamentalsurgery.com/>.

Jason Gardiner, Ira Sherr, and Enrico Scarpella. Expression of DOF genes identifies early stages of vascular development in Arabidopsis leaves. *The International Journal of Developmental Biology*, 54(8-9):1389–1396, 2010. ISSN 0214-6282. doi: 10.1387/ijdb.093006jg. URL <http://www.intjdevbiol.com/paper.php?doi=093006jg>.

Rocco Gasteiger, Christian Tietjen, Alexandra Baer, and Bernhard Preim. Curvature- and Model-Based Surface Hatching of Anatomical Structures Derived from Clinical

- Volume Datasets. In Andreas Butz, Brian Fisher, Antonio Krüger, Patrick Olivier, and Marc Christie, editors, *Smart Graphics*, volume 5166, pages 255–262. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-85410-4 978-3-540-85412-8. doi: 10.1007/978-3-540-85412-8\_25. URL [http://link.springer.com/10.1007/978-3-540-85412-8\\_25](http://link.springer.com/10.1007/978-3-540-85412-8_25).
- Deanna H. Gates and Jonathan B. Dingwell. The effects of muscle fatigue and movement height on movement stability and variability. *Experimental Brain Research*, 209(4):525–536, April 2011. ISSN 0014-4819, 1432-1106. doi: 10.1007/s00221-011-2580-8. URL <http://link.springer.com/10.1007/s00221-011-2580-8>.
- Carole T. Gee. Applying microCT and 3D Visualization to Jurassic Silicified Conifer Seed Cones: A Virtual Advantage Over Thin-Sectioning. *Applications in Plant Sciences*, 1(11):1300039, November 2013. ISSN 2168-0450. doi: 10.3732/apps.1300039. URL <http://doi.wiley.com/10.3732/apps.1300039>.
- Todd A. Gillette, Kerry M. Brown, and Giorgio A. Ascoli. The DIADEM Metric: Comparing Multiple Reconstructions of the Same Neuron. *Neuroinformatics*, 9(2-3):233–245, September 2011. ISSN 1539-2791, 1559-0089. doi: 10.1007/s12021-011-9117-y. URL <http://link.springer.com/10.1007/s12021-011-9117-y>.
- L. Grady. Random Walks for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, November 2006. ISSN 0162-8828, 2160-9292. doi: 10.1109/TPAMI.2006.233. URL <http://ieeexplore.ieee.org/document/1704833/>.
- Nehemiah Grew. *The Anatomy of Plants*. Printed by W. Rawlins, for the author,, [London] :, 1682. doi: 10.5962/bhl.title.4. URL <http://www.biodiversitylibrary.org/bibliography/4>.

- M. Hadwiger, J. Beyer, Won-Ki Jeong, and H. Pfister. Interactive Volume Exploration of Petascale Microscopy Data Streams Using a Visualization-Driven Virtual Memory Approach. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2285–2294, December 2012. ISSN 1077-2626. doi: 10.1109/TVCG.2012.240. URL <http://ieeexplore.ieee.org/document/6327233/>.
- Markus Hadwiger, Patric Ljung, Christof Rezk Salama, and Timo Ropinski. GPU-Based Volume Ray-Casting with Advanced Illumination. *Eurographics 2009 - Tutorials*, page 173 pages, 2009. ISSN 1017-4656. doi: 10.2312/EGT.20091069. URL <https://diglib.eg.org/handle/10.2312/egt.20091069.039-211>.
- M. Hardens and G. Szekely. Enhancing human-computer interaction in medical segmentation. *Proceedings of the IEEE*, 91(9):1430–1442, September 2003. ISSN 0018-9219. doi: 10.1109/JPROC.2003.817125. URL <http://ieeexplore.ieee.org/document/1230218/>.
- Pheng-Ann Heng, Hanqiu Sun, Kwong-Wai Chen, and Tien-Tsin Wong. Interactive Navigation of Virtual Vessel Tracking with 3D Intelligent Scissors. *International Journal of Image and Graphics*, 01(02):273–285, April 2001. ISSN 0219-4678, 1793-6756. doi: 10.1142/S0219467801000177. URL <https://www.worldscientific.com/doi/abs/10.1142/S0219467801000177>.
- Christoph Hennersperger and Maximilian Baust. Play for Me: Image Segmentation via Seamless Playsourcing. *The Computer Games Journal*, 6(1-2):1–16, June 2017. ISSN 2052-773X. doi: 10.1007/s40869-016-0030-3. URL <http://link.springer.com/10.1007/s40869-016-0030-3>.
- Frida Hernell, Patric Ljung, and Anders Ynnerman. Efficient Ambient and Emissive Tissue Illumination using Local Occlusion in Multiresolution Volume Rendering. *Eurographics/IEEE VGTC Symposium on Volume Graphics*, page 8 pages, 2007. ISSN 1727-8376.

doi: 10.2312/VG/VG07/001-008. URL <http://diglib.eg.org/handle/10.2312/VG.VG07.001-008>.

Frida Hernell, Patric Ljung, and Anders Ynnerman. Local Ambient Occlusion in Direct Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 16(4):548–559, July 2010. ISSN 1077-2626. doi: 10.1109/TVCG.2009.45. URL <http://ieeexplore.ieee.org/document/4840341/>.

Els Herremans, Pieter Verboven, Maarten L. A. T. M. Hertog, Dennis Cantre, Mattias van Dael, Thomas De Schryver, Luc Van Hoorebeke, and Bart M. Nicolai. Spatial development of transport structures in apple (*Malus x domestica* Borkh.) fruit. *Frontiers in Plant Science*, 6, September 2015. ISSN 1664-462X. doi: 10.3389/fpls.2015.00679. URL <http://journal.frontiersin.org/article/10.3389/fpls.2015.00679>.

Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. Consumed endurance: A metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1063–1072, Toronto Ontario Canada, April 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557130. URL <https://dl.acm.org/doi/10.1145/2556288.2557130>.

Mian Huang and Ghassan Hamarneh. SwifTree: Interactive Extraction of 3D Trees Supporting Gaming and Crowdsourcing. In M. Jorge Cardoso, Tal Arbel, Su-Lin Lee, Veronika Cheplygina, Simone Balocco, Diana Mateus, Guillaume Zahnd, Lena Maier-Hein, Stefanie Demirci, Eric Granger, Luc Duong, Marc-André Carbonneau, Shadi Albarqouni, and Gustavo Carneiro, editors, *Intravascular Imaging and Computer Assisted Stenting, and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*, volume 10552, pages 116–125. Springer International Publishing, Cham, 2017. ISBN 978-3-319-67533-6 978-3-319-67534-3. doi: 10.1007/978-3-319-67534-3\_13. URL [http://link.springer.com/10.1007/978-3-319-67534-3\\_13](http://link.springer.com/10.1007/978-3-319-67534-3_13).

Maximilian Kaluschke, Rene Weller, Niels Hammer, Luigi Pelliccia, Mario Lorenz, and Gabriel Zachmann. Realistic Haptic Feedback for Material Removal in Medical Simulations. In *2020 IEEE Haptics Symposium (HAPTICS)*, pages 920–926, Crystal City, VA, USA, March 2020. IEEE. ISBN 978-1-72810-234-4. doi: 10.1109/HAPTICS45997.2020.ras.HAP20.74.13165668. URL <https://ieeexplore.ieee.org/document/9086302/>.

Julie Kang, John Tang, Petra Donnelly, and Nancy Dengler. Primary vascular pattern and expression of ATHB-8 in shoots of Arabidopsis. *New Phytologist*, 158(3):443–454, May 2003. ISSN 0028646X, 14698137. doi: 10.1046/j.1469-8137.2003.00769.x. URL <http://doi.wiley.com/10.1046/j.1469-8137.2003.00769.x>.

Ichirou Karahara, Daisuke Yamauchi, Kentaro Uesugi, and Yoshinobu Mineyuki. Three-dimensional imaging of plant tissues using X-ray micro-computed tomography. *PLANT MORPHOLOGY*, 27(1):21–26, 2015. ISSN 0918-9726, 1884-4154. doi: 10.5685/plmorphol.27.21. URL [https://www.jstage.jst.go.jp/article/plmorphol/27/1/27\\_21/\\_article](https://www.jstage.jst.go.jp/article/plmorphol/27/1/27_21/_article).

Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, January 1988. ISSN 0920-5691, 1573-1405. doi: 10.1007/BF00133570. URL <http://link.springer.com/10.1007/BF00133570>.

G. Keshavareddy, A.R.V. Kumar, and Vemanna S. Ramu. Methods of Plant Transformation-A Review. *International Journal of Current Microbiology and Applied Sciences*, 7(07): 2656–2668, July 2018. ISSN 23197692, 23197706. doi: 10.20546/ijcmas.2018.707.312. URL <https://www.ijcmas.com/abstractview.php?ID=8916&vol=7-7-2018&SNo=312>.

Khronos Group. OpenGL, n.d. URL <https://www.opengl.org/>.

J. Kniss, G. Kindlmann, and C. Hansen. Multidimensional transfer functions for interactive

volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3): 270–285, July 2002. ISSN 1077-2626. doi: 10.1109/TVCG.2002.1021579. URL <http://ieeexplore.ieee.org/document/1021579/>.

Aaron T. Kuan, Jasper S. Phelps, Logan A. Thomas, Tri M. Nguyen, Julie Han, Chiao-Lin Chen, Anthony W. Azevedo, John C. Tuthill, Jan Funke, Peter Cloetens, Alexandra Pacureanu, and Wei-Chung Allen Lee. Dense neuronal reconstruction through X-ray holographic nano-tomography. *Nature Neuroscience*, 23(12):1637–1643, December 2020. ISSN 1097-6256, 1546-1726. doi: 10.1038/s41593-020-0704-9. URL <https://www.nature.com/articles/s41593-020-0704-9>.

Labelbox. Image Annotation 101, n.d. URL <https://labelbox.com/image-annotation-overview>.

Gunnar Lathen, Stefan Lindholm, Reiner Lenz, Anders Persson, and Magnus Borga. Automatic Tuning of Spatially Varying Transfer Functions for Blood Vessel Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2345–2354, December 2012. ISSN 1077-2626. doi: 10.1109/TVCG.2012.203. URL <http://ieeexplore.ieee.org/document/6327239/>.

Tsui-Ying Law and PhengAnn Heng. Automated extraction of bronchus from 3D CT images of lung based on genetic algorithm and 3D region growing. In Kenneth M. Hanson, editor, *Medical Imaging 2000*, pages 906–916, San Diego, CA, June 2000. doi: 10.1117/12.387756. URL <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=923079>.

T.C. Lee, R.L. Kashyap, and C.N. Chu. Building Skeleton Models via 3-D Medial Surface Axis Thinning Algorithms. *CVGIP: Graphical Models and Image Processing*, 56(6):462–478, November 1994. ISSN 10499652. doi: 10.1006/cgip.1994.1042. URL <https://linkinghub.elsevier.com/retrieve/pii/S104996528471042X>.

- David Lesage, Elsa D. Angelini, Isabelle Bloch, and Gareth Funka-Lea. A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes. *Medical Image Analysis*, 13(6):819–845, December 2009. ISSN 13618415. doi: 10.1016/j.media.2009.07.011. URL <https://linkinghub.elsevier.com/retrieve/pii/S136184150900067X>.
- Rui Li, Muye Zhu, Junning Li, Michael S. Bienkowski, Nicholas N. Foster, Hanpeng Xu, Tyler Ard, Ian Bowman, Changle Zhou, Matthew B. Veldman, X. William Yang, Hourii Hintiryan, Junsong Zhang, and Hong-Wei Dong. Precise segmentation of densely interweaving neuron clusters using G-Cut. *Nature Communications*, 10(1):1549, December 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-09515-0. URL <http://www.nature.com/articles/s41467-019-09515-0>.
- Zhenxing Li, Maria Kiiveri, Jussi Rantala, and Roope Raisamo. Evaluation of haptic virtual reality user interfaces for medical marking on 3D models. *International Journal of Human-Computer Studies*, 147:102561, March 2021. ISSN 10715819. doi: 10.1016/j.ijhcs.2020.102561. URL <https://linkinghub.elsevier.com/retrieve/pii/S1071581920301634>.
- J. C. R. Licklider. Man-Computer Symbiosis. *IRE Transactions on Human Factors in Electronics*, HFE-1(1):4–11, March 1960. ISSN 0099-4561, 2168-2836. doi: 10.1109/THFE2.1960.4503259. URL <http://ieeexplore.ieee.org/document/4503259/>.
- David Livingston, Tan Tuong, Marco Nogueira, and Thomas Sinclair. Three-dimensional reconstruction of soybean nodules provides an update on vascular structure. *American Journal of Botany*, 106(3):507–513, March 2019. ISSN 0002-9122, 1537-2197. doi: 10.1002/ajb2.1249. URL <https://onlinelibrary.wiley.com/doi/10.1002/ajb2.1249>.
- K. Lundin, B. Gudmundsson, and A. Ynnerman. General Proxy-Based Haptics for Volume Visualization. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 557–560, Pisa, Italy, 2005. IEEE.

ISBN 978-0-7695-2310-1. doi: 10.1109/WHC.2005.62. URL <http://ieeexplore.ieee.org/document/1407007/>.

Karljohan Lundin, Anders Ynnerman, and Bjorn Gudmundsson. Proxy based Haptic Feedback from Volumetric Density Data. In *Proceedings of the Eurohaptic Conference*, pages 104–109, University of Edinburgh, United Kingdom, July 2002. URL [https://www.researchgate.net/publication/2919496\\_Proxy-based\\_Haptic\\_Feedback\\_from\\_Volumetric\\_Density\\_Data](https://www.researchgate.net/publication/2919496_Proxy-based_Haptic_Feedback_from_Volumetric_Density_Data).

Matthew M. Mahan and Amber L. Doiron. Gold Nanoparticles as X-Ray, CT, and Multimodal Imaging Contrast Agents: Formulation, Targeting, and Methodology. *Journal of Nanomaterials*, 2018:1–15, 2018. ISSN 1687-4110, 1687-4129. doi: 10.1155/2018/5837276. URL <https://www.hindawi.com/journals/jnm/2018/5837276/>.

Filip Malmberg, Erik Vidholm, and Ingela Nyström. A 3D Live-Wire Segmentation Method for Volume Images Using Haptic Interaction. In Attila Kuba, László G. Nyúl, Kálmán Palágyi, David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, and Gerhard Weikum, editors, *Discrete Geometry for Computer Imagery*, volume 4245, pages 663–673. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-47651-1 978-3-540-47652-8. doi: 10.1007/11907350\_56. URL [http://link.springer.com/10.1007/11907350\\_56](http://link.springer.com/10.1007/11907350_56).

MathWorks. Types of Morphological Operations. URL <https://www.mathworks.com/help/images/morphological-dilation-and-erosion.html>.

Kevin McGuinness and Noel E. O’Connor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444, February 2010. ISSN

00313203. doi: 10.1016/j.patcog.2009.03.008. URL <https://linkinghub.elsevier.com/retrieve/pii/S0031320309000818>.

Erik Meijering. Neuron tracing in perspective. *Cytometry Part A*, 77A(7):693–704, March 2010. ISSN 15524922, 15524930. doi: 10.1002/cyto.a.20895. URL <https://onlinelibrary.wiley.com/doi/10.1002/cyto.a.20895>.

Meta. Reality Labs Haptic Gloves. URL <https://tech.fb.com/inside-reality-labs-meet-the-team-thats-bringing-touch-to-the-digital-world/>.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571, Stanford, CA, USA, October 2016. IEEE. ISBN 978-1-5090-5407-7. doi: 10.1109/3DV.2016.79. URL <http://ieeexplore.ieee.org/document/7785132/>.

Javier A. Montoya-Zegarra, Erica Russo, Peter Runge, Maria Jadhav, Ann-Helen Willrodt, Szymon Stoma, Simon F. Nørrelykke, Michael Detmar, and Cornelia Halin. AutoTube: A novel software for the automated morphometric analysis of vascular networks in tissues. *Angiogenesis*, 22(2):223–236, May 2019. ISSN 0969-6970, 1573-7209. doi: 10.1007/s10456-018-9652-3. URL <http://link.springer.com/10.1007/s10456-018-9652-3>.

Satoshi Moriwaki, Yasuhiko Terada, Katsumi Kose, Tomoyuki Haishi, and Yoshihiko Sekozawa. Visualization and Quantification of Vascular Structure of Fruit Using Magnetic Resonance Microimaging. *Applied Magnetic Resonance*, 45(6):517–525, June 2014. ISSN 0937-9347, 1613-7507. doi: 10.1007/s00723-014-0537-3. URL <http://link.springer.com/10.1007/s00723-014-0537-3>.

Dan Morris, Hong Tan, Federico Barbagli, Timothy Chang, and Kenneth Salisbury. Haptic Feedback Enhances Force Skill Learning. In *Second Joint EuroHaptics Conference*

and *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*, pages 21–26, Tsukuba, March 2007. IEEE. ISBN 978-0-7695-2738-3. doi: 10.1109/WHC.2007.65. URL <http://ieeexplore.ieee.org/document/4145145/>.

National Institute of Biomedical Imaging and Bioengineering. Computed Tomography. URL <https://www.nibib.nih.gov/science-education/science-topics/computed-tomography-ct>.

Tan Khoa Nguyen, Anders Eklund, Henrik Ohlsson, Frida Hernell, Patric Ljung, Camilla Forsell, Mats Andersson, Hans Knutsson, and Anders Ynnerman. Concurrent Volume Visualization of Real-Time fMRI. *IEEE/ EG Symposium on Volume Graphics*, page 8 pages, 2010. ISSN 1727-8376. doi: 10.2312/VG/VG10/053-060. URL <http://diglib.eg.org/handle/10.2312/VG.VG10.053-060>.

S.D Olabarriaga and A.W.M Smeulders. Interaction in the segmentation of medical images: A survey. *Medical Image Analysis*, 5(2):127–142, June 2001. ISSN 13618415. doi: 10.1016/S1361-8415(00)00041-4. URL <https://linkinghub.elsevier.com/retrieve/pii/S1361841500000414>.

Ian Ormesher. Convolution Filters, May 2020. URL <https://medium.com/@ianormy/convolution-filters-4971820e851f>.

Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, January 1979. ISSN 0018-9472, 2168-2909. doi: 10.1109/TSMC.1979.4310076. URL <http://ieeexplore.ieee.org/document/4310076/>.

Hanchuan Peng, Fuhui Long, Ting Zhao, and Eugene Myers. Proof-editing is the Bottleneck Of 3D Neuron Reconstruction: The Problem and Solutions. *Neuroinformatics*, 9(2-3):

103–105, September 2011. ISSN 1539-2791, 1559-0089. doi: 10.1007/s12021-010-9090-x.  
URL <http://link.springer.com/10.1007/s12021-010-9090-x>.

Hanchuan Peng, Alessandro Bria, Zhi Zhou, Giulio Iannello, and Fuhui Long. Extensible visualization and analysis for multidimensional images using Vaa3D. *Nature Protocols*, 9(1):193–208, January 2014a. ISSN 1754-2189, 1750-2799. doi: 10.1038/nprot.2014.011.  
URL <http://www.nature.com/articles/nprot.2014.011>.

Hanchuan Peng, Jianyong Tang, Hang Xiao, Alessandro Bria, Jianlong Zhou, Victoria Butler, Zhi Zhou, Paloma T. Gonzalez-Bellido, Seung W. Oh, Jichao Chen, Ananya Mitra, Richard W. Tsien, Hongkui Zeng, Giorgio A. Ascoli, Giulio Iannello, Michael Hawrylycz, Eugene Myers, and Fuhui Long. Virtual finger boosts three-dimensional imaging and microsurgery as well as terabyte volume image visualization and analysis. *Nature Communications*, 5(1):4342, September 2014b. ISSN 2041-1723. doi: 10.1038/ncomms5342.  
URL <http://www.nature.com/articles/ncomms5342>.

Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, June 1975. ISSN 0001-0782, 1557-7317. doi: 10.1145/360825.360839.  
URL <https://dl.acm.org/doi/10.1145/360825.360839>.

Nicolas Piche, Isabelle Bouchard, and Mike Marsh. Dragonfly SegmentationTrainer - A General and User-Friendly Machine Learning Image Segmentation Solution. *Microscopy and Microanalysis*, 23(S1):132–133, July 2017. ISSN 1431-9276, 1435-8115. doi: 10.1017/S1431927617001349. URL [https://www.cambridge.org/core/product/identifier/S1431927617001349/type/journal\\_article](https://www.cambridge.org/core/product/identifier/S1431927617001349/type/journal_article).

Csaba Pinter, Andras Lasso, Saleh Choueib, Mark Asselin, Jean-Christophe Fillion-Robin, Jean-Baptiste Vimort, Ken Martin, Matthew A. Jolley, and Gabor Fichtinger. SlicerVR for Medical Intervention Training and Planning in Immersive Virtual Reality. *IEEE Transac-*

*tions on Medical Robotics and Bionics*, 2(2):108–117, May 2020. ISSN 2576-3202. doi: 10.1109/TMRB.2020.2983199. URL <https://ieeexplore.ieee.org/document/9047949/>.

Regina Pohle and Klaus D. Toennies. Segmentation of medical images using adaptive region growing. In Milan Sonka and Kenneth M. Hanson, editors, *Medical Imaging 2001*, pages 1337–1346, San Diego, CA, July 2001. doi: 10.1117/12.431013. URL <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=906741>.

Kelvin Poon, Ghassan Hamarneh, and Rafeef Abugharbieh. Live-Vessel: Extending Livewire for Simultaneous Extraction of Optimal Medial and Boundary Paths in Vascular Images. In Nicholas Ayache, Sébastien Ourselin, and Anthony Maeder, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007*, volume 4792, pages 444–451. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-75758-0. doi: 10.1007/978-3-540-75759-7\_54. URL [http://link.springer.com/10.1007/978-3-540-75759-7\\_54](http://link.springer.com/10.1007/978-3-540-75759-7_54).

Bernhard Preim and Charl Botha. *Visual Computing for Medicine: Theory, Algorithms, and Applications*. Elsevier, Amsterdam ; Boston, second edition edition, 2014. ISBN 978-0-12-415873-3.

P. Rheingans and D. Ebert. Volume illustration: Nonphotorealistic rendering of volume models. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):253–264, July 2001. ISSN 10772626. doi: 10.1109/2945.942693. URL <http://ieeexplore.ieee.org/document/942693/>.

Felix Ritter, Tobias Boskamp, A. Homeyer, Hendrik Laue, Michael Schwier, Florian Link, and H.-O. Peitgen. Medical Image Analysis. *IEEE Pulse*, 2(6):60–70, November 2011. ISSN 2154-2287, 2154-2317. doi: 10.1109/MPUL.2011.942929. URL <https://ieeexplore.ieee.org/document/6088923/>.

- Punam K. Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja. A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters*, 76:3–12, June 2016. ISSN 01678655. doi: 10.1016/j.patrec.2015.04.006. URL <https://linkinghub.elsevier.com/retrieve/pii/S0167865515001233>.
- K. Salisbury, F. Conti, and F. Barbagli. Haptic rendering: Introductory concepts. *IEEE Computer Graphics and Applications*, 24(2):24–32, March 2004. ISSN 0272-1716. doi: 10.1109/MCG.2004.1274058. URL <http://ieeexplore.ieee.org/document/1274058/>.
- Mirko Sattler, Ralf Sarlette, Thomas Mücken, and Reinhard Klein. Exploitation of human shadow perception for fast shadow rendering. In *Proceedings of the 2nd Symposium on Applied Perception in Graphics and Visualization - APGV '05*, page 131, A Coruña, Spain, 2005. ACM Press. ISBN 978-1-59593-139-9. doi: 10.1145/1080402.1080426. URL <http://portal.acm.org/citation.cfm?doid=1080402.1080426>.
- Enrico Scarpella, Danielle Marcos, Jiří Friml, and Thomas Berleth. Control of leaf vascular patterning by polar auxin transport. *Genes & Development*, 20(8):1015–1027, April 2006. ISSN 0890-9369, 1549-5477. doi: 10.1101/gad.1402406. URL <http://genesdev.cshlp.org/lookup/doi/10.1101/gad.1402406>.
- Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, Jean-Yves Tinevez, Daniel James White, Volker Hartenstein, Kevin Eliceiri, Pavel Tomancak, and Albert Cardona. Fiji: An open-source platform for biological-image analysis. *Nature Methods*, 9(7):676–682, July 2012. ISSN 1548-7091, 1548-7105. doi: 10.1038/nmeth.2019. URL <http://www.nature.com/articles/nmeth.2019>.
- Hyunseok Seo, Masoud Badieli Khuzani, Varun Vasudevan, Charles Huang, Hongyi Ren, Ruoxiu Xiao, Xiao Jia, and Lei Xing. Machine learning techniques for biomedical image segmentation: An overview of technical aspects and introduction to state-of-art

applications. *Medical Physics*, 47(5), May 2020. ISSN 0094-2405, 2473-4209. doi: 10.1002/mp.13649. URL <https://onlinelibrary.wiley.com/doi/10.1002/mp.13649>.

Carlos M. Serrano, Paul R. Wesselink, and Johanna M. Vervoorn. First experiences with patient-centered training in virtual reality. *Journal of Dental Education*, 84(5):607–614, May 2020. ISSN 00220337. doi: 10.1002/jdd.12037. URL <https://onlinelibrary.wiley.com/doi/10.1002/jdd.12037>.

James Sharpe. Optical Projection Tomography. *Annual Review of Biomedical Engineering*, 6(1):209–228, August 2004. ISSN 1523-9829, 1545-4274. doi: 10.1146/annurev.bioeng.6.040803.140210. URL <http://www.annualreviews.org/doi/10.1146/annurev.bioeng.6.040803.140210>.

Wen Shi, Peter Xiaoping Liu, and Minhua Zheng. Cutting procedures with improved visual effects and haptic interaction for surgical simulation systems. *Computer Methods and Programs in Biomedicine*, 184:105270, February 2020. ISSN 01692607. doi: 10.1016/j.cmpb.2019.105270. URL <https://linkinghub.elsevier.com/retrieve/pii/S0169260719317444>.

Alexa F. Siu, Eric J. Gonzalez, Shenli Yuan, Jason B. Ginsberg, and Sean Follmer. shapeShift: 2D Spatial Manipulation and Self-Actuation of Tabletop Shape Displays for Tangible and Haptic Interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13, Montreal QC Canada, April 2018. ACM. ISBN 978-1-4503-5620-6. doi: 10.1145/3173574.3173865. URL <https://dl.acm.org/doi/10.1145/3173574.3173865>.

Mandayam A. Srinivasan. What is Haptics? *Laboratory for Human and Machine Haptics: The Touch Lab, Massachusetts Institute of Technology*, pages 1–11, 1995. URL <http://touchlab.mit.edu>.

Detlev Stalling, Malte Westerhoff, and Hans-Christian Hege. Amira: A Highly Interactive System for Visual Data Analysis. In *Visualization Handbook*, pages 749–767. Elsevier, 2005. ISBN 978-0-12-387582-2. doi: 10.1016/B978-012387582-2/50040-X. URL <https://linkinghub.elsevier.com/retrieve/pii/B978012387582250040X>.

Soeren Strauss, Adam Runions, Brendan Lane, Dennis Eschweiler, Namrata Bajpai, Nicola Trozzi, Anne-Lise Routier-Kierzkowska, Saiko Yoshida, Sylvia Rodrigues da Silveira, Athul Vijayan, Rachele Tofanelli, Mateusz Majda, Emillie Echevin, Constance Le Gloanec, Hana Bertrand-Rakusova, Milad Adibi, Kay Schneitz, George Bassel, Daniel Kierzkowski, Johannes Stegmaier, Miltos Tsiantis, and Richard S. Smith. MorphoGraphX 2.0: Providing context for biological image analysis with positional information. Preprint, *Developmental Biology*, August 2021. URL <http://biorxiv.org/lookup/doi/10.1101/2021.08.12.456042>.

Ryo Suzuki, Eyal Ofek, Mike Sinclair, Daneil Leithinger, and Mar Gonzalez-Franco. HapticBots: Distributed Encountered-type Haptics for VR with Multiple Shape-changing Mobile Robots. *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 1269–1281, October 2021. doi: 10.1145/3472749.3474821. URL <http://arxiv.org/abs/2108.10829>.

Robert J. Teather and Wolfgang Stuerzlinger. Guidelines for 3D positioning techniques. In *Proceedings of the 2007 Conference on Future Play - Future Play '07*, page 61, Toronto, Canada, 2007. ACM Press. ISBN 978-1-59593-943-2. doi: 10.1145/1328202.1328214. URL <http://portal.acm.org/citation.cfm?doid=1328202.1328214>.

The Qt Company. Qt, n.d. URL <https://www.qt.io/>.

Thermo Fisher Scientific. Avizo Software. URL <https://www.thermofisher.com/ca/en/home/electron-microscopy/products/software-em-3d-vis/avizo-software.html>.

Will Usher, Pavol Klacansky, Frederick Federer, Peer-Timo Bremer, Aaron Knoll, Jeff Yarch, Alessandra Angelucci, and Valerio Pascucci. A Virtual Reality Visualization Tool for Neuron Tracing. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):994–1003, January 2018. ISSN 1077-2626. doi: 10.1109/TVCG.2017.2744079. URL <http://ieeexplore.ieee.org/document/8017617/>.

Pieter Verboven, Els Herremans, Lukas Helfen, Quang T. Ho, Metadel Abera, Tilo Baumbach, Martine Wevers, and Bart M. Nicolai. Synchrotron X-ray computed laminography of the three-dimensional anatomy of tomato leaves. *The Plant Journal*, 81(1):169–182, January 2015. ISSN 09607412. doi: 10.1111/tpj.12701. URL <https://onlinelibrary.wiley.com/doi/10.1111/tpj.12701>.

Erik Vidholm, Sven Nilsson, and Ingela Nyström. Fast and Robust Semi-automatic Liver Segmentation with Haptic Interaction. In Rasmus Larsen, Mads Nielsen, Jon Sporring, David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, and Gerhard Weikum, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006*, volume 4191, pages 774–781. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-44727-6 978-3-540-44728-3. doi: 10.1007/11866763\_95. URL [http://link.springer.com/10.1007/11866763\\_95](http://link.springer.com/10.1007/11866763_95).

L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, June 1991. ISSN 01628828. doi: 10.1109/34.87344. URL <http://ieeexplore.ieee.org/document/87344/>.

Yong Wan, Hideo Otsuna, Holly A. Holman, Brig Bagley, Masayoshi Ito, A. Kelsey Lewis, Mary Colasanto, Gabrielle Kardon, Kei Ito, and Charles Hansen. FluoRen-

der: Joint freehand segmentation and visualization for many-channel fluorescence data analysis. *BMC Bioinformatics*, 18(1):280, December 2017. ISSN 1471-2105. doi: 10.1186/s12859-017-1694-9. URL <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1694-9>.

Yimin Wang, Qi Li, Lijuan Liu, Zhi Zhou, Zongcai Ruan, Lingsheng Kong, Yaoyao Li, Yun Wang, Ning Zhong, Renjie Chai, Xiangfeng Luo, Yike Guo, Michael Hawrylycz, Qingming Luo, Zhongze Gu, Wei Xie, Hongkui Zeng, and Hanchuan Peng. TeraVR empowers precise reconstruction of complete 3-D neuronal morphology in the whole brain. *Nature Communications*, 10(1):3474, December 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-11443-y. URL <http://www.nature.com/articles/s41467-019-11443-y>.

Robert H Webb. Confocal optical microscopy. *Reports on Progress in Physics*, 59:427–471, 1996. URL <https://www.semanticscholar.org/paper/Confocal-optical-microscopy-Webb/6fc8dcfb066ec31f979fba64cbf06c919219e491>.

Lei Wei, Zoran Najdovski, Wael Abdelrahman, Saeid Nahavandi, and Harrison Weisinger. Augmented optometry training simulator with multi-point haptics. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2991–2997, Seoul, Korea (South), October 2012. IEEE. ISBN 978-1-4673-1714-6 978-1-4673-1713-9 978-1-4673-1712-2. doi: 10.1109/ICSMC.2012.6378250. URL <http://ieeexplore.ieee.org/document/6378250/>.

Eric W. Weisstein. Sphere Point Picking, n.d. URL <https://mathworld.wolfram.com/>.

Philip J. Withers, Charles Bouman, Simone Carmignato, Veerle Cnudde, David Grimaldi, Charlotte K. Hagen, Eric Maire, Marena Manley, Anton Du Plessis, and Stuart R. Stock. X-ray computed tomography. *Nature Reviews Methods Primers*, 1(1):18, December 2021. ISSN 2662-8449. doi: 10.1038/s43586-021-00015-4. URL <http://www.nature.com/articles/s43586-021-00015-4>.

- Pingjun Xia. New advances for haptic rendering: State of the art. *The Visual Computer*, 34(2):271–287, February 2018. ISSN 0178-2789, 1432-2315. doi: 10.1007/s00371-016-1324-y. URL <http://link.springer.com/10.1007/s00371-016-1324-y>.
- Hang Xiao and Hanchuan Peng. APP2: Automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree. *Bioinformatics*, 29(11):1448–1454, June 2013. ISSN 1460-2059, 1367-4803. doi: 10.1093/bioinformatics/btt170. URL <https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btt170>.
- Jian Yang, Ming Hao, Xiaoyang Liu, Zhijiang Wan, Ning Zhong, and Hanchuan Peng. FMST: An Automatic Neuron Tracing Method Based on Fast Marching and Minimum Spanning Tree. *Neuroinformatics*, 17(2):185–196, April 2019. ISSN 1539-2791, 1559-0089. doi: 10.1007/s12021-018-9392-y. URL <http://link.springer.com/10.1007/s12021-018-9392-y>.
- Yong Wan, H. Otsuna, Chi-Bin Chien, and C. Hansen. An interactive visualization tool for multi-channel confocal microscopy data in neurobiology research. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1489–1496, November 2009. ISSN 1077-2626. doi: 10.1109/TVCG.2009.118. URL <http://ieeexplore.ieee.org/document/5290765/>.
- Teng Zhang, Mikolaj Cieslak, Andrew Owens, Feng Wang, Suvi K. Broholm, Teemu H. Teeri, Paula Elomaa, and Przemyslaw Prusinkiewicz. Phyllotactic patterning of gerbera flower heads. *Proceedings of the National Academy of Sciences*, 118(13):e2016304118, March 2021. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2016304118. URL <http://www.pnas.org/lookup/doi/10.1073/pnas.2016304118>.
- Feng Zhao and Xianghua Xie. An Overview on Interactive Medical Image Segmentation. *Annals of the BMVA*, 2013(7):23, 2013.

# Appendix A

## User Interface Windows

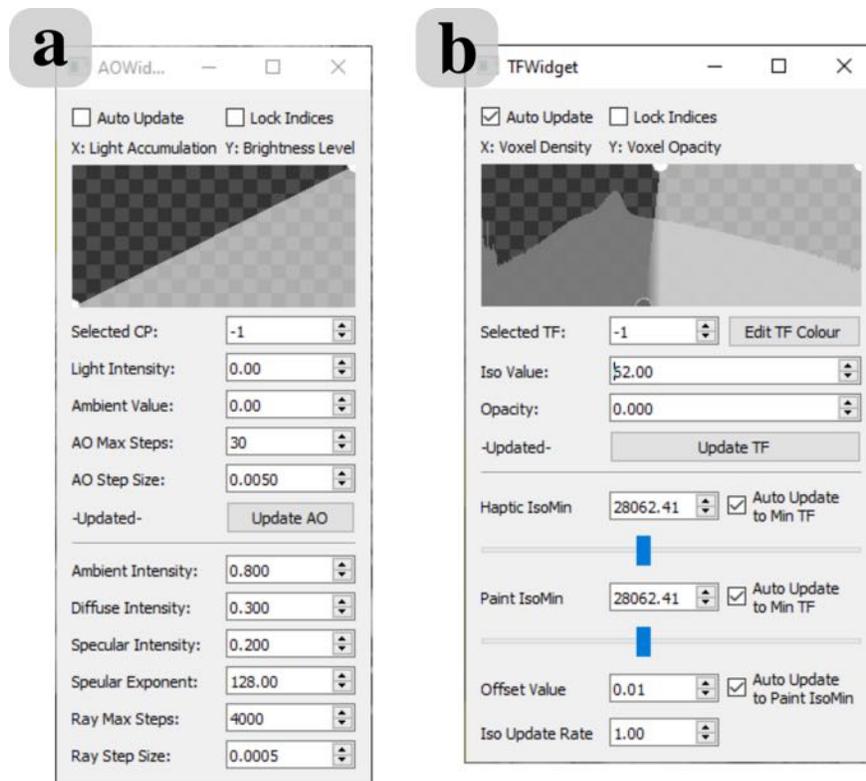


Figure A.1: (a) Ambient Occlusion Editor adjusts the settings for the ambient occlusion and lighting. (b) Transfer Function Editor adjusts the opacity and colour of the volume at different intensity values, haptics and painting threshold, and offset value for density painting.

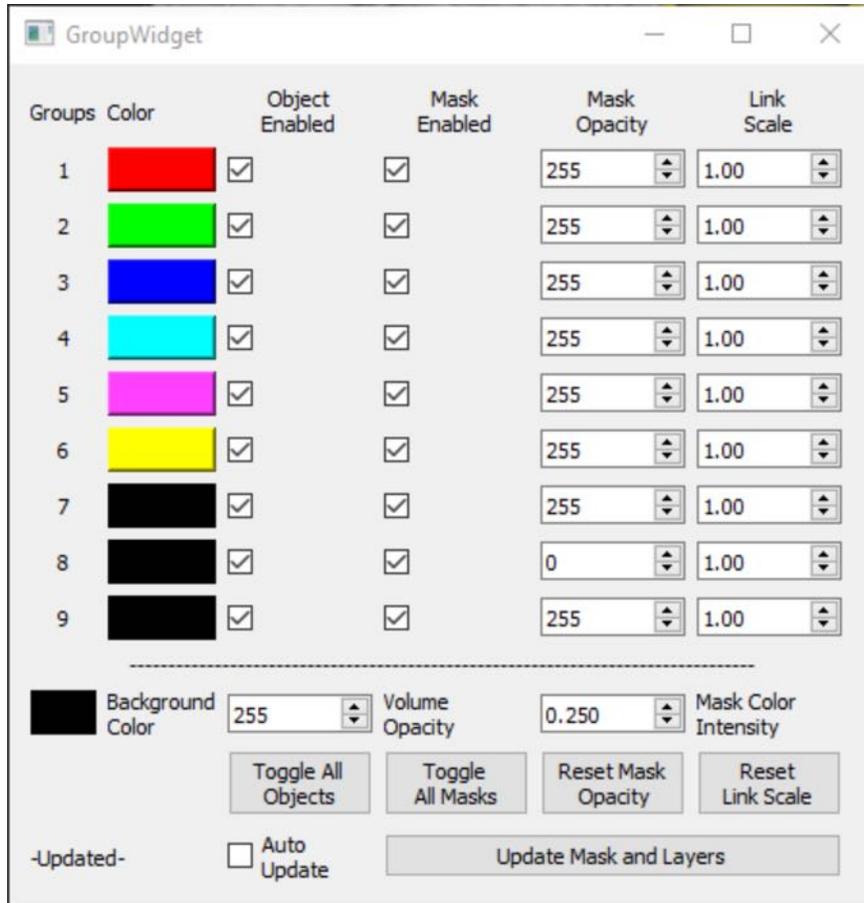


Figure A.2: Group Settings Widget edits the various colour groups. It is used to control the colour and opacity of annotation. It can also be used to adjust the opacity of the volume, and the colour of the background.

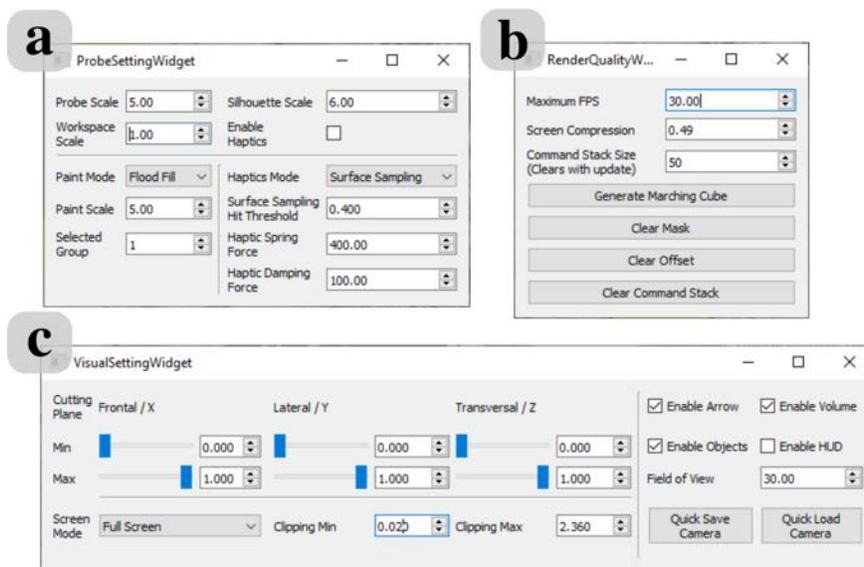


Figure A.3: (a) Probe Settings Widget adjusts the appearance of the probe, workspace scale, painting settings and haptic settings. (b) Render Quality Settings Widget adjusts the performance of the program. (c) Visualization Settings Widget adjusts the position of cutting and clipping planes, the viewing mode, and other visual assistances.





## Painting

1. Was painting easy to learn and understand?

Not at all 1 ————— 2 ————— 3 ————— 4 ————— 5 Easy to learn

2. Was it easy to start using haptic feedback to assist painting?

Not easy 1 ————— 2 ————— 3 ————— 4 ————— 5 Very easy

3. Was the haptic feedback useful for painting?

Not useful 1 ————— 2 ————— 3 ————— 4 ————— 5 Very useful

Check all the modes that you have used for painting:

### *Haptic Modes*

- No Haptics
- Surface Sampling
- Object Following
- Proximity Following

### *Viewing Modes*

- Overview Mode
- Stereoscopic Mode
- Full View Mode

What did you find easy during painting?

What did you find challenging during painting?

3

Figure B.3: Page 3.

## Tracing

1. Was tracing easy to learn and understand?

Not at all 1 ————— 2 ————— 3 ————— 4 ————— 5 Easy to learn

2. Was it easy to start using haptic feedback to assist tracing?

Not easy 1 ————— 2 ————— 3 ————— 4 ————— 5 Very easy

3. Was the haptic feedback useful for tracing?

Not useful 1 ————— 2 ————— 3 ————— 4 ————— 5 Very useful

Check all the modes that you have used for tracing:

*Haptic Modes*

- No Haptics
- Surface Sampling
- Object Following
- Proximity Following

*Viewing Modes*

- Overview Mode
- Stereoscopic Mode
- Full View Mode

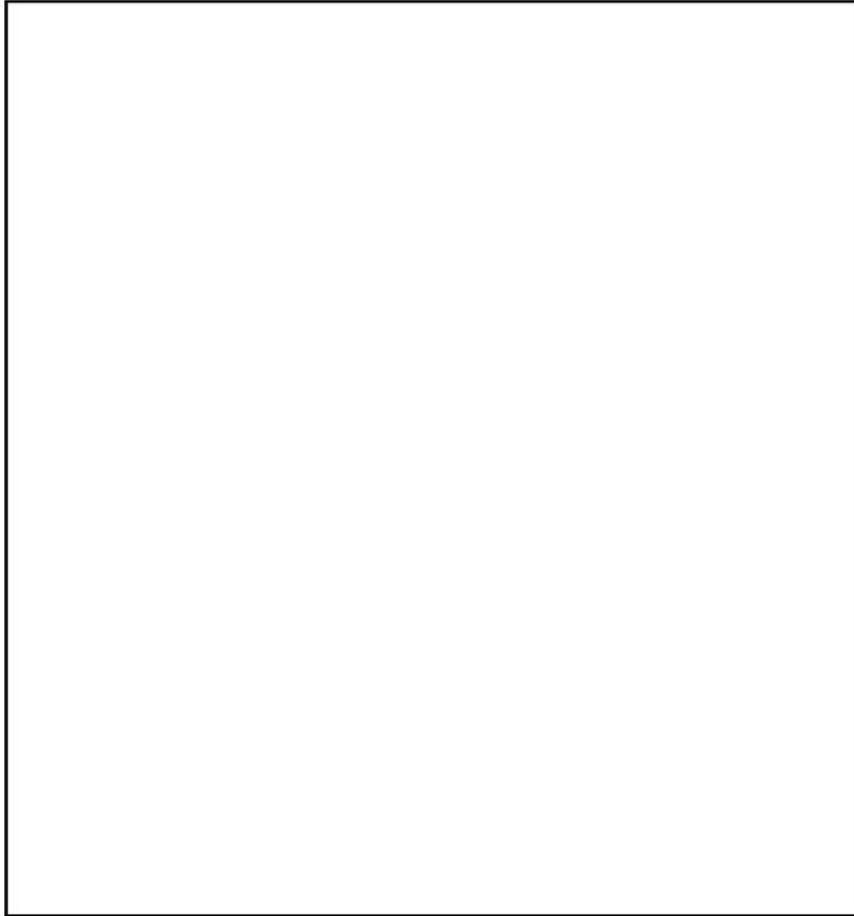
What did you find easy during tracing?

What did you find challenging during tracing?

4

Figure B.4: Page 4.

*Additional Comments*

A large, empty rectangular box with a thin black border, intended for additional comments or feedback.

*Thank you for participating!*

# Appendix C

## Keyboard Shortcuts

Shortcut	Function	Modifier 1
<b>ANNOTATION</b>		
Ctrl + Z	Undo	
Ctrl + Shift + Z	Redo	
N	Add Node	
M	Move Node	
,	Add Internode between Nearby and Selected Node	
.	Select Nearby Node/Internode	
/	Deselect Selected Node/Internode	
Delete	Delete Selected Node/Internode	
[0-9]	Change Colour Group	
Ctrl + [0-9]	Change Colour Group of Selected Node/Internode	
End	Cycle Painting Modes	
]	Increase Painting Size	
[	Decrease Painting Size	
<b>INTERACTION</b>		
Space	Toggle Haptic Assistance	
Home	Cycle Haptic Modes	
PageUp	Increase Probe Size	
PageDown	Decrease Probe Size	

Figure C.1: Page 1.

Shortcut	Function	Modifier 1
<b>NAVIGATION</b>		
Alt + [W,S,A,D,Q,E]	Rotate View	Shift - Slow Rate
[W,S,A,D,Q,E]	Translate View	Shift - Slow Rate
R	Increase Blend Rate	Shift - Slow Rate
Ctrl + R	Decrease Blend Rate	Shift - Slow Rate
T	Increase Blend Size	Shift - Slow Rate
Ctrl + T	Decrease Blend Size	Shift - Slow Rate
[Y,U,I]	Increase Near Cutting Planes	Shift - Slow Rate
Ctrl + [Y,U,I]	Decrease Near Cutting Planes	Shift - Slow Rate
[H,J,K]	Decrease Far Cutting Planes	Shift - Slow Rate
Ctrl + [H,J,K]	Increase Far Cutting Planes	Shift - Slow Rate
F7	Increase Workspace Scale	
Ctrl + F7	Decrease Workspace Scale	
<b>VISUALIZATION</b>		
P	Save Screenshot	
[-,=]	Change Size of Internodes in Selected Colour Group (No node selected)	Shift - Slow Rate
[-,=]	Change Size of Selected Node	Shift - Slow Rate
Alt + [-,=]	Change Size of All Nodes in Selected Colour Group (No node selected)	Shift - Slow Rate
C	Increase Near Clipping Plane	Shift - Slow Rate
Ctrl + C	Decrease Near Clipping Plane	Shift - Slow Rate
V	Decrease Far Clipping Plane	Shift - Slow Rate
Ctrl + V	Increase Far Clipping Plane	Shift - Slow Rate
F1	Toggle Arrow	
F2	Toggle Volumetric Rendering	
F3	Toggle Mesh Objects	
F4	Toggle Thin Shell Iso-Surface	
F8	Update Ambient Occlusion Volume	
F9	Overview Viewing Mode	
F10	Full Screen Viewing Mode	
F11	Stereoscopic Viewing Mode	

Figure C.2: Page 2.

# Appendix D

## Using SHVR

When SHVR application is started, a default scene will be loaded using default settings that are stored in the “settings” folder.

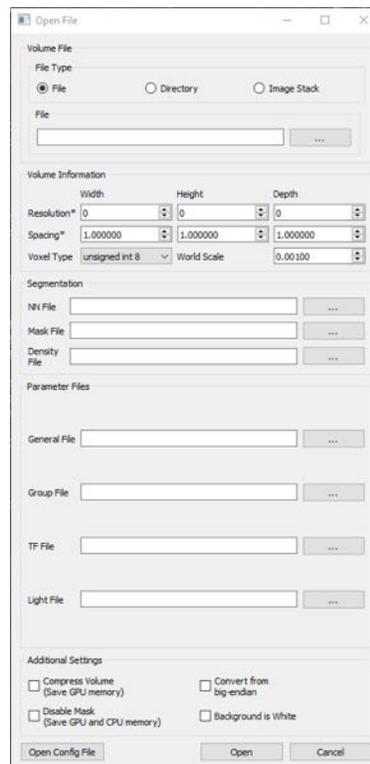


Figure D.1: Load File Widget.

Loading a volume requires the file paths to volumetric image data, settings files and annotation data, and the resolution and spacing of the volumetric image data. In addition, additional settings, such as limiting GPU memory consumption or reversing the intensity value of voxels, need to be chosen. These items can be filled in by opening the “Load File” widget, dragging and dropping files of interest on the application window, or opening a configuration file containing all the necessary information to load a volume (Figure D.1). If a file path is left blank, then the the corresponding setting file in the “settings” folder is used instead.

For using SHVR for annotation, refer to the section “Annotation Workflow” (Section 5.1).

### **D.0.1 System Requirements**

A user needs a 6 DOF kinesthetic haptic device (e.g. The Touch<sup>TM</sup> by 3D Systems), mouse, keyboard, and computer with Windows operating system. SHVR was used on a computer with 48GB of RAM, AMD Ryzen<sup>TM</sup> 3600 CPU, and AMD Radeon<sup>TM</sup> 5700 XT graphics card with 8GB of graphics memory. A stereoscopic display is optional to use the stereoscopic viewing mode.

### **D.0.2 File Formats**

SHVR uses several file formats to load into SHVR, and to save output. Volumetric image data use RAW or TIF file formats. RAW files contain uncompressed image data without any additional information about the voxel size, volume spacing, or volume resolution. Such information needs to be stored separately, and specified when the RAW file is loaded. On the other hand, TIF files contain all the above additional information. Thus, TIF files are generally recommended for use.

Segmentation mask data is saved as a RAW file with unsigned 8-bit voxels, and the same resolution as the image data. To prevent users from confusing segmentation mask data from image data, the filename extension is changed to “.mask”.

Density painting is stored as a RAW file with signed 16-bit voxels if the volumetric image data consists of 8-bit voxels, or signed 32-bit voxels if the image data consists of 16-bit voxels. The extra voxel size is used to losslessly store density painting, and to potentially reduce the intensity value of voxels. However, density painting has only been used to increase the intensity value of voxels so far.

Setting files include transfer functions setting, colour group settings, initial configuration setting, light settings, and general settings. These store settings from one or more setting widgets into a file. Appendix A shows all the potential parameters in settings widgets.

Nodes and internodes are stored in a text file. Each node is stored as “-v *x-position, y-position, z-position, colour\_group, node\_size*”. Each internode is stored as “-l *node\_1\_index, node\_2\_index, colour\_group*”. The index of nodes is updated by creating a list of pointers to node objects, where deleted nodes are not included. An integer index value is updated to match the list index for each node.

Note that ambient occlusion data is not saved in any file. This is calculated in the GPU when image data is loaded, or when ambient occlusion is updated.

# Appendix E

## Copyright Forms

### E.1 Figure 2.2



This is a License Agreement between Philmo Gu ("User") and Copyright Clearance Center, Inc. ("CCC") on behalf of the Rightsholder identified in the order details below. The license consists of the order details, the CCC Terms and Conditions below, and any Rightsholder Terms and Conditions which are included below.

All payments must be made in full to CCC in accordance with the CCC Terms and Conditions below.

<b>Order Date</b>	22-Mar-2022	<b>Type of Use</b>	Republish in a thesis/dissertation
<b>Order License ID</b>	1202478-1	<b>Publisher</b>	BLACKWELL PUBLISHING
<b>ISSN</b>	1365-3040	<b>Portion</b>	Chart/graph/table/figure

## LICENSED CONTENT

<b>Publication Title</b>	Plant, cell & environment	<b>Publication Type</b>	e-Journal
<b>Article Title</b>	The physiological implications of primary xylem organization in two ferns.	<b>Start Page</b>	1898
		<b>End Page</b>	1911
		<b>Issue</b>	11
<b>Date</b>	12/31/1977	<b>Volume</b>	35
<b>Language</b>	English	<b>URL</b>	<a href="http://onlinelibrary.wiley.com/journal/10.1111/(ISSN)1365-3040">http://onlinelibrary.wiley.com/journal/10.1111/(ISSN)1365-3040</a>
<b>Country</b>	United Kingdom of Great Britain and Northern Ireland		
<b>Rightsholder</b>	John Wiley & Sons - Books		

## REQUEST DETAILS

<b>Portion Type</b>	Chart/graph/table/figure	<b>Distribution</b>	Worldwide
<b>Number of charts / graphs / tables / figures requested</b>	2	<b>Translation</b>	Original language of publication
<b>Format (select all that apply)</b>	Electronic	<b>Copies for the disabled?</b>	No
<b>Who will republish the content?</b>	Academic institution	<b>Minor editing privileges?</b>	No
<b>Duration of Use</b>	Life of current edition	<b>Incidental promotional use?</b>	No
<b>Lifetime Unit Quantity</b>	Up to 500,000	<b>Currency</b>	CAD
<b>Rights Requested</b>	Main product		

## NEW WORK DETAILS

<b>Title</b>	Annotation of Developing Vascular Structure of Plants using Haptic Assistance	<b>Institution name</b>	University of Calgary
		<b>Expected presentation date</b>	2022-04-30
<b>Instructor name</b>	Dr. Przemyslaw Prusinkiewicz		

## ADDITIONAL DETAILS

The requesting person /  
organization to appear  
on the license Philmo Gu

## REUSE CONTENT DETAILS

<b>Title, description or numeric reference of the portion(s)</b>	Various representations of vascular bundles.	<b>Title of the article/chapter the portion is from</b>	The physiological implications of primary xylem organization in two ferns.
<b>Editor of portion(s)</b>	BRODERSEN, CRAIG R.; ROARK, LINDSEY C.; PITTERMANN, JARMILA	<b>Author of portion(s)</b>	BRODERSEN, CRAIG R.; ROARK, LINDSEY C.; PITTERMANN, JARMILA
<b>Volume of serial or monograph</b>	35	<b>Publication date of portion</b>	2012-10-31
<b>Page or page range of portion</b>	1898-1911		

## RIGHTSHOLDER TERMS AND CONDITIONS

No right, license or interest to any trademark, trade name, service mark or other branding ("Marks") of WILEY or its licensors is granted hereunder, and you agree that you shall not assert any such right, license or interest with respect thereto. You may not alter, remove or suppress in any manner any copyright, trademark or other notices displayed by the Wiley material. This Agreement will be void if the Type of Use, Format, Circulation, or Requestor Type was misrepresented during the licensing process. In no instance may the total amount of Wiley Materials used in any Main Product, Compilation or Collective work comprise more than 5% (if figures/tables) or 15% (if full articles/chapters) of the (entirety of the) Main Product, Compilation or Collective Work. Some titles may be available under an Open Access license. It is the Licensors' responsibility to identify the type of Open Access license on which the requested material was published, and comply fully with the terms of that license for the type of use specified Further details can be found on Wiley Online Library <http://olabout.wiley.com/WileyCDA/Section/id-410895.html>.

## CCC Terms and Conditions

1. Description of Service; Defined Terms. This Republication License enables the User to obtain licenses for republication of one or more copyrighted works as described in detail on the relevant Order Confirmation (the "Work(s)"). Copyright Clearance Center, Inc. ("CCC") grants licenses through the Service on behalf of the rightsholder identified on the Order Confirmation (the "Rightsholder"). "Republication", as used herein, generally means the inclusion of a Work, in whole or in part, in a new work or works, also as described on the Order Confirmation. "User", as used herein, means the person or entity making such republication.
2. The terms set forth in the relevant Order Confirmation, and any terms set by the Rightsholder with respect to a particular Work, govern the terms of use of Works in connection with the Service. By using the Service, the person transacting for a republication license on behalf of the User represents and warrants that he/she/it (a) has been duly authorized by the User to accept, and hereby does accept, all such terms and conditions on behalf of User, and (b) shall inform User of all such terms and conditions. In the event such person is a "freelancer" or other third party independent of User and CCC, such party shall be deemed jointly a "User" for purposes of these terms and conditions. In any event, User shall be deemed to have accepted and agreed to all such terms and conditions if User republishes the Work in any fashion.
3. Scope of License; Limitations and Obligations.
  - 3.1. All Works and all rights therein, including copyright rights, remain the sole and exclusive property of the Rightsholder. The license created by the exchange of an Order Confirmation (and/or any invoice) and payment by User of the full amount set forth on that document includes only those rights expressly set forth in the Order Confirmation and in these terms and conditions, and conveys no other rights in the Work(s) to User. All rights not expressly granted are hereby reserved.

- 3.2. General Payment Terms: You may pay by credit card or through an account with us payable at the end of the month. If you and we agree that you may establish a standing account with CCC, then the following terms apply: Remit Payment to: Copyright Clearance Center, 29118 Network Place, Chicago, IL 60673-1291. Payments Due: Invoices are payable upon their delivery to you (or upon our notice to you that they are available to you for downloading). After 30 days, outstanding amounts will be subject to a service charge of 1-1/2% per month or, if less, the maximum rate allowed by applicable law. Unless otherwise specifically set forth in the Order Confirmation or in a separate written agreement signed by CCC, invoices are due and payable on "net 30" terms. While User may exercise the rights licensed immediately upon issuance of the Order Confirmation, the license is automatically revoked and is null and void, as if it had never been issued, if complete payment for the license is not received on a timely basis either from User directly or through a payment agent, such as a credit card company.
- 3.3. Unless otherwise provided in the Order Confirmation, any grant of rights to User (i) is "one-time" (including the editions and product family specified in the license), (ii) is non-exclusive and non-transferable and (iii) is subject to any and all limitations and restrictions (such as, but not limited to, limitations on duration of use or circulation) included in the Order Confirmation or invoice and/or in these terms and conditions. Upon completion of the licensed use, User shall either secure a new permission for further use of the Work(s) or immediately cease any new use of the Work(s) and shall render inaccessible (such as by deleting or by removing or severing links or other locators) any further copies of the Work (except for copies printed on paper in accordance with this license and still in User's stock at the end of such period).
- 3.4. In the event that the material for which a republication license is sought includes third party materials (such as photographs, illustrations, graphs, inserts and similar materials) which are identified in such material as having been used by permission, User is responsible for identifying, and seeking separate licenses (under this Service or otherwise) for, any of such third party materials; without a separate license, such third party materials may not be used.
- 3.5. Use of proper copyright notice for a Work is required as a condition of any license granted under the Service. Unless otherwise provided in the Order Confirmation, a proper copyright notice will read substantially as follows: "Republished with permission of [Rightsholder's name], from [Work's title, author, volume, edition number and year of copyright]; permission conveyed through Copyright Clearance Center, Inc. " Such notice must be provided in a reasonably legible font size and must be placed either immediately adjacent to the Work as used (for example, as part of a by-line or footnote but not as a separate electronic link) or in the place where substantially all other credits or notices for the new work containing the republished Work are located. Failure to include the required notice results in loss to the Rightsholder and CCC, and the User shall be liable to pay liquidated damages for each such failure equal to twice the use fee specified in the Order Confirmation, in addition to the use fee itself and any other fees and charges specified.
- 3.6. User may only make alterations to the Work if and as expressly set forth in the Order Confirmation. No Work may be used in any way that is defamatory, violates the rights of third parties (including such third parties' rights of copyright, privacy, publicity, or other tangible or intangible property), or is otherwise illegal, sexually explicit or obscene. In addition, User may not conjoin a Work with any other material that may result in damage to the reputation of the Rightsholder. User agrees to inform CCC if it becomes aware of any infringement of any rights in a Work and to cooperate with any reasonable request of CCC or the Rightsholder in connection therewith.
4. Indemnity. User hereby indemnifies and agrees to defend the Rightsholder and CCC, and their respective employees and directors, against all claims, liability, damages, costs and expenses, including legal fees and expenses, arising out of any use of a Work beyond the scope of the rights granted herein, or any use of a Work which has been altered in any unauthorized way by User, including claims of defamation or infringement of rights of copyright, publicity, privacy or other tangible or intangible property.
5. Limitation of Liability. UNDER NO CIRCUMSTANCES WILL CCC OR THE RIGHTSHOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES (INCLUDING WITHOUT LIMITATION DAMAGES FOR LOSS OF BUSINESS PROFITS OR INFORMATION, OR FOR BUSINESS INTERRUPTION) ARISING OUT OF THE USE OR INABILITY

TO USE A WORK, EVEN IF ONE OF THEM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In any event, the total liability of the Rightsholder and CCC (including their respective employees and directors) shall not exceed the total amount actually paid by User for this license. User assumes full liability for the actions and omissions of its principals, employees, agents, affiliates, successors and assigns.

6. Limited Warranties. THE WORK(S) AND RIGHT(S) ARE PROVIDED "AS IS". CCC HAS THE RIGHT TO GRANT TO USER THE RIGHTS GRANTED IN THE ORDER CONFIRMATION DOCUMENT. CCC AND THE RIGHTSHOLDER DISCLAIM ALL OTHER WARRANTIES RELATING TO THE WORK(S) AND RIGHT(S), EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ADDITIONAL RIGHTS MAY BE REQUIRED TO USE ILLUSTRATIONS, GRAPHS, PHOTOGRAPHS, ABSTRACTS, INSERTS OR OTHER PORTIONS OF THE WORK (AS OPPOSED TO THE ENTIRE WORK) IN A MANNER CONTEMPLATED BY USER; USER UNDERSTANDS AND AGREES THAT NEITHER CCC NOR THE RIGHTSHOLDER MAY HAVE SUCH ADDITIONAL RIGHTS TO GRANT.
7. Effect of Breach. Any failure by User to pay any amount when due, or any use by User of a Work beyond the scope of the license set forth in the Order Confirmation and/or these terms and conditions, shall be a material breach of the license created by the Order Confirmation and these terms and conditions. Any breach not cured within 30 days of written notice thereof shall result in immediate termination of such license without further notice. Any unauthorized (but licensable) use of a Work that is terminated immediately upon notice thereof may be liquidated by payment of the Rightsholder's ordinary license price therefor; any unauthorized (and unlicensable) use that is not terminated immediately for any reason (including, for example, because materials containing the Work cannot reasonably be recalled) will be subject to all remedies available at law or in equity, but in no event to a payment of less than three times the Rightsholder's ordinary license price for the most closely analogous licensable use plus Rightsholder's and/or CCC's costs and expenses incurred in collecting such payment.
8. Miscellaneous.
  - 8.1. User acknowledges that CCC may, from time to time, make changes or additions to the Service or to these terms and conditions, and CCC reserves the right to send notice to the User by electronic mail or otherwise for the purposes of notifying User of such changes or additions; provided that any such changes or additions shall not apply to permissions already secured and paid for.
  - 8.2. Use of User-related information collected through the Service is governed by CCC's privacy policy, available online here:<https://marketplace.copyright.com/rs-ui-web/mp/privacy-policy>
  - 8.3. The licensing transaction described in the Order Confirmation is personal to User. Therefore, User may not assign or transfer to any other person (whether a natural person or an organization of any kind) the license created by the Order Confirmation and these terms and conditions or any rights granted hereunder; provided, however, that User may assign such license in its entirety on written notice to CCC in the event of a transfer of all or substantially all of User's rights in the new material which includes the Work(s) licensed under this Service.
  - 8.4. No amendment or waiver of any terms is binding unless set forth in writing and signed by the parties. The Rightsholder and CCC hereby object to any terms contained in any writing prepared by the User or its principals, employees, agents or affiliates and purporting to govern or otherwise relate to the licensing transaction described in the Order Confirmation, which terms are in any way inconsistent with any terms set forth in the Order Confirmation and/or in these terms and conditions or CCC's standard operating procedures, whether such writing is prepared prior to, simultaneously with or subsequent to the Order Confirmation, and whether such writing appears on a copy of the Order Confirmation or in a separate instrument.
  - 8.5. The licensing transaction described in the Order Confirmation document shall be governed by and construed under the law of the State of New York, USA, without regard to the principles thereof of conflicts of law. Any case, controversy, suit, action, or proceeding arising out of, in connection with, or related to such licensing transaction shall be brought, at CCC's sole discretion, in any federal or state court located in the County of New York, State of New York, USA, or in any federal or state court whose geographical

3/23/22, 5:23 PM

<https://marketplace.copyright.com/rs-ui-web/mp/license/617de5c7-784d-4ba1-a912-0e1e1ac2b6ac/c42ac7af-bf30-4ac0-8899-acc4ab0833b9>

jurisdiction covers the location of the Rightsholder set forth in the Order Confirmation. The parties expressly submit to the personal jurisdiction and venue of each such federal or state court. If you have any comments or questions about the Service or Copyright Clearance Center, please contact us at 978-750-8400 or send an e-mail to [support@copyright.com](mailto:support@copyright.com).

v 1.1