University of Calgary

PRISM Repository **https://prism.ucalgary.ca**

The Vault Open Theses and Dissertations

2013-06-14

Focus+Context via Snaking Paths

Packer, Jeffrey F.

Packer, J. F. (2013). Focus+Context via Snaking Paths (Master's thesis, University of Calgary, Calgary, Canada). Retrieved from https://prism.ucalgary.ca. doi:10.11575/PRISM/26200 http://hdl.handle.net/11023/755

Downloaded from PRISM Repository, University of Calgary

UNIVERSITY OF CALGARY

Focus+Context via Snaking Paths

by

Jeffrey F. Packer

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

June, 2013

© Jeffrey F. Packer 2013

Abstract

Focus+context visualizations reveal specific structures in high detail while effectively depicting its surroundings, often relying on transitions between the two areas to provide context. We present an approach to generate focus+context visualizations depicting cylindrical structures along snaking paths that enables the structures themselves to become the transitions and focal areas, simultaneously. A method to automatically create a snaking path through space by applying a path finding algorithm is presented. A 3D curve is created based on the 2D snaking path. We describe a process to deform cylindrical structures in segmented volumetric models to match the curve and provide preliminary geometric models as templates for artists to build upon. Structures are discovered using our constrained volumetric sculpting method that enables removal of occluding material while leaving them intact. We find the resulting visualizations effectively mimic a set of motivating illustrations and discuss some limitations of the automatic approach.

Table of Contents

Abstract ii

Table of Contents iii

List of Figures v

1 Introduction 1
- 1.1 Motivation . . . 3
- 1.2 Goal . . . 5
- 1.3 Problems and Challenges . . . 8
- 1.4 Methodology . . . 9
- 1.5 Contributions . . . 13
- 1.6 Thesis Overview . . . 14

2 Background 15
- 2.1 Focus+Context . . . 15
  - 2.1.1 Distortion Magnification Methods . . . 17
  - 2.1.2 Structural Highlighting Including Ghosting and Cutaways . . . 19
  - 2.1.3 Continuous Multiscale Visualizations . . . 21
- 2.2 Hamiltonian Pathfinding . . . 23
- 2.3 Volume Deformation . . . 24
- 2.4 Curve Skeletons . . . 26
- 2.5 Volumetric Sculpting . . . 27
- 2.6 Interfaces for 3D Curve Modelling . . . 29

3 Snaking Curve Generation 31
- 3.1 Problem Formulation . . . 31
- 3.2 Point Ordering . . . 37
- 3.3 3D Point Creation . . . 42
- 3.4 Final Curve Generation . . . 44

4 Creating Models for the Snaking Paths 49
- 4.1 Volume Deformation . . . 51
  - 4.1.1 Focal Structure Requirements . . . 51
  - 4.1.2 Focal Subvolume Creation . . . 52
  - 4.1.3 Curve Skeletons . . . 53
  - 4.1.4 Local Curve Frames and Mapping . . . 55
  - 4.1.5 Sweeping Deformation . . . 57
- 4.2 Multiple Levels of Focus+Context . . . 61

5 Constrained Sculpting via Point Radiation for Focus+Context **69**
5.1 A Brief Overview of Point Radiation 70
5.2 Constrained Point Radiation 71
5.3 Constrained Sculpting and Snaking Curve Focus+Context 73

6 Results and Discussion **78**
6.1 Evaluation Criteria 78
6.2 Curve Generation Results 79
6.2.1 Curve Evaluation 82
6.3 A Visualization from a Textured Snaking Curve 85
6.4 Generated Visualizations 85
6.4.1 Visualizations of Objects Created via Swept Volume 86
6.4.2 Visualizations of Acquired Volume Data 93
6.5 Limitations 98

7 Conclusion **99**
7.1 Snaking Curve Generation 100
7.2 Focus+Context Visualizations via Snaking Paths 101
7.3 Future Work 102

Bibliography **104**

List of Figures and Illustrations

1.1 An illustration motivating the work in this thesis. A human limb is cut revealing a cross section of the muscle (upper left). The structures that compose its interior are illustrated along a snaking path. This multiscale image demonstrates all structures of muscle tissue down to the microscopic level. (Illustration courtesy of Natasha Shevchenko. Used with permission.) 2

1.2 A second motivating illustration. A nerve is illustrated along a snaking path. The cell body (upper left) acts as the context while the axon (light red and blue along the curve) is the focus. The final focus of the image depicts a view of the molecular processes between nerves. (Illustration courtesy of Natasha Shevchenko. Used with permission.) 2

1.3 A photograph of a stripped coaxial cable (a) is shown next to an illustration of the layers of a coaxial cable (b). Illustrations of tubular objects similar to (b) demonstrate the typical methods used to convey the structure of such objects, where consecutive outer layers are stripped off to reveal inner layers. Contrasting (a) with (b), we observe that (b) is more clear in demonstrating the separation between components because of the distinct coloring. (Image (b)[70] is used under Creative Commons 3.0 `http://creativecommons.org/licenses/by/3.0/`.) 3

1.4 One of the motivating illustrations is labelled to show where the focal points, transitions, and largest scale in the illustration are located. 4

1.5 Region-based magnification focus+context. The circle is the context in (a) while the leaf is the context in (b). A large component of these illustrations are the gray transition areas that provide visual cues designating the origin of each magnified portion. (Image (b) courtesy of Mahmudul Hasan and Faramarz Samavati. Used with permission. Image of leaf in (b)[38] is used under the Creative Commons 2.0 `http://creativecommons.org/licenses/by/2.0/deed.en` license.) 6

1.6 The motivating illustrations with 2D bounding areas overlaid. The snaking paths sweep out the space inside the bounding area in order to depict the structures contained along the transitions as well as at the focal points. . . . 7

1.7 An overview of the flow of the entire approach used to create focus+context visualizations via snaking paths. Space is designated to confine the snaking path. Our path generation approach creates a set of points, or nodes, and uses a pathfinding algorithm to generate a path through the set of nodes. A final 3D snaking curve is generated from the 2D nodes in the path using an automatic depth assignment. Finally, a section of a structure is deformed to follow the generated snaking curve. Several resolutions of data can be used to create the multiple levels of focus+context, each containing more detail than the previous. This flow shows a final visualization involving multiple levels of focus+context of the aorta. 10

2.1 A simple multiscale focus+context where the context remains at one scale and the magnified focal points are depicted at larger scales. Image used with permission, courtesy of Mahmudul Hasan and Faramarz Samavati (original leaf image [38]). 16
2.2 A cutaway illustration demonstrating how a series of cubes layered inside of each other can be revealed. Notice that by cutting away occluding material, portions of the original outside cubes are removed, destroying part of the context. 18
2.3 Material removed due to copyright. Images showed (a) a magnification of the human aorta including red blood cells inside a human torso and (b) a magnification of the leg of a stag beetle. Non-linear projections are used to magnify specific portions while blending with the original objects to create a continuous, multiscale, focus+context visualization. Images can be found in Hsu et al. [47] (Figure 10(c) and 13(a)). 22
2.4 Free-form volume deformation using the method of Sederberg and Parry [66]. Instead of using trivariate Bezier patches, B-Spline patches were used here to create a more local deformation. Buckyball volume model from `http://lgdv.cs.fau.de/External/vollib/`. 25

3.1 Our observations of the motivating illustrations. These observations led to the requirements for the approach described in this chapter. 32
3.2 An example of (a) zero, (b) one, and (c) two iterations of the Peano space-filling curve (solid black) filling a given space (dotted red). A curve is created to follow the direction of the Peano curve in the first two iterations ((b),(c), dashed green). We consider the possibility of using a space-filling curve to automatically create our desired snaking curves but their fractal nature gives the curve too many smaller turns. 34
3.3 A path that sweeps out the height of the bounding area several times as it travels across the area. This path matches our list of requirements better than a space-filling curve such as a Peano curve. 34
3.4 Two examples of a snaking path in which the path (a) follows a sinusoid and (b) one in which the paths doubles back on itself and continues slithering forward (sideways in the shown 2D path) in a sinusoidal pattern. 35
3.5 An outline of the steps in the snaking curve generation process: (1) A bounding box is defined, (2) a grid of nodes is created based on a number of direction changes, (3) a path through all nodes is found, and (4) the nodes are automatically transformed into 3D and a curve is created using the nodes as guides in 3D. 38
3.6 The connectivity of the nodes used in finding a Hamiltonian path of minimum total turning angle. 38
3.7 A curve (a) produced with a grid size of three-by-three and a curve (b) produced with a grid size of three-by-five (second numbers are width). The additional interior nodes in the three-by-five grid affect the shape of the resulting curve but not the snaking path. 41

3.8 The effect on the path and shape of the 2D bounding area by excluding a node from the pathfinding algorithm. 41
3.9 Rays (not shown) originating from the 2D coordinates in the viewing plane are shot into the scene and intersected with view-aligned planes to find 3D positions. An interpolating curve is formed by utilizing the 3D points. 42
3.10 An approximation of the logarithmic spacing described in this section. Each tick represents the location of a node in the depth dimension. 43
3.11 An example of (a) uniform depth steps between points versus (b) logarithmic depth steps. 45
3.12 An example of (a),(d) small total depth change versus (c)(f) large total depth change. For comparison, (b)(e) a moderate total depth change is provided between the images of the small and large depth changes. The top row is created using uniform spacing while the bottom row is created using logarithmic depth spacing. 46
3.13 Control points on abrupt corners can cause (b) sharper turns, when using B-Spline curves (weights of all control points in the NURBS curve are *one*). Thus, in the final curve we enable the use of weighted control points to create a NURBS curve and assign lower weights to the control points that contribute to sharp corners (c) in order to create smoother corners. 47
3.14 A curve created (a) by weighting the control point in the upper left corner with a large weight (5) in order to create a sharp turn. Another curve (b) without the upper left control point weighted is provided for comparison. . . 47
3.15 A curve created by (a) not specifying its end tangent contrasted with a curve (b) whose end tangent is forced to the right. 48

4.1 An outline of the deformation process. 50
4.2 A generalized cylinder created with a circular cross section. The diameter of the cross section changes along the trajectory of the curve. The curve skeleton (black) of this generalized cylinder is the same curve as the trajectory used to create the generalized cylinder. 52
4.3 A comparison of a portion of a volume (a) before masking using discrete segmentation data and (b) after. The aliased appearance after the masking is due to the discrete classification from segmentation, denoting whether or not a voxel is within the segmented region. 52
4.4 A demonstration the curve skeleton found by MAT: (a) a volume and its (b) curve skeleton. 54
4.5 Jagged curve skeleton (top) compared to a curve skeleton smoothed by cubic b-spline reverse subdivision (bottom). The smoothed curve has three levels of reverse subdivision applied. 55
4.6 Parallel transport frames of a curve. 56
4.7 An example of the sweeping deformation applied to (a) a structure containing a turn and (b) the resulting deformation to create an axis-aligned volume. . 57

4.8 A rectangular hexahedron (a) is deformed to follow an arbitrary curve (b) using the sweeping deformation approach for volumes described in this section. The deformed volume appears thinner than the original because the length of the structure has increased. 59
4.9 Deforming a simple volume with a semi-cylindrical nature. The (a) original volume, shown along with (b) its extruded curve skeleton, is (c) deformed to be axis-aligned. The errors in the deformation help us to determine what sort of artifacts may occur in the deformation of acquired volume data because of our assumption of cylindrical structure. 60
4.10 Irregularities resulting from deforming a sharp, twisted curve into a straight path. 61
4.11 A deformation to demonstrate the loss of surface detail when structures are deformed. (b) The area of lost detail is circled. This example uses the FFD method of Sederberg et al. [66] rather than the sweeping approach presented in this section. However, the sweeping approach suffers from the same limitation when the deformation is large. 62
4.12 Colored blocks denote the curve skeleton. Locating the next point along the curve skeleton along the border between low and high resolution areas requires finding the nearest point in the curve skeleton in the higher resolution. In this case, (2,1) in the low resolution matches up with (5,4) in the high resolution which can not be located without a search in the neighborhood of the points in the high resolution that fill (2,1) in the low resolution (those are (4,2), (4,3), (5,2), and (5,3) in the high resolution). 63
4.13 A result containing three sections along the curve. Each section receives its volume data from a consecutively higher resolution of the original volume data. Because the original volume data was already fairly low resolution, the differences are minor. 65
4.14 A visualization with a curve generated with our approach. The volume model of the torso is very low resolution while the aorta, found by sculpting and deformed over first half of the curve, is a higher resolution than the volume model in the background. The last portion of the curve is a surface model created along the generated curve. The surface model was exported and edited to be partially cut and contain red blood cells to illustrate the possibility of quick edits for depicting certain structures in higher detail or larger scale than the original acquired volume data allows. 66

5.1 A 3D footprint of size $2R$ used for spreading energy radially using a Gaussian kernel of radius R. 70
5.2 Rays are cast from a screen-space tool mask that defines a drilling tool towards the surface of the volume. Points, radiated in their 3D footprints, are spawned in layers according to the view direction. 71

5.3 A simplifed 2D illustration of voxels that must be discarded, shaded red, during constrained point radiation. Footprints covering sets of voxels, designated by the circles, and a curve that denotes the surface of a forbidden region that should not be affected by the current interaction are shown. The magenta shaded voxels represent areas of that should not be sculpted because they lie within the forbidden region. Notice that some footprints cover areas that should be sculpted and areas that should not. 72
5.4 A smoothly sculpted dataset using constrained point radiation on multiple layers of a layered cube. 74
5.5 Sculpting the skull down to the segmented white matter. Dataset courtesy of BrainWeb: Simulated Brain Database (http://www.bic.mni.mcgill.ca/brainweb/). 74
5.6 Aliased appearance due to sampling an integer label volume with nearest neighbor interpolation. The solution is to use an appropriate rendering method for integer labelled volumes. 75
5.7 The interaction process we went through to generate illustrations in this thesis using constrained point radiation. In this example, the aorta is chosen as the structure of interest and sculpted towards using constrained point radiation. A bounding box, the start, and the end point for the snaking curve generation is specified and a final visualization is automatically created. 76

6.1 An example of a 3D curve created using our snaking curve generation approach. The curve is split into four sections each of which terminates with a larger area that is oriented towards the viewpoint to create an area of focus. The curve is generated with the input on the left and a three-by-four grid of nodes for the path finding (three direction changes). 80
6.2 A demonstration of how the orientation of the focal points change as they move along the curve. They can not lie anywhere on the curve because the orientation may not be toward the viewpoint. However, in this curve that is only within the labelled area. 80
6.3 The curve shown here was generated using a bounding box wider than it is tall (magenta). Compared to Figure 6.1, this curve sweeps out a wider path, which changes the orientation of each of the focal areas. 81
6.4 Curves generated with wide bounding boxes contain many areas of poor orientation. 81
6.5 An example of how a curve can be customized by moving the 2D nodes. The original, automatically generated curve (a) has its control points moved in 2D to produce the resulting slanted curve (b). Both curves use the same depth assignment when generating the final 3D curve. 82
6.6 The positions of the original 16 control points (green) are moved in the direction of the arrows (red) to a final position (magenta) in order to produce the curve shown in Figure 6.5. 83
6.7 A particularly smooth, snaking curve produced using our approach. 83
6.8 A final visualization may require insets and labels in specific locations, thus, the generated curve may not be desirable for the purposes of final visualization layout. 83

6.9 A fully-featured visualization showing a CT scan of a human head as the context, a segmented portion of an angiography of a cranial vessel as the focus, and the original dataset shown before and after a cut-away. The path of the cranial vessel towards the viewpoint is generated using the same configuration as in Figure 6.8 with some nodes excluded from the pathfinding. (See Figure 6.10.) 84
6.10 The difference between the automatic pathfinding when the lower right node is included (left) versus excluded (right). The curve generated before excluding the node is shown in Figure 6.7 while a visualization using a curve generated by excluding the lower right node is shown in Figure 6.9. 84
6.11 A visualization of the structure of muscle tissue created in likeness to the motivating image in Figure 1.1. The 3D model along the snaking curve was textured in Blender [37] after it was generated using our approach. The rendering is overlaid onto the muscle from the motivating image in Figure 1.1 for context. 86
6.12 A visualization of a coaxial cable produced using our process. The original visualization with an image of a real coaxial cable is shown (a). A rotated view of the visulization is shown in (b), without the image of a real coaxial cable. The same process to generate Figure 6.5 were used to generate this result. 87
6.13 Another visualization of a coaxial cable created with our process, this time with more layers shown. Our model is overlaid on an image of a real coaxial cable (top-right). The volume data for this visualization was created via swept volume from an image taken by the author. 88
6.14 The image of the coaxial cable cross section used to create the volume data for the coaxial cable visualizations. We see the outer, tan segment at the rear of Figure 6.13, followed by the black segment, grey segment, and inner copper wire as we travel along the path towards the closest portion. Image created by the author. 88
6.15 Visualization from a magnified, stained cross section of the stem of a Hypericum perforatum (St. John's Wort) overlaid on an image of a cut stem of the same plant. The snaking curve for this visualization was produced using the setup shown in Figure 3.4(b). 89
6.16 This visualization uses the same curve path as in Figure 6.15 but using uniformly spaced control points (in depth) instead of logarithmically spaced points. While the uniform spacing mitigates the sharp turn near the last focal point, the furthest focal point is no longer oriented towards the viewpoint. 90
6.17 The original stained cross sections of a Hypericum perforatum (St. John's Wort) stem that were swept to create the volumes used in Figure 6.15. Images used under the Creative Commons 3.0 `http://creativecommons.org/licenses/by/3.0`, courtesy of RolfDieterMueller. 90

6.18 A visualization created from a cross section of a fossil of a prehistoric tree (Hermanophyton). Cross section images can be found in Figure 6.20. The goal of this visualization is to bring the inner black portion of the fossil into focus in front of the viewpoint while providing a view of the context (original fossil) in the lower left corner. 91
6.19 An example of how a bad choice for the total depth change of the snaking curve can affect the final model. The curve does not appear to be travelling along a nice snaking path through space and instead appears contorted. The focal cross sections were moved slightly so that they did not lie on the twisted portion of the generated curve. 92
6.20 The original fossil cross sections used to create the volume used in Figure 6.18. Images courtesy of Mike Viney and the Virtual Petrified Wood Museum [74]. Used with permission. 92
6.21 One level of focus+context created using the stented aorta dataset. The end of the aorta, which is closest to the heart in the original dataset, is deformed along the snaking curve causing it to stretch out. However, most of its shape is retained allowing a viewer to get a sense of the shape of the aorta at its position in the original context. 95
6.22 A close-up visualization of the original aorta dataset. Part of the dataset has been sculpted to remove the occluding material around the aorta so that this model can be used as the context for Figure 6.21. 95
6.23 The volume model of the torso is of a lower resolution in this visualization, while the aorta is deformed over first half of the curve and is a higher resolution than the volume model in the background. A surface model is created along the last portion of the snaking curve, closest to the viewpoint. The red blood cells were added afterwards and the entire visualization was rendered in Blender [37]. 96
6.24 A cranial vessel from the angiography dataset is brought into focus along a snaking curve containing only one switchback. The path of the original generated curve has been altered via our methods to accomodate labels and insets in order to produce the final visualization (See Section 6.2). 97

Chapter 1

Introduction

Illustrations help convey information in a way that succinctly clarifies the form or components of an object. Illustrative visualization, a field of computer graphics, is concerned with generating an image that helps clarify components or form. For example, an exploded view of a piece of furniture can help clarify the location of all its components, such as interior supports or fasteners, in relation to the whole. Such a view is common in assembly instructions. Illustrative visualization also aims to re-create techniques used by artists as well as discovering and evaluating new methods of visualization that are best at portraying certain types of information. Much of illustrative visualization deals with non-photorealistic rendering techniques instead of trying to recreate realism; the goal is to communicate information in the best way possible and to not create exact images. Prominent examples of illustrative visualization can be found in technical illustrations such as the example of furniture assembly instructions and medical illustration such as those found in educational anatomy material.

A large subfield of illustrative visualization is that of focus+context. Focus+context visualizations reveal a specific structure, known as the focus, in high detail while clearly depicting the origin or location of the focus within the larger picture, known as the the context. Some artistic medical illustrations depict structures using focus+context. Illustrations such as those in Figure 1.1 and Figure 1.2 present a structure and transitions between its layers over a snaking curve. The snaking curve acts as an illustrative transition that helps to emphasize the length of the structure and the form of its outer layers. For example, human muscle tissue may be drawn along a snaking path to demonstrate that the tissue as a whole is made up of long strands while depicting the structure of its outer layers as in Figure 1.1.

In this thesis we present an algorithmic method of generating such focus+context vi-

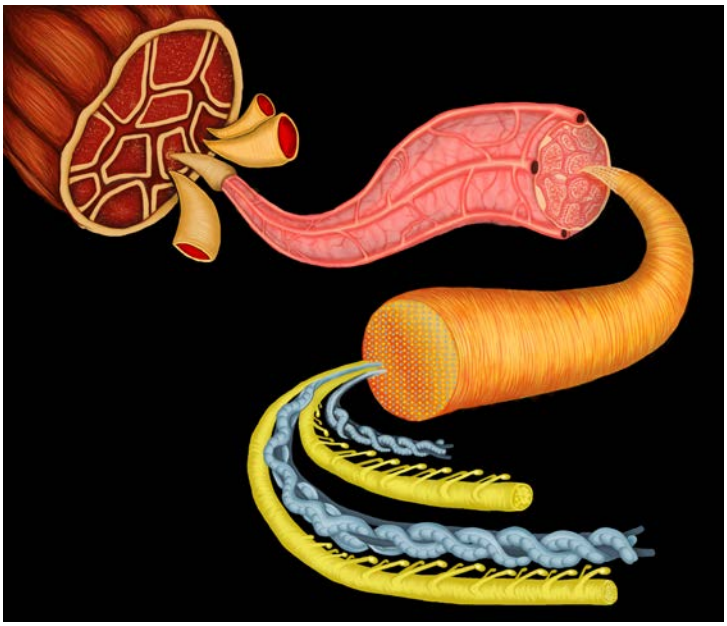
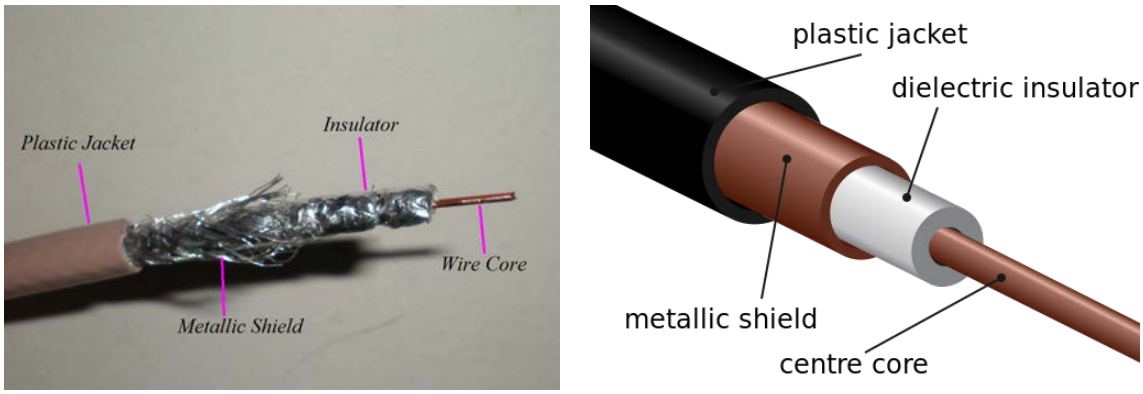


Figure 1.1: An illustration motivating the work in this thesis. A human limb is cut revealing a cross section of the muscle (upper left). The structures that compose its interior are illustrated along a snaking path. This multiscale image demonstrates all structures of muscle tissue down to the microscopic level. (Illustration courtesy of Natasha Shevchenko. Used with permission.)

Figure 1.2: A second motivating illustration. A nerve is illustrated along a snaking path. The cell body (upper left) acts as the context while the axon (light red and blue along the curve) is the focus. The final focus of the image depicts a view of the molecular processes between nerves. (Illustration courtesy of Natasha Shevchenko. Used with permission.)



(a) (b)

Figure 1.3: A photograph of a stripped coaxial cable (a) is shown next to an illustration of the layers of a coaxial cable (b). Illustrations of tubular objects similar to (b) demonstrate the typical methods used to convey the structure of such objects, where consecutive outer layers are stripped off to reveal inner layers. Contrasting (a) with (b), we observe that (b) is more clear in demonstrating the separation between components because of the distinct coloring. (Image (b)[70] is used under Creative Commons 3.0 `http://creativecommons.org/licenses/by/3.0/`.)

sualizations along snaking paths. We focus on generating snaking paths through three-dimensional (3D) space and deforming models along the paths to create a visualization.

1.1 Motivation

The work in this thesis is motivated by illustrations similar to those shown in Figure 1.1 and Figure 1.2. The particular feature of these illustrations that motivates our work is the visualization of a structure of interest along a snaking path with several levels of focus+context. These types of images are more imaginative renditions of images similar to that in Figure 1.3(b) as opposed to purely technical or educational in nature. Illustrations similar to Figure 1.3(b) are excellent at clarifying the different layers but are poor at depicting information about the length of the overall object or exteriors of each layer.

The motivating illustrations clearly depict the layers of the shown structures or objects. However, they also aid in demonstrating the length of the objects. This is due to the fact that they are designed to illustrate tubular structures with several layers. Thus, these illustrations

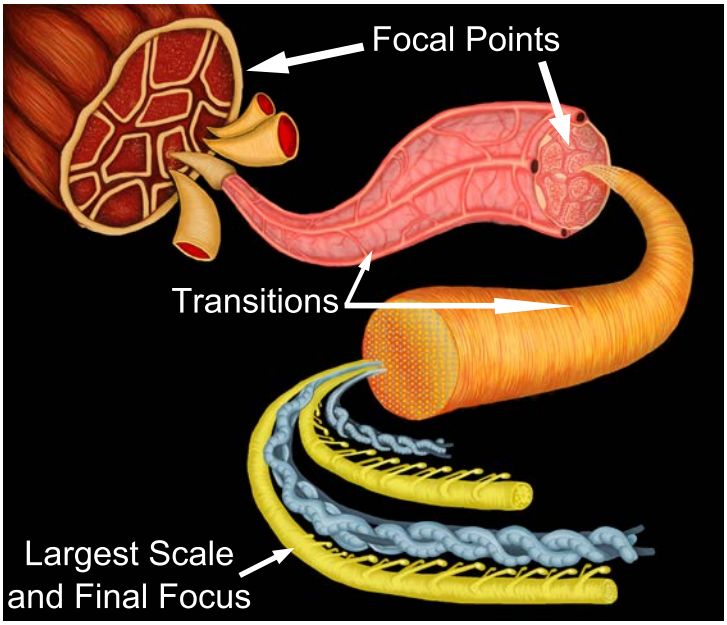


Figure 1.4: One of the motivating illustrations is labelled to show where the focal points, transitions, and largest scale in the illustration are located.

could be found in areas including medical illustration, as shown by the motivating images, and technical illustration. Imagine an illustration of a coaxial cable that is similar to that in Figure 1.3(b) but along a snaking path to demonstrate its length or to create extra interest. We imagine similar illustrations found when demonstrating the structure of pipes or tubular biological structures, even if the structures are normally rigid like a tree trunk.

We describe the motivating illustrations in order to reveal the motivation in mimicking their methods to create computer generated visualizations. The motivating images depict a multiscale visualization of a cylindrical structure with several levels of focus+context. A focus+context effect is created in Figure 1.1 with the initial context shown in the background while each successive scale is brought into focus along a snaking path towards the viewpoint. Several focal points are presented along the path while the path itself acts as space for the transitions between the focal points. The location of the focal points and transitions is shown in Figure 1.4. The transitions in Figure 1.4 provide more information than transitions in traditional region-based magnification methods such as those demonstrated in Figure 1.5. Not only do they provide visual cues of the origin of the magnified focal points but they also

demonstrate both the length and form of the structure, which is beneficial when working in 3D. It is not possible to illustrate the images in Figure 1.5 in such a fashion, as they are two-dimensional (2D). However, there are many 3D anatomical or industrial tubular structures that may be naturally illustrated in a manner similar to the muscle fiber in Figure 1.1 such as cables and pipes. These objects would benefit from the visual continuity and additional information conveyed along the illustrated transitions involving their interior structures, especially when multiple levels of focus are required.

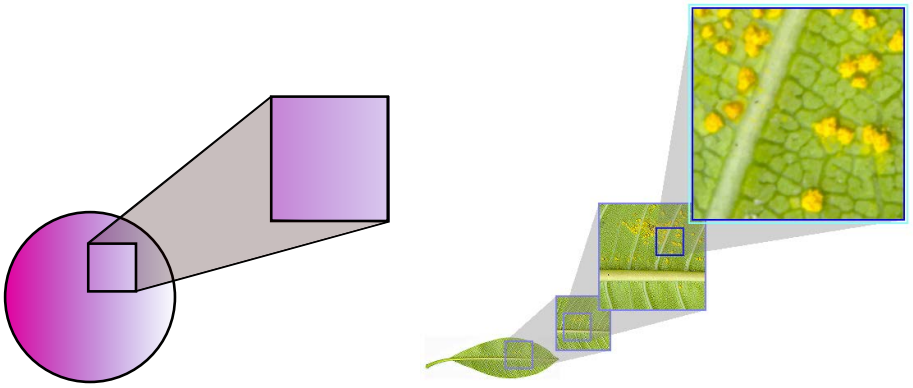
The multiscale nature of the motivating illustrations also demonstrates an important application of creating such a visualization approach. Figure 1.1 depicts several resolutions of data ranging from the size of a human arm (centimetres) down to the microscopic structures creating the muscle (micrometres). These smallest structures are four orders of magnitude smaller than the largest structure. A dataset containing every detail on this smallest order of magnitude would be far too large to fully load into memory[1]. By splitting the visualization into sections along the path we enable the ability to load only the required portions of each volume. Some of these volumes may be captured at a very small scale while the others may be captured at a large scale. This enables our approach to be extended to extreme multiscale visualizations.

1.2 Goal

The goal of the work in this thesis is to provide an approach for creating computer generated focus+context visualizations similar to those found in the motivating images (Figures 1.1 and 1.2) from acquired volumetric data[2]. The purpose of algorithmically generating the visualizations is two-fold: (1) to mitigate the amount of time spent modelling a 3D curve through space for the snaking path and (2) to produce a volume model from acquired data

[1] Hundreds of petabytes.

[2] We refer to volumetric data as volume data or volume models which are discrete arrays of sampled values, called voxels.



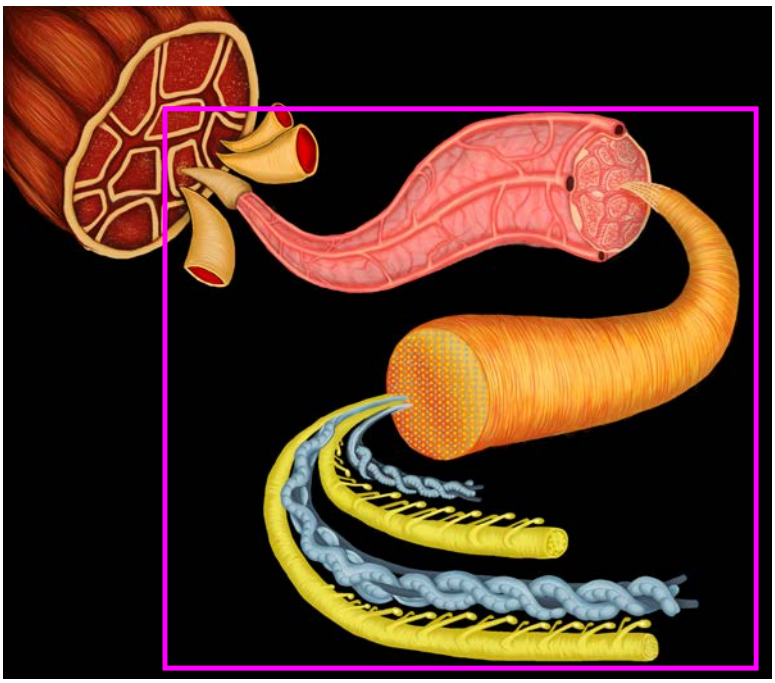
(a) (b)

Figure 1.5: Region-based magnification focus+context. The circle is the context in (a) while the leaf is the context in (b). A large component of these illustrations are the gray transition areas that provide visual cues designating the origin of each magnified portion. (Image (b) courtesy of Mahmudul Hasan and Faramarz Samavati. Used with permission. Image of leaf in (b)[38] is used under the Creative Commons 2.0 `http://creativecommons.org/licenses/by/2.0/deed.en` license.)

along the snaking path to enable visualization of data captured by volumetric scanners.

We choose to utilize volume data so that we can create visualizations of the desired form directly from acquired volume data, such as clinical data. Utilizing existing acquired volume data along with an algorithmic approach for generating the desired illustration enables those with little modelling experience to create the visualizations. Volume data is typically difficult to interact with and deform, thus, our approach to the deformation of volume data should be automatic. Volume data, however, can lack the level of resolution necessary to differentiate between very small structures. Thus, we consider an additional goal of generating a preliminary geometric model along the snaking path that provides a template to be further enhanced using external 3D modelling software.

There are several observations that we make about these types of illustrations that provide targeted goals for the work in this thesis. The first goal is to algorithmically create a 3D snaking path whose 2D projection is loosely bound in a 2D area. In the motivating images, we observe that the path sweeps back and forth across a 2D area, producing its snaking behaviour. Overlaying imagined bounding areas over the motivating images, as in Figure 1.6,



(a)

(b)

Figure 1.6: The motivating illustrations with 2D bounding areas overlaid. The snaking paths sweep out the space inside the bounding area in order to depict the structures contained along the transitions as well as at the focal points.

helps to demonstrate that the paths sweep out the space inside the bounding area. This helps depict the structures along the transitions as well as at the focal points.

The second goal is to enable the creation of a model along the snaking path with multiple levels of focus+context, preferably from the original acquired volume data. We observe that in the motivating illustrations several scales are presented along a path. Each scale has its own designated section of the path and these sections end with an enlarged cap, slightly oriented towards the viewpoint to create an area of focus. When combined, multiple levels of focus+context are created.

1.3 Problems and Challenges

There are several issues that we must solve in order to create our approach. These include: (1) the location of structures of interest from volume models, (2) the creation of a snaking path in 2D, (3) the generation of a 3D curve whose projection into 2D matches the snaking curve, and (4) the creation of a model along the snaking curve from the original volume model.

We require a method of interactively exploring a volume model so that structures for the final visualization can be discovered and a visualization of the context can be created. For example, if a major artery in a volume model of a human torso is to be the focus, then the artery must be located in the original model to create a context for the focus.

We separate the creation of a snaking path in 2D from the generation of a 3D curve because the extra dimension adds additional complexity. If the snaking path were created directly in 3D it would be necessary to consider the shape of the curve and how it travels through the third dimension simultaneously. Even with the separation of the two problems, the additional third dimension may still be assigned arbitrarily, making it difficult to automatically create a 3D curve from 2D points.

An overview of how we solve each of these problems is given in the next section (Sec-

tion 1.4).

1.4 Methodology

Our general approach is to algorithmically generate all components for a final visualization based on a few constraints. We begin by manually defining a bounding area in which a snaking curve is then automatically created. A region of a structure of interest is chosen from the original acquired volume to be deformed along the snaking curve. If a multiscale image is to be created then it is necessary to have volume data for each scale. The deformation is automatically created by matching the shape of the structure to the shape of the snaking curve. The flow of the process used in this thesis is shown in Figure 1.7.

To locate the structure of interest we use sculpting methods because sculpting enables an exploratory discovery experience. Sculpting operations include chiseling, drilling, and peeling. These operations enable us to remove occluding material to help locate structures to be used in the visualizations. One problem with these volume sculpting operations is that they lack the ability to selectively sculpt structures. We create a constrained sculpting approach that prevents the sculpting from affecting a specified structure. The constrained sculpting is based on a technique called point radiation, presented by Chen et al. [19].

The process of generating the snaking curve begins by defining a bounding box into which the final visualization will fit. This bounding box defines the projected space in which the final space-filling, snaking path will be located. A method of automatically finding a snaking path that sweeps the majority of the space with a snaking shape is required. The flow diagram given in Figure 1.7 demonstrates that our approach involves creating a set of points, or nodes, within the space and applying a pathfinding algorithm through these nodes. However, not just any path through the nodes is a desirable path. Observing the motivating illustration we see that the shape or curvature of the turns are what make the snaking curves desirable. Thus, we define curves by their total turning angle.

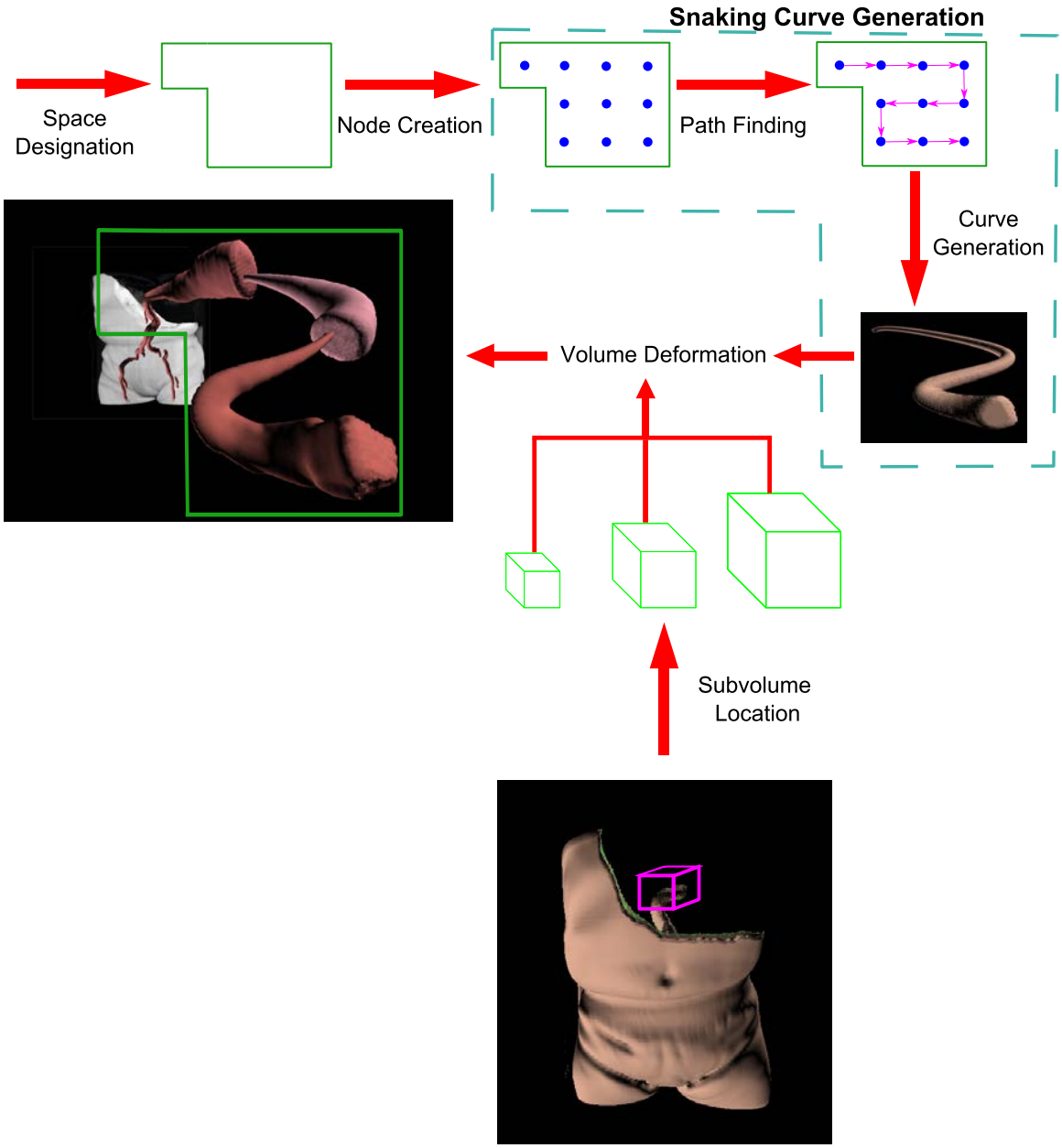


Figure 1.7: An overview of the flow of the entire approach used to create focus+context visualizations via snaking paths. Space is designated to confine the snaking path. Our path generation approach creates a set of points, or nodes, and uses a pathfinding algorithm to generate a path through the set of nodes. A final 3D snaking curve is generated from the 2D nodes in the path using an automatic depth assignment. Finally, a section of a structure is deformed to follow the generated snaking curve. Several resolutions of data can be used to create the multiple levels of focus+context, each containing more detail than the previous. This flow shows a final visualization involving multiple levels of focus+context of the aorta.

Our pathfinding approach seeks to minimize the total turning angle of the path through all of the 2D nodes, using specified start and end nodes. This minimization has the effect of forcing the curve to sweep out as much vertical or horizontal space in the bounding area as possible before turning. We find that we can apply the solution to the angular-metric travelling salesman problem [1] to find a path with the desired properties. The angular-metric travelling salesman problem finds a path that passes through all nodes but with an end node equal to the start node. We wish to find a path with differing start and end nodes. The solution to this problem is presented in Chapter 3.

We take this algorithmic approach instead of directly creating or drawing a 2D curve[3] because we think of the final visualizations in our work as being user-guided, computer-generated visualizations where the user-guidance is in the specification of the constraints for the path finding. Allowing a user to directly create the 2D curve, however, is a valid approach if more control is required. Due to the fact that we are focusing on replicating the snaking property of the curves found in the motivating illustrations, we prefer this algorithmic approach and enable finer control over the modelling afterwards. More control is enabled through interactions with the 2D nodes such as modifying their location or excluding them from the original path finding.

A 3D curve must be created once a 2D path is found. Each 2D node has a position on the screen and this position must be transformed into a 3D position, thereby creating a 3D path from the original 2D path. Enabling this transformation requires a method for assigning a depth to the position of each 2D node. However, the additional dimension creates an ambiguity because there are an infinite number of ways to assign depths to a set of 2D positions if no other constraints exist. Since the goal is to provide a curve travelling towards a viewpoint through 3D space it is possible to constrain the depth assignment such that consecutive points are spaced along the depth-dimension in a strictly descreasing

[3]Either through sketching or existing control point-based methods such as Bezier curves or B-Spline curves.

manner. That is, the depths assigned to the points range from a specified maximum and strictly descrease to a specified minimum. This ensures the curve always travels towards the viewpoint. Without experimentation there is no ideal method for automatically assigning depth to the points in such a way that creates a desirable curve. We experiment with two different approaches based on varying the difference in depth between consecutive points: uniform depth-spacing and logarithmic depth-spacing.

The resulting 3D points are used as control points for a curve modelling scheme. Control points for curve modelling schemes are points that are used to evaluate a function that defines the curve. We wish for the final curve to be as customizable as possible, both during automatic creation and further manual editing. This led to the choice of representing the curve as a Non-Uniform Rational B-spline (NURBS) due to the ability to weight control points [58]. NURBS allow for better control of the shape of the curve, especially around corners. Once a curve is created it is split into a specified number of sections, denoting the number of focal points or scales.

A specified section of a structure from the original model is now deformed to match the shape of the snaking curve. Curve skeletons, sometimes used in shape matching [25], are used to help perform the deformation. A curve skeleton is an abstract representation of the shape of a structure, similar to if the human skeleton were represented by a set of lines. The deformation is completed by mapping the curve skeleton of the structure onto the shape of the snaking curve. The original volume is deformed using the resulting mapping.

Each section of the snaking curve will depict a different scale. As such, we represent each scale as different resolutions of volume data because that is the most natural way to achieve higher detail or a larger scale with a discrete set of values. The first scale, furthest from the viewpoint, deforms a low resolution version of the volume containing the structure onto the snaking curve. As the scales increase, so does the required resolution.

Finally, to render the deformed data it must first be resampled onto a regular grid because

the sample values will no longer be equally spaced. This resampling is necessary in order to take advantage of efficient rendering of volume data via texture mapping. After resampling, the resulting model can be rendered using direct volume rendering or a geometric model can be created through isosurface extraction.

If the highest resolution of volume data lacks a desired level of detail for a higher scaled portion of the curve, then it is possible to use our approach to create a geometric model as a template to build upon. We enable this template creation by modelling a deformed cylinder along the appropriate portion of the curve.

1.5 Contributions

This thesis provides a flexible approach for generating focus+context visualizations and models that present focal structures along snaking paths. The paths the structures follow enable the structures themselves to become the transitions between the focus and context. The paths help to provide clear visual continuity between multiple focal points or scales, ultimately enabling our approach to be easily extended to be extremely multiscale.

A breakdown of several smaller contributions or steps that are presented in this work as the approach for generating such visualizations are as follows:

- Adapt an optimization algorithm to automatically find a 2D snaking path through a specified area.
- Provide an approach for creating a 3D curve whose 2D projection closely matches the 2D snaking path.
- Create models for the visualizations using acquired volume data, via a shape matching deformation. Geometric models may also be generated to ensure maximum flexibility in the approach.

- An adaptation of the point radiation method of Chen et al. [19] to create a constrained sculpting technique. The constrained sculpting technique enables easy removal of occluding material, which aids in discovering structures of interest for the focus+context visualizations.

1.6 Thesis Overview

The layout of this thesis is as follows. Chapter 2 provides background information and describes methods from several areas that are pertitinent to the approaches taken in this thesis. Chapter 3 describes our approach for generating 3D snaking curves. Chapter 4 presents our process for extracting a particular structure from segmented volumetric data and deforming a specified portion of it in order to create a model along the generated 3D snaking curve. Chapter 5 presents a method for the constrained sculpting of volumes based on the point radiation technique. Chapter 6 presents, discusses, and evaluates our results. Finally, Chapter 7 describes our conclusions and future work.

Chapter 2

Background

The main categorization of the work in this thesis is focus+context for which background material is described in Section 2.1. However, our work utilizes several other areas in computer graphics and computer science for pieces of our approach. In Section 2.2 we describe finding Hamiltonian paths. These are used for methods of automatically finding a route through a graph built to find different snaking curves based on different bounding areas and start and end points. Section 2.3 desribes work in deformation of volumetric models which helps to enable the creation of a model for the goal visualization from acquired data (for example, computed tomography (CT)). The acquired data is not originally in the shape of our generated curve for visualization and must be deformed to fit the curve. Volumetric sculpting, described in Section 2.5, is used to discover interior structures to be used in our final visualizations. Finally, interfaces for 3D curve modelling are described in Section 2.6. These interfaces influence our approach in defining, creating, and manipulating the generated curve.

2.1 Focus+Context

Focus+context involves visualizations that reveal a specific structure in high detail (focus) while demonstrating the origin or location of the structure in the larger picture (context). For example, the bones in the palm of a hand may be made visible by altering the transparency of the skin covering the bones. This technique, called ghosting, reveals that the bones are inside the hand and helps to locate the bones of the palm in reference to the rest of the hand (below the fingers, but above the wrist).

Cohen and Brodlie [24][23] categorize focus+context techniques into two major categories:

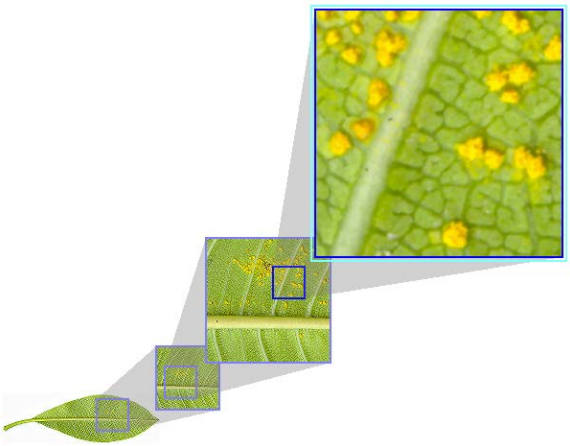


Figure 2.1: A simple multiscale focus+context where the context remains at one scale and the magnified focal points are depicted at larger scales. Image used with permission, courtesy of Mahmudul Hasan and Faramarz Samavati (original leaf image [38]).

distorted and non-distorted. Examples of distorted techniques are fish-eye magnification and some lens-based magnification methods. Examples of non-distorted techniques include region-based close-ups based on multiresolution [68], and techniques that reveal interior structures through ghosting [52] or cutaways.

Another important distinction that Cohen and Brodlie make is between those methods that are continuous and those that are discontinuous [24][23]. Region-based close-ups create discontinuity by separating the close-up from the context [68] as demonstrated in the simple example in Figure 1.5. Distortion magnification methods are usually continuous because they provide transitions between the focus and context via the distortion [59][60][75].

Some focus+context illustration techniques are multiscale; they create illustrations that contain structures shown at multiple scales. These scales usually include magnified close-ups as focal points. Figure 2.1 shows an example of a non-distortion magnification, multiscale technique. The context remains at one scale and the magnified focal points are depicted at larger scales. Because some focus+context illustrations are not multiscale but can be classified as distortion or non-distortion (for example, ghosting) we can further classify focus+context approaches into the category of multiscale.

The focus+context illustration technique we are attempting to mimic in this work can

be classified as distortion-based, but continuous. It is also multiscale because structures are illustrated from a scale demonstrating their original size down to their microscopic interiors. The illustrations highlight the long, cylindrical shape of the structures which in turn acts as a method of structural highlighting. Thus, these illustrations create continuous multiscale visualizations with the benefit of enabling a magnification of a particular structure.

We present background and previous works in distortion magnification methods (Section 2.1.1), structural highlighting methods (Section 2.1.2), and continuous multiscale visualizations (Section 2.1.3) due to our classification of the focus+context technique used in this thesis.

2.1.1 Distortion Magnification Methods

Some works in focus+context deal with the magnification of specified portions of the data by placing a lens in the region of interest while preserving the original context. The document lens [63] provides focus+context for documents through the magnification of text using a frustum-shaped lens. To break the sudden transition between magnified text and unmagnified text, magnified text is presented through the top of the frustum while the transition is presented as greeked text magnified through the angled sides of the frustum. Carpendale and Montagnese [16] manipulate points on a 2D plane in 3D space according to a lens-based model. They provide different types of transitions ranging from continuous to discontinuous depending on the model of lens used. Other lens-based methods [59][60] utilize interpolation of data around the edges of the lens to provide continuous transitions between magnified and unmagnified portions.

These methods work well within the confines of two dimensions, however, the transitions they provide may cause confusion if presented with 3D data. For example, fish-eyed views are a method used in the magic lens for volume data which distorts the original volume around the lens [75]. Simpler lenses enable the visualization of occluded areas without distorting the original volume by changing the rendering style within the area of the lens [73], however, the

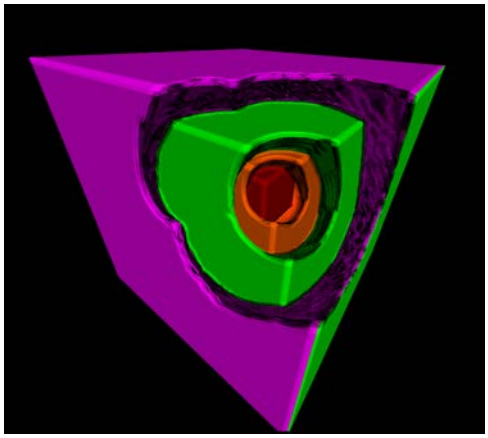


Figure 2.2: A cutaway illustration demonstrating how a series of cubes layered inside of each other can be revealed. Notice that by cutting away occluding material, portions of the original outside cubes are removed, destroying part of the context.

context of outer layers are lost within the lens. Recent 3D methods attempt to solve distortion and aid to preserve context through the use of "feature-preserving data reduction" [78] techniques and by using conformal mapping[1] [82]. Feature-preserving data reduction uses a mixture of feature specification, downsampling, and deformation to give features more emphasis through magnification and reducing the size of less important areas [78]. Zhao et al. [82] utilize conformal mapping to preserve features in the area of magnification in their lens-based magnification. The shape of the magnifier in their system is defined by a surface model upon which they create the conformal mapping.

The transition areas add information to the focus+context illustrations. A problem with these magnification methods is that the transition areas are often very distorted. Also, some of the magnification methods, such as fish-eye lenses, would be very difficult to modify to portray multiple levels of focus or magnification as the distortions would compound. Illustrating a structure along a snaking path, as in this thesis, has the benefit of mitigating distortion while enabling multiple levels of clear focus+context.

2.1.2 Structural Highlighting Including Ghosting and Cutaways

Highlighting important structures in volume visualization or illustration is an effective method of directing the attention of a viewer. Several methods can be used to highlight important structures inside of some original context. Removing occluding material on the exterior of a model to reveal its interior is one method used to highlight interior structures. Illustrations produced in this manner are called cutaways. Cutaway illustrations, however, completely remove occluding material, and thus, may remove part of the context. For example, Figure 2.2 reveals a small cube inside of several layers of cubes by cutting away the occluding material. Ghosting is also a method of visualizing the interior of a model, however, instead of completely removing occluding material the transparency of the exterior is varied, thus, reducing the removal of portions of the context.

Ghosting and cutaways are, by themselves, only somewhat useful. The original context can still be lost (cutaways and ghosting) or remain but become less clear (ghosting) due to transparency changes. Techniques that mitigate these kind of losses usually involve ghosting so that none of the context is removed. Instead, these techniques usually render the context in a different style or technique and provide a smooth transition for the ghosting. For example, Bruckner et al. [12] present a context-preserving volume rendering process that utilizes ghosting and several contrast enhancements to shift focus to interior structures. This context-preserving volume rendering process builds on the idea of two-level volume rendering [44] where a focus+context effect is created by rendering the exterior of a volume using different style than the interior.

Several importance driven methods to ghosting enable the visualization of several layers at once using a feature-based importance [52][15]. Transparency or the amount of shading is modulated based on the importance of structures. Transparency is modulated according to several different models, including view-based and curvature-based [52] and shading is

[1] A conformal mapping is a mapping that preserves angles.

modulated such that less important structures receive less shading from a chosen shading method [15].

Examples of contrast enhancements include contour enhancements [32], as well as depth and focus cues, orientation cues, distance fading, and sketch lines, which are all presented by Ebert and Rheingans [35]. These methods can be used to render different portions of ghosting or cutaway illustrations in different styles in order to highlight an area of focus.

Volumeshop, created by Bruckner et al. [11], provides support for ghosting, cutaways, and selecting and extracting subvolumes that can then be rendered to another part of the screen. Their tools enable users to intuitively create illustrations from volumetric data by providing a means of exploring a volume and extracting certain structures for further visualization. However, there is no visual continuity between the extracted structures and the original context when subvolumes are extracted and moved.

While ghosting and cutaways are not directly related to the work in this thesis they provide examples of structural highlighting. They demonstrate a class of illustrations that depict structures within the confines of the original volume visualization without moving the structures. However, each of these methods can be used to further enhance the focal sections of the snaking visualizations that are the work of this thesis.

Both ghosting and cutaways attempt to solve the problem of occlusion when attempting to view interior structures. They suffer, however, from the necessity of having to remove some of the occluding material. Exploded views highlight a structure of interest on the interior without removing material. Instead, they cut the context into smaller pieces and move these pieces such that they no longer occlude the structure of interest. Bruckner et al. [13] provide a force-based approach to move pieces of occluding material away from the focal structure. While exploded views are not directly related to our goal, they provide an excellent example of structural highlighting with a discontinuous focus+context technique that contrasts continuous techniques.

Correa and Silver [31] highlight the focal structure with an animation along its path or location. Concentrating on structures that have a considerable length, they visualize the direction of flow through a structure by highlighting portions of the structure in succession in an animation. This animation, however, is not the result of a deformation or change in position, and is instead only a change in color which leads to their formulation of a motion-controlled transfer function. The animation along the path of the structure guides focus along its length.

Our work, modelled after the motivating illustrations, brings a focal structure towards the viewer, away from the context, while maintaining a clear sense of its origin. This creates a very clear highlighting of the structure as it becomes the majority of the illustration. Illustrating a focal structure along a snaking path also serves to guide focus along the length of the structure as the path is followed, without explicit highlighting or style changes.

2.1.3 Continuous Multiscale Visualizations

The distortion magnification methods to focus+context such as those described in Section 2.1.1 provide multiscale visualization. However, some of them are discontinuous and all of them enable visualization from only a single vantage point at a time. That is, there are many parts that remain occluded but can be revealed through 3D navigation and applying the magnification again.

Hsu et al. [47] present a continuous multiscale, distortion focus+context visualization that enables viewing a scene from multiple camera placements in a single image. This enables objects to be viewed from different angles, sides, or scales, within a single image. Their method is based on non-linear projections and involves shooting non-linearly bent rays from several cameras into a scene. Each camera captures a particular scale and viewpoint. These are blended so that a single image, containing a continuous multiscale visualization, is created.

Figure 2.3 demonstrates results from the work of Hsu et al. [47]. Figure 2.3(a) shows the

(a) (b)

Figure 2.3: Material removed due to copyright. Images showed (a) a magnification of the human aorta including red blood cells inside a human torso and (b) a magnification of the leg of a stag beetle. Non-linear projections are used to magnify specific portions while blending with the original objects to create a continuous, multiscale, focus+context visualization. Images can be found in Hsu et al. [47] (Figure 10(c) and 13(a)).

interior of a human aorta with red blood cells found inside as the focus and a body in the background as the context. An overall view of a stag beetle with a magnification of one of its hind legs, is shown in Figure 2.3(b). The two views are seamlessly blended together so that the resulting image is continuous. If the scene of the aorta were viewed without using their method from the same vantage point it would be impossible to see the interior of the aorta.

Their last visualization, of the aorta, creates an effect very similar to the goal of this work. Their images contain large distortions due to the blending of several different viewpoints and these distortions can sometimes be extreme or confusing. Even with the distortions, their method could be used to visualize a snaking path by strategically placing multiple cameras and it could be interesting to see how such an image would appear. Such distortions may create a very artistic effect. However, it would be very tedious to place all cameras and wait for the final ray-traced image before correcting mistakes.

The greatest difference between the approach of Hsu et al. and the approach in this thesis is that Hsu et al. focus on blending views from multiple cameras, while our work focuses on deforming structures to fit a specific shape suitable to match the motivating illustrations. As such, their approach can only produce images while our approach can produce models that can be further manipulated.

2.2 Hamiltonian Pathfinding

Typical search and pathfinding algorithms such as the A* algorithm [43], an improved version of Dijkstra's algorithm [34] involve finding the shortest path through a graph[2] with weighted or unweighted edges. However, the purpose of these algorithms is not to find a path containing every node in the graph.

Paths containing every node in a graph are called Hamiltonian paths. Finding Hamiltonian paths is often associated with the travelling salesman problem (TSP) in which case it is a cycle that is found. The TSP involves finding the shortest tour in a complete graph starting and ending at the same node. The TSP was first solved in large scale with linear programming [33][3]. The problem can also be formulated with differing start and end nodes in an incomplete graph becoming the Hamiltonian path problem. Both problems have been shown to be NP-complete [50] and can be solved using dynamic programming [9][45], branch-and-bound [57][71], or branch-and-cut methods [21].

The TSP described above uses a distance metric as the edge cost between nodes but these costs are constant. The costs remain constant whether the problem is symmetric (costs in both directions between two nodes are equal) or assymetric (cost from node A to node B is different than the cost from B to A) and are not dependent on which edge was visited prior to an edge currently in consideration. The angular-metric TSP minimizes the total turning angle in the tour [1][4]. The turning angles depend on the ordering of the edges in the path. That is, for edges (u, v) and (v, w) the turning angle is the angle of direction change between the two edges travelling from u to w through v which may be different than the turning angle from another path travelling from another node t through v to w by way of edge (t, v) [1]. Aggarwal et al. solve the angular-metric TSP using dynamic programming similar to the solution for the regular TSP [1] while Arkin et al. approximate the solution using polygonal subdivisions [4]. The angular-metric TSP is akin to arc routing routing problems such as

[2]A graph is a set of objects, called nodes or vertices, where select pairs are connected to each other by links called edges.

the postman problem, and the snowplough problem [5] that are more concerned with the paths between nodes than the distances.

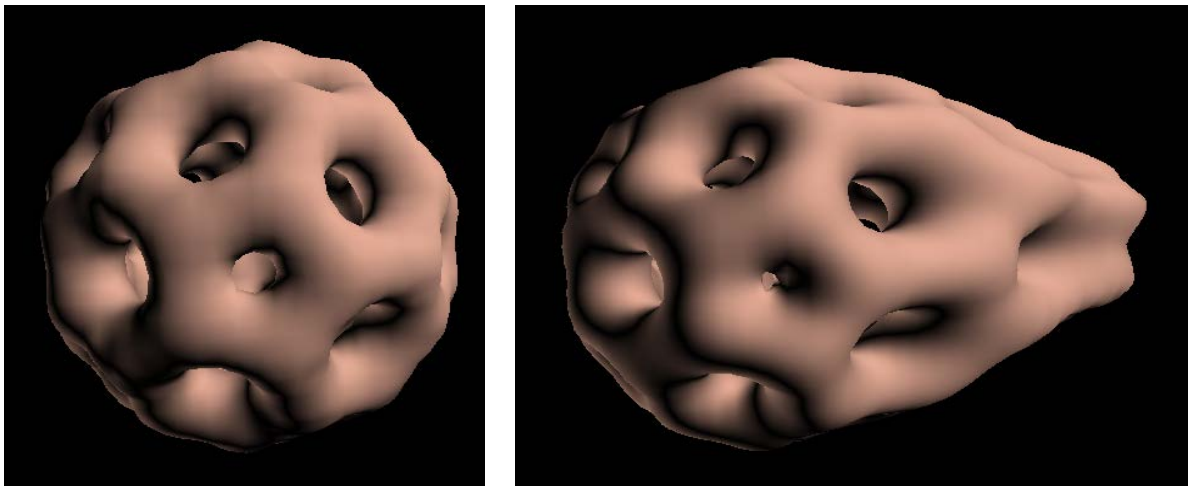
The approach presented in this thesis finds a minimum-turn cost Hamiltonian path in a set of nodes while ignoring the distance or total path length, leading to our use of the solution to the angular-metric TSP. Our implementation uses the dynamic programming approach presented by Aggarwal et al. [1] due to the fact that only a small number of nodes are used in our approach described in Section 3.2.

2.3 Volume Deformation

Volume deformation is important in this thesis because it enables the focus to be brought out of the context and deformed to follow the snaking path. The deformation of a structure from the original volumetric data enables the focal structure to maintain its likeness while following the generated snaking path for artistic purposes in the final visualization.

Deformation types include rigid and non-rigid or free form deformation (FFD). Rigid deformation is not considered because it does not allow for the elongation or flexible shape changes that would be necessary to deform structures along a snaking path. However, we acknowledge that it is superior in maintaining the original shape and structure of objects as it is routinely used in animation, such as in inverse kinematics [56].

Work on (FFD) of volumes includes deformation based on a tensor product of trivariate patches [66]. Inspired by the previous work of Sederberg and Parry [66] (example in Figure 2.4) and deformation via ray deflectors [53], which performs the deformation during ray casting by deforming the cast rays, Chen et al. [17] expand on the original FFD. They perform the deformation in the parallel environment of special purpose graphics hardware by adaptively subdividing the original ray into curve segments whose paths follow the deformation specified by cubic B-spline curves [17]. Other classes of approaches involving the use of special purpose graphics hardware includes object space approaches and texture space



(a) (b)

Figure 2.4: Free-form volume deformation using the method of Sederberg and Parry [66]. Instead of using trivariate Bezier patches, B-Spline patches were used here to create a more local deformation. Buckyball volume model from `http://lgdv.cs.fau.de/External/vollib/`.

approaches. Object space approaches deform the volume by deforming proxy geometry, usually created by subdividing a volume into subvolume geometry and then deform the resulting geometry. Rezk et al. [61] create sub-cubes whose corners and edges are deformed and the geometry of the sub-cubes is rendered and texture mapped for visualization. Texture space approaches store the deformation as displacement vectors in an additional 3D texture and use the deformation texture to compute the original location in order to look-up the original volume value [79][14].

The previous FFD methods perform global deformation which often does not aid in volume exploration. Deformations have been presented as an effective method for browsing volumetric data by McGuffin et al. [55] that introduced local deformation tools such as the 'leafer' and 'spreader'. Utilizing the paradigm of deformation for browsing, Correa et al. [28] present a local deformation method that is feature-aligned instead of axis-aligned. They provide tools to retract and to cut volumes aligned against either surfaces or segments. These operations are similar to sculpting operations but deform the area of the volume around the retracted or cut portions and do not prevent self collision. Displacement maps are generalized

to volume deformation and provide a process for high-quality modelling and rendering of discontinuous deformations such as peeling and retracting [29]. Correa et al. [30], also present a constrained deformation extending their notion of displacement mapping with constrained displacement fields that helps mimic physical constraints Their constrained deformation prevents self-collision and enables deformation of heterogenous materials which allows the movement of malleable materials in an area while rigid structures in the same area remain in place.

In this thesis we manipulate cylindrical structures only. This leads us to investigate a more specialized method of deformation. Our process for deformation is similar to the sweep-based approach for free-form deformations that maps parts of a model to different sweep surfaces and deforms each sweep surface individually [81]. In this thesis, the given cylindrical structures map directly to a cylindrical sweep surface. We can eliminate the mapping phase because the mapping is assumed to be inherent. In the deformation portion of this thesis we utilize volumetric data instead of surface models, thus, our sweep-based approach is slightly different.

2.4 Curve Skeletons

A curve skeleton of a 3D shape is a "simplified 1D representation of the medial surface" [25]. Curve skeletons are typically known just as 'skeletons' and are often used in deformations for animation. Modelling and animation software such as 3DSMax [6] and Maya [7] enable the creation and assignment of skeletons to geometric models. A model is deformed as the skeleton is transformed according to an animation, thereby, animating the original model. Other uses of curve skeletons include shape matching and flow tracking [25]. Curve skeletons are also used in focus+context techniques to highlight the shape and path of structures [31] (previously described in Section 2.1.2).

Various methods including thinning, such as medial axis thinning [54], and potential

field approaches [41][20][27] can be used to find curve skeletons. In 2D, thinning approaches erode the contour of a shape in succession until a single-pixel wide skeleton is produced. Similar methods exist for volume data since volume data consists of a stack of 2D images. Potential field methods consider the boundary of a discretized volume as point charges that generate a repulsive force field [41][20][27] and use the resulting vector field to find stream lines defining the curve skeleton. We refer the reader to Cornea et al. [26] for a full review of curve skeletons.

Curve skeletons are used in this thesis as part of the deformation process as part of the mapping from original volume data to the shape of the snaking curve.

2.5 Volumetric Sculpting

Although sculpting is a small portion of the work in this thesis it is important because it aids in the creation of visualizations using our snaking curve approach. It also provides a way to further explore the interior of the volumetric portions of the generated visualizations. A select number of works that successively improve in quality, efficiency, and ease of use are described in this section.

Sculpting for volume data was popularized as a way to shape clay or wax using a variety of tools such as "sandpaper" or a "heat gun" [40]. These tools were imprecise and highly aliased due to directly calculating tool-volume intersections with a lack of filtering. Wang and Kaufman [77] introduced a set of carving tools based on shapes of real sculpting tools. Instead of simply determining tool-volume intersections, their tools prevent aliasing by positioning a filter kernel over a sample point and splatting the points of the tool in 3D space onto neighboring sample points. The filters attenuate to a null effect at the radius of the kernel, thereby providing a mechanism of anti-aliasing.

Improvements to the sculpting metaphor include higher quality tools based on potential fields [36] and point radiation [19]. Ferley et al. [36] define a tool's shape by a potential field

and the way that potential field combines with the existing 3D model. However, because the tools they create are fully 3D they can be awkward to use with a 2D pointing device such as a mouse or stylus. Chen et al. [19] present a point-based method for an efficient, high-quality sculpting and segmentation based on energy radiating from tool-affected voxels. Point radiation defines a 3D footprint of radius R for each point using a Gaussian kernel for spreading energy radially. The 3D footprint of each point combined implicitly defines a volume whose surface is anti-aliased due to the smoothing of the Gaussian kernel. Using a painting metaphor, sets of 3D footprints are projected from the screen into the affected volume and collected in a separate sculpting volume. When rendering the original volume, the sculpting volume is sampled and if the value contained in the sculpting volume is below a certain threshold the original volume is rendered. Otherwise, the original volume is not rendered. This formulation creates the basis for a number of deletion tools such as drills, lasers, and peelers. Their toolset is created for use with a 2D pointing device such as a mouse or stylus, with emphasis on sketch-based interaction.

A problem with any of the previous methods is that they do not differentiate between voxels of different structures. If segmented volume data is available it is useful to be able to sculpt without worry of affecting a structure of interest (or, perhaps, affecting only the structure of interest). As part of this thesis, we constrain the point radiation method of Chen et al. [19] to only operate on a specfied range of labels in segmented volume data. This enables easy removal of occluding material which is necessary to locate structures of interest for the snaking visualizations we create.

A method that attempts to constrain sculpting, more for accuracy than for convenience, constrains the depth drilled based on distance gradient changes that occur across successive drilling operations [48]. Anytime the distance gradient change exceeds a threshold the drilling operation is ignored, thus, preventing drilling to an unintended depth. An advantage of this method is that it does not rely on structural information but they provide no way of

constraining the drilling to only certain structures (such as drilling bone as they propose for the use of their tool).

2.6 Interfaces for 3D Curve Modelling

The approach described in this thesis for creating a snaking curve loosely relates to interfaces for 3D curve modelling. Our approach deals with a specific curve structure, however, a select review of interfaces for 3D curve modelling is presented in this section. Interfaces for 3D curve modelling range from those found in 3D modelling software that involve the manipulation of control points to those found in sketch-based modelling.

Fully featured 3D modeling packages such as Maya [7], 3DSMax [6], and Blender [37], all provide curve modelling interfaces by enabling manipulation of control points and/or tangents for B-spline, NURBS, or Bezier curves. Accurate modelling is enabled through manipulation of the control points and/or tangents in the curve. However, modelling curves through the one-by-one placement of control points is both time consuming and may be more difficult than other methods [49].

Defining non-planar curves via sketch-based interface by drawing the curve in the current view and then defining its shadow by drawing another curve on a specified floor plane of the scene [22]. They map the shadow curve to the drawn curve (the curve to be modelled in 3D) by matching critical points, found using a definition of the directions left or right in the image plane, of the shadow curve to critical points of the curve to be modelled. A ruled surface is created from the shadow curve by extruding the shadow curve along the projection vector and the curve to be modelled is projected onto these shadow surfaces. The shadow curve and its extruded shader sufaces are used to assign depth to the 2D sketch of a curve and a 3D curve from just two sketches are created without annotations. This is an example of a two-curve method for sketch-based modelling a single curve. Another two-curve method include drawing a curve from one viewpoint and then rotating and drawing the curve from

another view [8].

Some sketch-based 3D curve modelling approaches allow piecewise sketching on specified 3D planes [8][49]. A plane is defined and then a piece of the curve drawn. A new plane is then defined and another piece of the curve is drawn. Piecing these curve segments together creates the entire 3D curve. A similar method is used to define and sketch onto an arbitrary 3D surface or using the current 3D geometry as a guide for snapping the drawn curve into the 3D scene [8][65].

Andrews et al. [2] create a sketch-based modelling system that allows the creation of 3D models from 2D vector art based on a linear optimization for inflation of the model to 3D. The inflation is constrained by the curves in the vector art. Which curves are considered constraints can be changed and they deem the constraints being considered active constraints while others inactive. These constraints can be changed at any time. They concentrate on modelling 3D shapes but 3D space curves can be automatically created from curves in the 2D vector art by marking some constraints as inactive. 3D space curves are created from inactive constraints when they are deformed according to the currently active constraints. In this way, their inflation method automatically produces 3D space curves that match the found surface according to the constraints.

Like most of the sketch-based approaches, the approach for creating modelling curves in this thesis relies on the current viewpoint, however, the resulting curve is meant to be viewed from one viewpoint in the final visualization. Thus, due to this specific curve structure we are able to automatically assign depth instead of relying on sketch-based methods such as gesture control or annotations for control over the space curve.

Chapter 3

Snaking Curve Generation

In this chapter we discuss our approach to automatically create 3D snaking curves given only a few 2D constraints. First, we describe several requirements for our process that are formed from observing the properties of the curves in the motivating illustrations to formulate the problem and our solution in Section 3.1. In Section 3.2 we present the algorithm used to solve the problem of finding a snaking path through 2D space before discussing how we transform a 2D snaking path to 3D in Section 3.3. Finally, our choice of Non-Uniform Rational B-spline (NURBS) for modelling the final curve and examples of how they are beneficial is discussed in Section 3.4.

3.1 Problem Formulation

By observing the shape and properties of curves from the motivating illustrations we have produced a number of requirements for the curves. We make several observations which are shown in Figure 3.1 that lead to a list of requirements which are given now.

Utilize a specified 2D space in a snake-like path shape The path should utilize as much of a 2D bounded space as possible, travelling over its entirety, in a snaking path. The snaking path travels through space towards the viewpoint to create the focus. This is how the length and outer portions of each inner layer is demonstrated in the motivating illustrations. However, it is not essential that the path completely use the entire space of the 2D bounding area. Notice that in Figure 3.1, the muscle illustration utilizes a large portion of the bounding space but the illustration of the nerve only runs along the length and the width of its bounding area; it does not try to fill all available space.

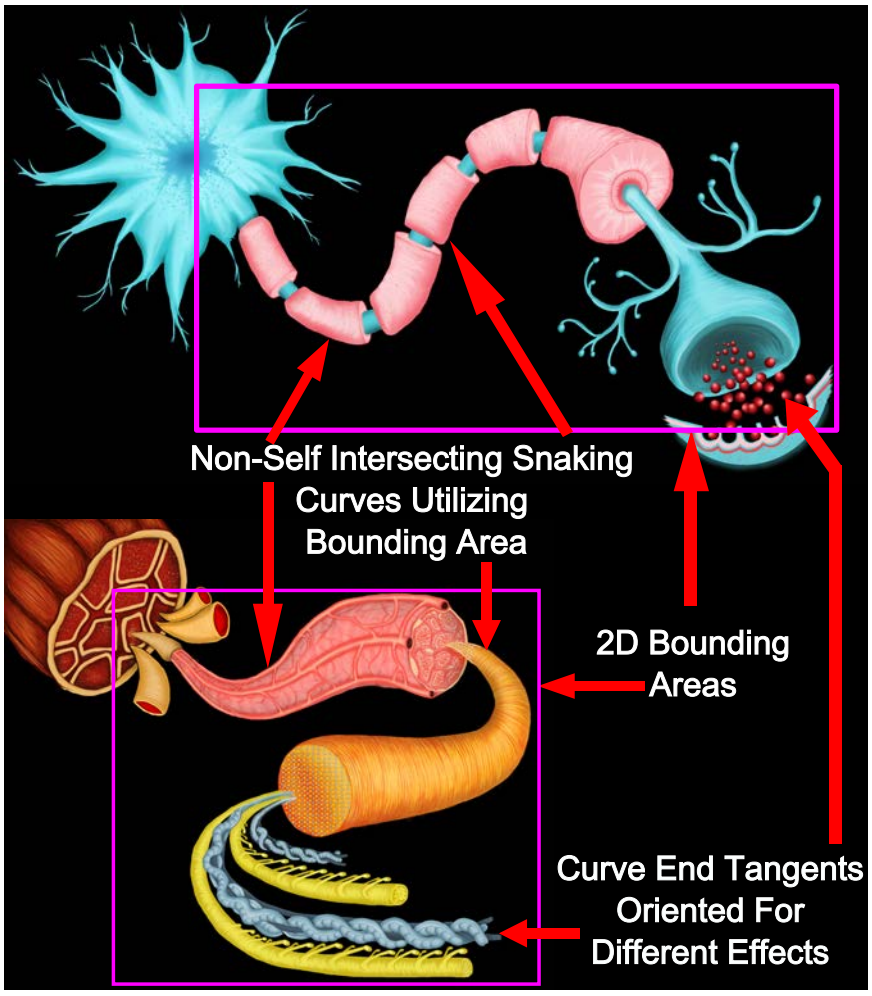


Figure 3.1: Our observations of the motivating illustrations. These observations led to the requirements for the approach described in this chapter.

No self-intersection or self-occlusion. The curve should not self-intersect. This is somewhat given by the idea that the curve should look like a snake travelling through space (as most snakes do not slither over themselves when travelling large distances). If the curve were to self-intersect then parts of the structures would be occluded at the points of intersection. Any occlusion would hide the exterior portions of the structure that are being illustrated along the path and defeat the purpose of illustrating the structure along the snaking path.

Tangents The tangent at the end of the curve should be customizable. Notice that the tangents at the end of the curves in Figure 3.1 are oriented differently to portray different effects. In the illustration with the nerve, the curve is oriented more towards the viewpoint to demonstrate the processes at the gap between nerves. However, in the muscle illustration, the curve is oriented to the right to illustrate the intertwined nature of the microscopic structures.

The process presented in this chapter is inspired by the requirement that the final curve must utilize a specified 2D space by following a snaking path. Utilizing a large portion of a 2D space could be accomplished with a space filling curve. However, space-filling curves have fractal constructions making them unsuitable for filling large areas using our list of requirements, especially the requirement for a snaking path. For example, observe the second iteration of the Peano curve in Figure 3.2(c). The overall shape of the curve in the second iteration is the same as that in the first iteration, but with several smaller turns. Further iterations contain many more smaller turns but the overall path of the curve remains the same as that in the first iteration. For the purposes of mimicking the motivating illustrations, we would rather have the path sweep out the width or height of the bounding area several times instead of containing several smaller direction changes such as the path in Figure 3.3.

Following the structure of the space-filling curves, we break the bounding space into

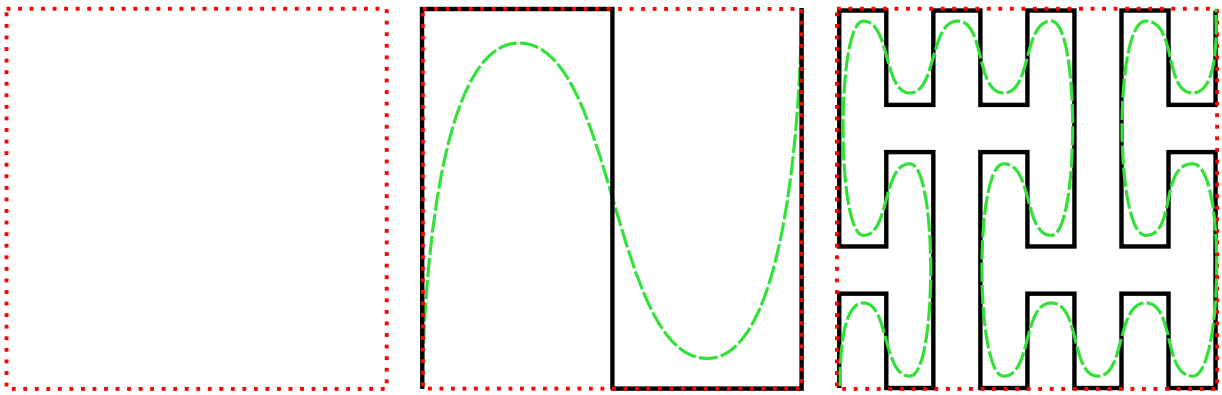


Figure 3.2: An example of (a) zero, (b) one, and (c) two iterations of the Peano space-filling curve (solid black) filling a given space (dotted red). A curve is created to follow the direction of the Peano curve in the first two iterations ((b),(c), dashed green). We consider the possibility of using a space-filling curve to automatically create our desired snaking curves but their fractal nature gives the curve too many smaller turns.

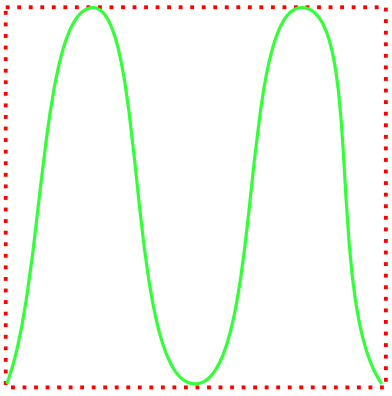
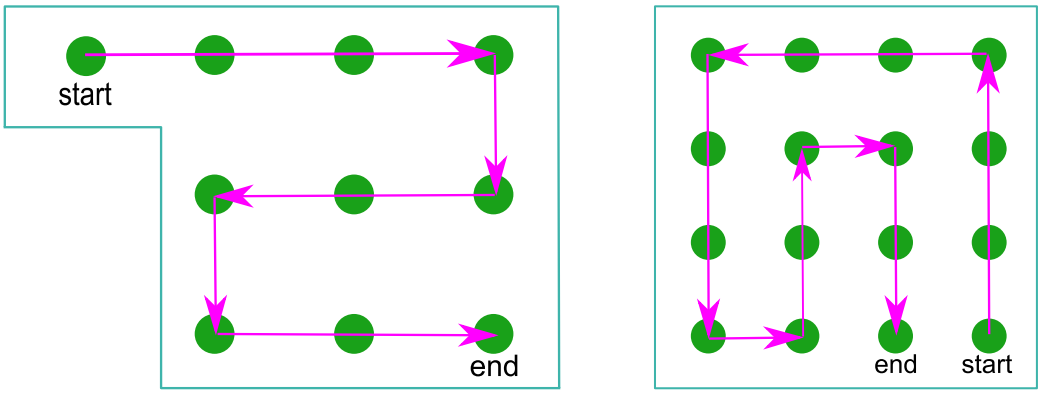


Figure 3.3: A path that sweeps out the height of the bounding area several times as it travels across the area. This path matches our list of requirements better than a space-filling curve such as a Peano curve.



(a) (b)

Figure 3.4: Two examples of a snaking path in which the path (a) follows a sinusoid and (b) one in which the paths doubles back on itself and continues slithering forward (sideways in the shown 2D path) in a sinusoidal pattern.

separate sections for which the final curve will travel between. We consider only rectangular bounding areas and build our approach such that slight changes can be made to support slightly more irregular bounding areas. The rectangular space is broken into smaller pieces by producing a grid. If a path can be found that passes through every cell in the grid and a curve created following that path then the space utilization requirement can be met. This path should have a specified start and end cell. Using specified start and end cells enables the creation of different paths using the same bounding area. This produces the need for a pathfinding method because a single path can not be preset.

The next requirement to fulfill is that of the snake-like path. For example, a simple snake-like path contains a number of locations where the curve changes direction completely and switches back on itself like a curve in the shape of an *S*. A more complicated snaking path may begin by moving up on a surface and then completely turn around on itself over an area and then continue forward in a sinusoidal fashion.

We observe the shape of these paths on grids of nodes. The nodes are located at the center of each grid cell and are connected to define the path. For example, Figure 3.4 shows the two aforementioned snaking paths. With a given start and end grid node, the final path passes through *all* nodes because we wish to utilize as much of the space as possible. A path

Algorithm 3.1 Our approach to generating a final 3D snaking curve by first finding a 2D snaking path through a graph and reverse projecting the 2D path to create 3D points. The 3D points act as control points to model the final 3D curve using NURBS.

Require: A 2D rectangular bounding area B, a start node s, and an end node, t.
Ensure: A 3D snaking curve of the form described in this section.

1: Create a regular grid of cells inside B and create a set of 2D nodes $\{G\}$ at the center of each grid cell. The size of the grid directly reflects the number of turns in the curve. For example, a 3x3 grid produces a snaking curve with at least 2 turns in the ideal S-curve.
2: Give $\{G\}$ a connectivity matching Figure 3.6.
3: Use Algorithm 3.2 to find a 2D Hamiltonian path through $\{G\}$ using s and t as input. This path is found using the constraints based on our observations.
4: Create a 3D curve, C, by reverse projecting the 2D nodes into 3D and assigning the 3D points depths according to the process in Section 3.4. Model the final curve as a NURBS curve.
5: **return** The final 3D snaking curve, C.

that travels through all nodes is called a Hamiltonian path.

The final problem to solve is that of automatically finding a path through the grid cells. Given a start and end node, we do not wish to find just any path that travels through the rest of the grid nodes. Instead, we wish to find a path that maintains a snaking, or S-curve-like, shape. Thus, a constraint that will enforce this property is required. The constraint we chose is to maintain a path that travels between the grid cells such that it has an overall minimum turning angle. This ensures that the resulting path will not contain any large or abrupt direction changes that may cause sharp corners.

Grid nodes are connected only to their nearest neighbors (Figure 3.6). For these simple paths this means we only have 90° direction changes but 135° and 45° direction changes can be created by travelling along the diagonals between the nodes.

This leads to our formulation of our approach given in Algorithm 3.1. The automatic pathfinding for the final curve is described first. Given a 2D bounding area, create a grid of cells inside the bounding area. The centers of these cells are nodes for a graph that forms a dual of the cells. Apply an algorithm that finds a Hamiltonian path[1] from a start point

[1]See Section 2.2 for background on Hamiltonian paths and algorithms for finding Hamiltonian paths.

to an end point, adhering to the constraint that the resulting path must have the minimum total turning angle (sum of all angles of turns travelled in the graph). The resulting path defines the order of points for a 2D curve that will travel the space and utilize a majority of the space.

A 3D curve must be created once a 2D snaking path is found. Our approach is to automatically place each 2D node, in the order according to the snaking path, at a particular depth. The depth is dependent upon the current viewpoint and several parameters. A view-dependent approach allows the 3D curve to be quickly regenerated from other viewpoints and allows different camera projections to be utilized to generate the curve. The usual perspective projection was used in this work.

Without experimentation there is no ideal method for automatically giving depth to the points to create a desirable curve. We test two different approaches based on the spacing between consecutive points: uniform spacing between points and logarithmic spacing between points. The resulting 3D points are used as control points for a NURBS curve (described in Section 3.4).

A pictorial outline of Algorithm 3.1 is shown in Figure 3.5. The method for finding a path through the nodes is described in Section 3.2 and the creation of a 3D curve from the 2D nodes and the path is described in Section 3.3.

3.2 Point Ordering

After creation of the grid of nodes, a connectivity graph is created. Every node is connected to its neighbors in the grid as demonstrated in Figure 3.6. The goal is to find a Hamiltonian path through the graph that has the minimum total turning angle. To find a solution to this problem, we create the following formulation.

Define the turning cost between a pair of edges u and v as the angle between the two edges, $\theta_{u,v}$. Over a path P with a beginning node, s, and an ending node, w, we seek to

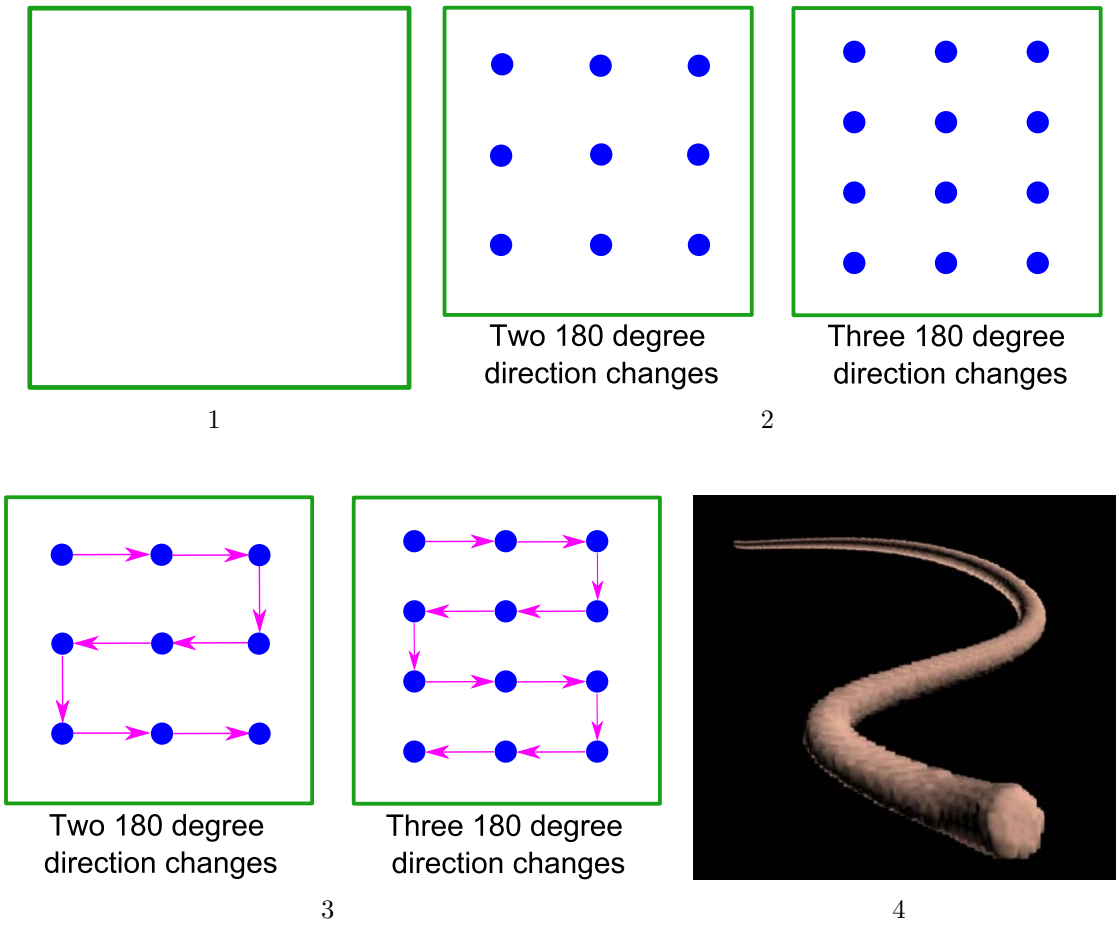


Figure 3.5: An outline of the steps in the snaking curve generation process: (1) A bounding box is defined, (2) a grid of nodes is created based on a number of direction changes, (3) a path through all nodes is found, and (4) the nodes are automatically transformed into 3D and a curve is created using the nodes as guides in 3D.

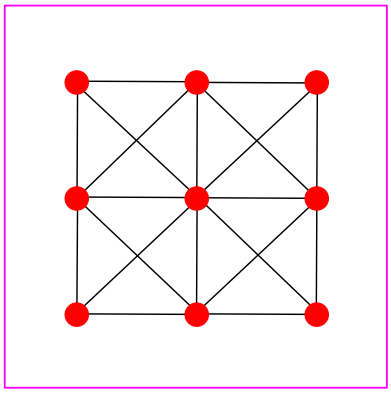


Figure 3.6: The connectivity of the nodes used in finding a Hamiltonian path of minimum total turning angle.

minimize $\sum_{u,v \in P} \theta_{u,v}$ for all edge pairs (u, v) in P. For a fully connected (complete) graph this problem is the angular-metric travelling salesman problem [1]. The solution presented by Aggarwal et al. [1], utilizes a dynamic programming approach that provides an improvement over brute force. Their solution is specific to complete graphs, however, the same solution can also be applied to a graph that is not fully connected by only iterating and memorizing those edges that are adjacent to the current node (instead of all other nodes as would be done in a complete graph).

A dynamic programming algorithm to find a path with beginning node, s, initial edge, e, with endpoint s, and a final node, w, is given in Algorithm 3.2. Following notation similar to that in Aggarwal et al. [1], we define $C(u, k, v)$ to be the cost of a path of cardinality k starting with edge u and ending at edge v. To prevent self-intersection, the cost of a path is set to infinite when an edge is encountered that causes the path to cross itself.

Algorithm 3.2 is a very similar result to the dynamic programming solution of Held and Karp for the travelling salesman problem and is of order $O(n^2 2^n)$ [45] where n is the number of nodes[2]. This large complexity makes the algorithm slow for large numbers of nodes. However, for the purposes in this thesis, this is rarely a problem as only a small number of nodes is required. A large number of nodes does not create a better snaking path because of the formulation of the grid; most of the interior nodes are unnecessary when determining the path. In fact, in a simple rectangular grid, the only nodes required are where the snaking path is to change directions. Figure 3.7 demonstrates this fact.

However, interior nodes are still useful. First, the interior nodes become control points for the final curve, therefore, they have influence over the final shape of the curve (see Figure 3.7, again). Second, it is possible to exclude specified grid nodes from the pathfinding process, thereby changing the shape of the bounding area. Excluding some nodes causes the path to travel among the remaining nodes in a different order as shown in Figure 3.8. As a result, the

[2]The calculation of the initial edge costs in lines 1 to 3 is at most $O(e^2)$ where e is the number of edges since the loop at most (in a fully connected graph) the loop iterates over every pair of edges, making it negligible.

Algorithm 3.2 Dynamic programming algorithm for the angular-metric travelling salesman problem [1] modified to find a path given a specified beginning node s, initial edge v, and end node w.

Require: A graph G, a start node s, an initial edge e (with endpoint s), and end node, w
Ensure: Minimum turning angle for a Hamiltonian path in G from s, through e, to w

1: **for all** adjacent edge pairs $(u, v) \in G$ **do**
2: Generate turning costs $\theta_{(u,v)}$
3: **end for**
4: Let $C(u, k, w)$ be the cost of a path of length k beginning at edge u, ending at edge w
5: $C(e, 1, e) \leftarrow 0$
6: **for** $i \leftarrow 2 \rightarrow |G|$ **do**
7: **for all** $S \subseteq G \mid |S| = i$ **do**
8: **for all** edges $v \in S \mid v$ is adjacent to an edge already memorized in C and $v \neq (r, w)$, for some vertex r, unless $i = |G|$ **do**
9: **if** v crosses an edge in the set of generated paths **then**
10: $C(e, i, v) \leftarrow \infty$
11: **else** {we still consider the path}
12: $C(e, i, v) \leftarrow \min_x\{C(e, k-1, x) + \theta_{x,v}\}$, x is an edge already in the path
13: **end if**
14: **end for**
15: **end for**
16: **end for**
17: $minAngle \leftarrow min_v\{C(u, |G|, v)\}$
18: **return** $minAngle$

shape of the bounding area also changes and, thus, our choice to use a rectangular bounding area does not limit the shape of the curve to be purely bound by the initial rectangular region. Results involving this feature are discussed in Section 6.2.

In our implementation the start and end nodes in the path finding are defined as the grid nodes closest to the interactively designated start and end points (in 2D space) for the final curve. When computing the minimum path through all edges adjacent to s, the algorithm is called once for each edge adjacent to s[3].

[3]With the optimization of precomputing the turning costs computed in lines 1 to 3.

(a) (b)

Figure 3.7: A curve (a) produced with a grid size of three-by-three and a curve (b) produced with a grid size of three-by-five (second numbers are width). The additional interior nodes in the three-by-five grid affect the shape of the resulting curve but not the snaking path.

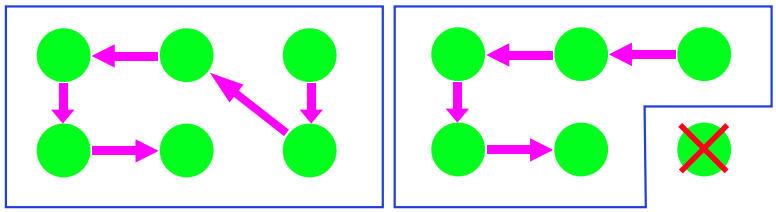


Figure 3.8: The effect on the path and shape of the 2D bounding area by excluding a node from the pathfinding algorithm.

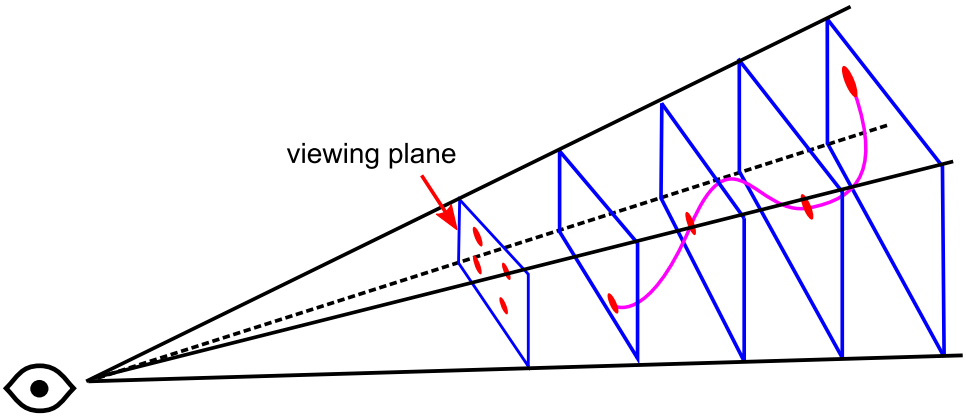


Figure 3.9: Rays (not shown) originating from the 2D coordinates in the viewing plane are shot into the scene and intersected with view-aligned planes to find 3D positions. An interpolating curve is formed by utilizing the 3D points.

3.3 3D Point Creation

The path generation is restricted to two dimensions but the final curve must be 3D. If we use the nodes in the path as control points (guides) for a curve modelling scheme then a 2D curve can be created. The shape of the 3D curve should match the shape of the 2D curve in that it should maintain the snaking shape and sweep out the 2D space. Thus, the problem is to create 3D points from the 2D nodes such that the projection of the 3D points back into 2D adheres to the constraints used to generate the 2D path. This is a difficult problem because the additional dimension can be assigned arbitrarily. We solve the problem in two steps. First, we find a way to reverse-project the points into 3D and then we determine a method for assigning depths to the points via experimentation.

Reverse-projecting the points from 2D to 3D is a typical problem encountered during object selection. A 2D point on the screen (viewing plane), for example, input via the mouse, must be turned into a 3D point to determine which object in a scene is being selected. The solution is to cast a ray from the 2D position of the mouse into the scene and perform ray-object intersection [72]. This is typically termed a picking operation because it involves picking an object from the scene.

We use the same concept to convert each 2D node to a 3D point. A ray originating at

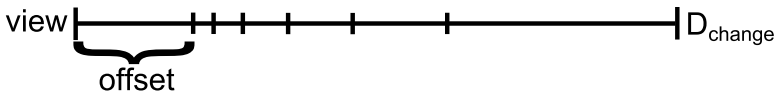


Figure 3.10: An approximation of the logarithmic spacing described in this section. Each tick represents the location of a node in the depth dimension.

each 2D node is intersected with a view-aligned plane placed at some distance away from the current viewpoint. This distance corresponds to the placement in the depth dimension the current node should lie in 3D[4]. Figure 3.9 shows this process and an example of a resulting curve.

This method of determining the 3D positions from 2D points was chosen because it allows the 3D point placement to be completed with the same process no matter what the view parameters are. The result is a view-dependent assignment that can be regenerated at any time to produce new 3D point placement from different viewpoints. This view-dependent assignment is justified since the type of illustration mimicked by our approach is by definition view-dependent.

The operation described above assumes that the depths of each view-aligned plane is known. Thus, we must determine a method of assigning depths to the planes to fully determine the 3D points. We constrain the depths to be strictly descreasing, from furthest point to closest point, because determining the depths is an arbitrary procedure. That is, are infinite many ways to assign depths to an original set of 2D points. Constraining the depths to be strictly decreasing allows us to produce a curve that travels through space towards the viewpoint.

The next issue is the distribution of the points over the total depth change. Natural choices for spacing the points include those that are easily represented mathematically and are predicatible. For example, uniformly spacing or logarithmically spacing the points, both of which are implemented in this thesis. Spacing the planes uniformly is self-explanatory so we describe how the planes are logaritmically spaced.

[4]Determining the distance is not pivotal to understanding this concept.

The idea of the logarithmic spacing is given pictorially in Figure 3.10. If the total specified depth change is defined as D_{change}, then the depth difference between successive planes is defined logarithmically according to the total number of points, c_{num}, with the relationship:

$$\log_b D_{change} = c_{num}. \tag{3.1}$$

The equation for the current depth is $d = b^{c_{num}-i}$ where i is the zero-based index of the current point for which the picking operation is being performed [5].

Finally, the plane corresponding to the picking operation of point i is defined as (n, p) where n is the negative of the view direction and

$$p = viewPos + (d + offset) * viewDirection \tag{3.2}$$

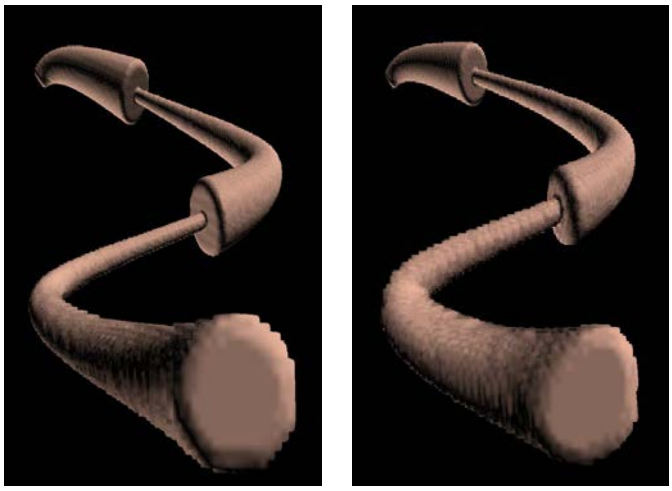
is a point on the plane. Here, $viewPos$ is the position of the viewpoint, $viewDirection$ is the direction the camera is facing, and $offset$ is a specified distance from the viewpoint to the end of the curve. That is, $offset$ is the distance from the viewpoint to the point on the curve that is closest to the viewpoint.

A comparison of the spacing methods is provided in Figure 3.11. The logarithmic spacing creates curves that are less slanted at the rear, more slanted at the front, and oriented more towards the viewpoint. Uniform depth steps create curves that appear more slanted at the rear but more flat at the front. Also, note that small depth changes create 3D curves that appear more like their initial 2D counterparts while large depth changes create snaking curves that actually seem to travel towards the viewpoint like a snake (Figure 3.12).

3.4 Final Curve Generation

The 3D points generated by the process described in Section 3.3 are used as points in a curve modelling scheme to create a final model for visualization. There are several options

[5] Lower indexed points are further away from the view plane.



(a) (b)

Figure 3.11: An example of (a) uniform depth steps between points versus (b) logarithmic depth steps.

for modelling the curve which include polyline, Bezier, B-Spline and NURBS. A polyline connects points with straight line segments but is a poor choice because our approach uses a small number of points. Without a large number of points a polyline does not appear to be a smooth curve[6]. Bezier curves have the property that each control point has a global effect over the shape of the resulting curve. This is undesirable for our purposes because it is better that each control point only have a local effect. That is, if a portion of the path changes we do not want the shape of the entire curve to change.

Thus, the curve can be either a B-Spline or NURBS curve. A problem with utilizing only a B-Spline curve is that sometimes corners in the curve appear too sharp or not sharp enough. For example, Figure 3.13(b) demonstrates that a B-Spline curve could produce corners that are too sharp. By utilizing a NURBS curve and weighting the control points contributing to the sharp corners less, the corners become smoother. The opposite is also true: corners can be made sharper for customization. Figure 3.14 demonstrates how the weights can be used to customize portions of the final curve to be sharper.

[6]In fact, all curves are displayed as polylines in computer graphics but a large number of points are found over the curve producing a curve that appears smooth.

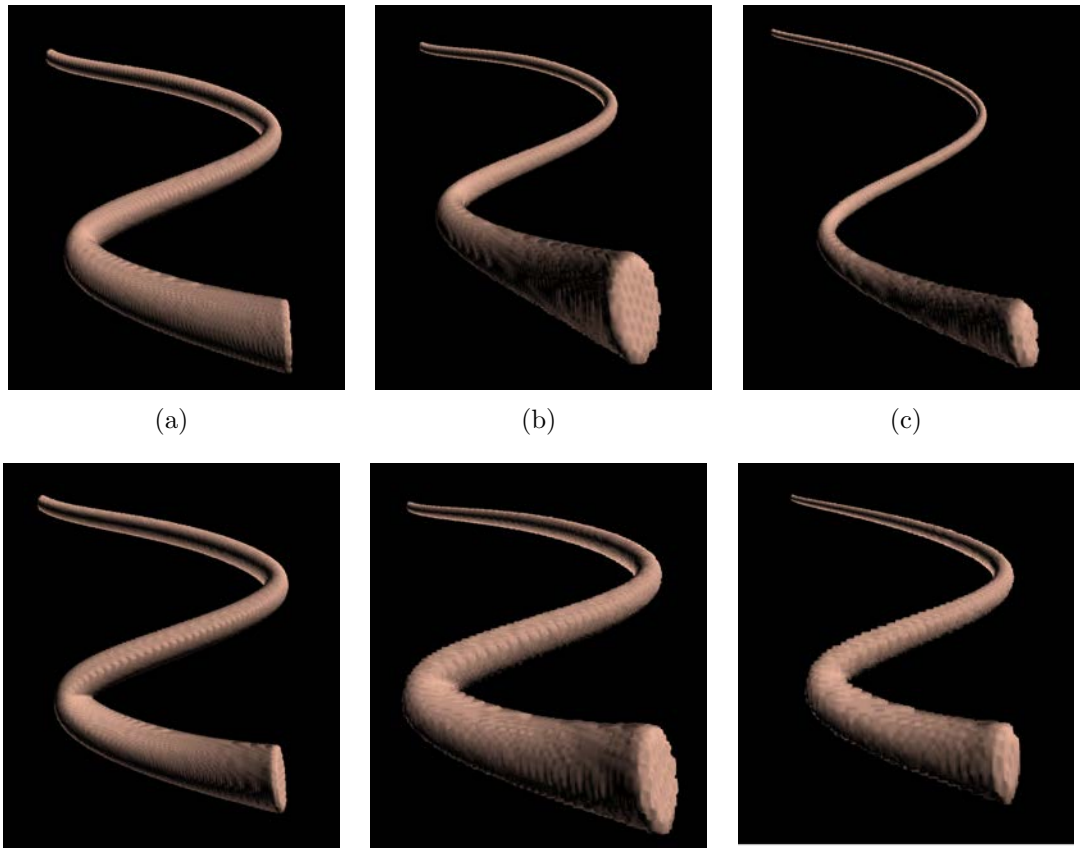
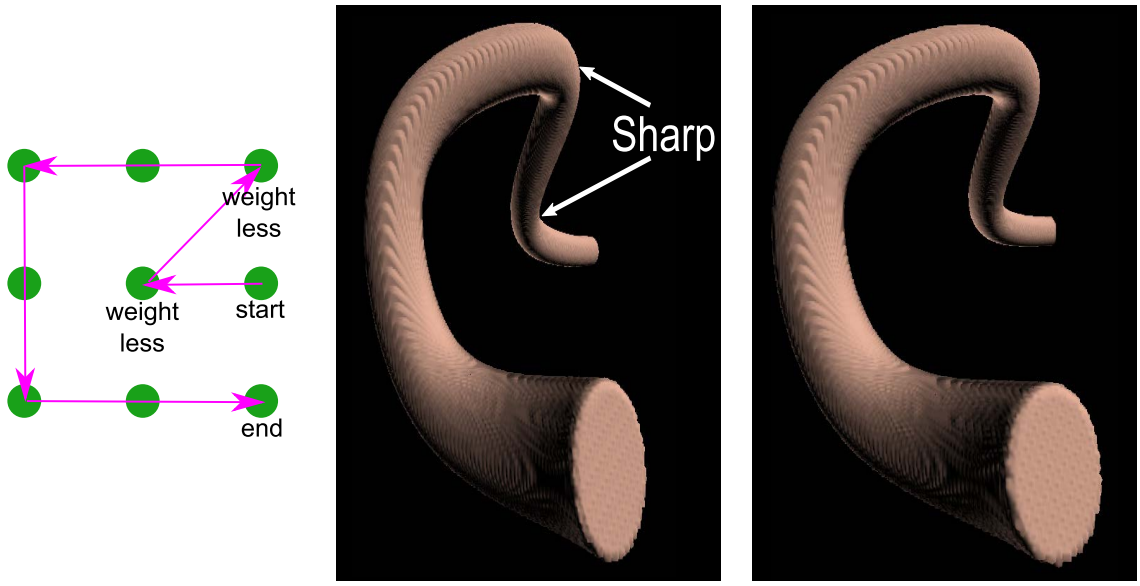


Figure 3.12: An example of (a),(d) small total depth change versus (c)(f) large total depth change. For comparison, (b)(e) a moderate total depth change is provided between the images of the small and large depth changes. The top row is created using uniform spacing while the bottom row is created using logarithmic depth spacing.

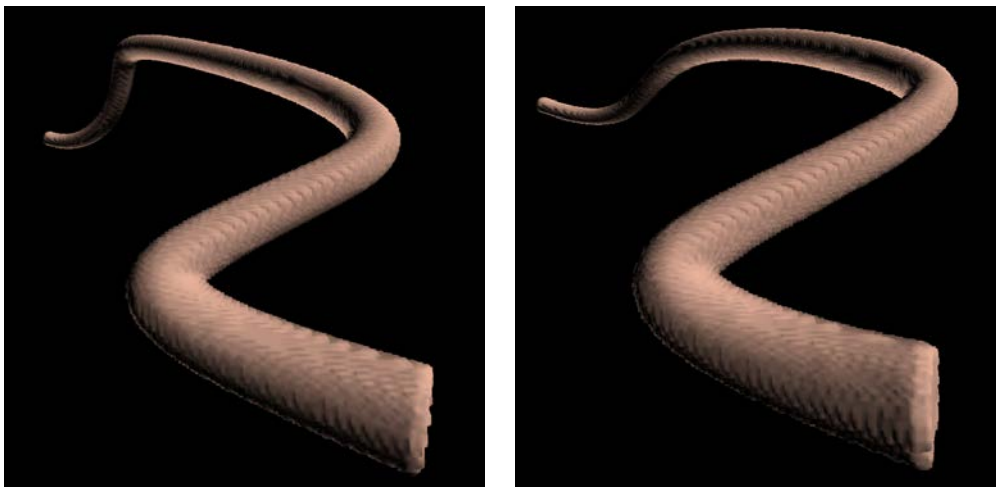
In our implementation, all weights are set to *one* as the default, making the initial curve a B-Spline curve. Abrupt turns in the path, such as those caused by 135° direction changes, are given a default weight of 0.75 in order to mitigate the sharpness of the turn, such as is shown in Figure 3.13(c).

Finally, for customization of the shape of the curve at the start and the end, the tangents are considered at those points. Control points are injected to force the start and end tangents to match those specified. Typically, we define the start tangent such that the transition from a context (or other part of the final visualization) to the generated curve seems continuous. The automatic depth assignment produces an end tangent such that it is mostly oriented



(a) (b) (c)

Figure 3.13: Control points on abrupt corners can cause (b) sharper turns, when using B-Spline curves (weights of all control points in the NURBS curve are *one*). Thus, in the final curve we enable the use of weighted control points to create a NURBS curve and assign lower weights to the control points that contribute to sharp corners (c) in order to create smoother corners.



(a) (b)

Figure 3.14: A curve created (a) by weighting the control point in the upper left corner with a large weight (5) in order to create a sharp turn. Another curve (b) without the upper left control point weighted is provided for comparison.

(a) (b)

Figure 3.15: A curve created by (a) not specifying its end tangent contrasted with a curve (b) whose end tangent is forced to the right.

towards the viewpoint. However, it is also possible to specify an end tangent. Figure 3.15 demonstrates the difference between (3.15(a)) not specifying an end tangent and (3.15(b)) specifying an end tangent.

Chapter 4

Creating Models for the Snaking Paths

In this chapter we discuss how to create a model to fit the snaking curve we generated in Chapter 3. We describe a process for deforming existing segmented volume data to follow the generated path. This is necessary because we wish to create visualizations from acquired volume data. We also demonstrate that it is possible to combine the initial volume rendering with geometric models for depicting certain portions with extra detail. This extra detail may either be unavailable in the acquired volume data or for a more artistic effect.

Volume data was chosen because one major goal in this thesis is to be able to create visualizations from acquired volume data, such as computed tomography (CT) data. Rendering acquired volume data has the ability to show complicated models, interior and exterior.

The goal is to visualize a specific structure along the snaking curve. Thus, it is necessary to differentiate between structures in the volume data, which segmented volume data enables. Recall that each sample value in a volume is called a voxel. We assume the existence of a separate volume consisting of integer voxel labels denoting to which structure a voxel in the original volume belongs. The segmentation can be performed using any method such as level set segmentation [62] or a flood fill approach [19]. However, the method of segmentation does not matter for our approach as it only requires that a classification of each voxel exists.

We perform the deformation of the structure into the shape of the snaking path. For this, a mapping from the shape of the structure to the shape of the curve is created. However, such a mapping can be difficult to find. The process to find this mapping and performing the deformation is described in Section 4.1.

Part of mimicking the motivational illustrations includes enabling the visualization of several focal areas along the generated snaking curve. As such, we break the generated

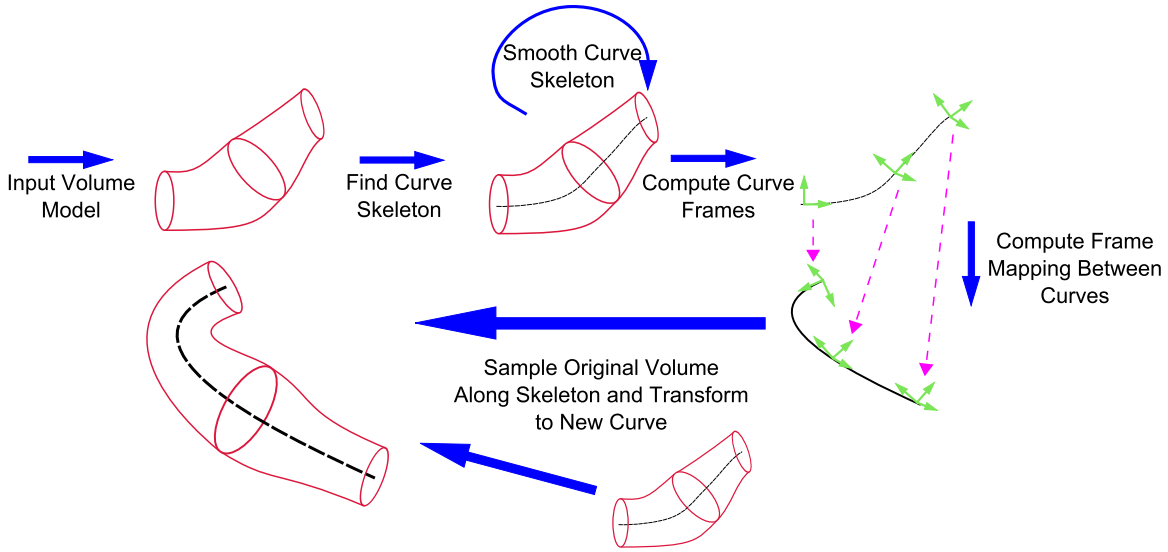


Figure 4.1: An outline of the deformation process.

snaking curve into several sections, each terminating with a focal area. This enables the visualization of multiple levels of focus+context within the same illustration. Section 4.2 describes the approach taken in this thesis to enable generated visualizations to contain multiple levels of focus+context.

The multiscale nature of the motivating illustrations demonstrates an important application of creating such an approach. Figure 1.1 depicts several resolutions of data ranging from the size of a human arm (centimetres) down to very microscopic structures that build the muscle in the arm (micrometres). These smallest structures are four orders of magnitude smaller than the size of the largest structure. A dataset containing every detail on this smallest order of magnitude would be too large to fully load into memory. By splitting the visualization into sections along the path we enable the ability to load only portions of each volume, each capturing a different level of resolution. Those portions can then be visualized along the snaking path using our approach presenting in Section 4.2.

4.1 Volume Deformation

The deformation of the original volume is performed by mapping the curve skeleton of the volume to the shape of the snaking curve generated for the visualization. We take a sweep volume approach to complete the deformation due to the generalized cylindrical nature of the structure. A graphical outline of the steps for the deformation is shown in Figure 4.1.

A sweep volume approach involves sweeping a template along the trajectory of a curve. The contour of the template defines the outer surface of the sweep volume as the template is swept along the trajectory of the curve. We set out requirements on the structures to be deformed and describe how we map the shape of the structures to the path of the snaking curve for the deformation.

4.1.1 Focal Structure Requirements

Structures that are specified for use in the deformation must have two properties: they must be cylindrical and branchless. It must be cylindrical because of the nature of the motivating illustrations. That is, the type of illustration being mimicked are created for depicting the interior layers and substructures of cylindrical objects. The structure must be branchless because a branching structure results in an unclear mapping for the deformation onto the branchless snaking curve. That is, the snaking curve onto which the structure is being mapped is branchless, thus, the original structure must also be branchless for the mapping to be successful.

Combining these requirements results in a general shape that we assume the structures used in our visualizations will have, which is a generalized cylinder. A generalized cylinder is created by sweeping the shape of a cross-section along an axis. This produces a surface with a cross-section of the same shape as the original cross-section but possibly scaled in places if the diameter of the cross-section is allowed to change during the sweep. A generalized cylinder created from a circular cross section with changing diameter and its curve skeleton

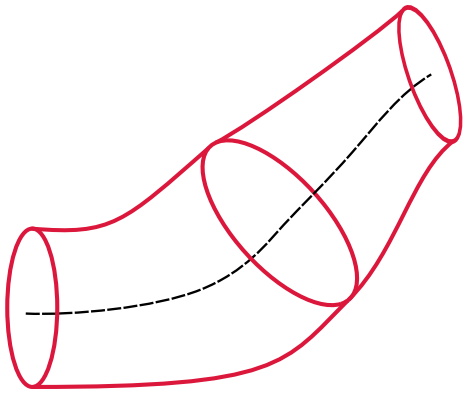
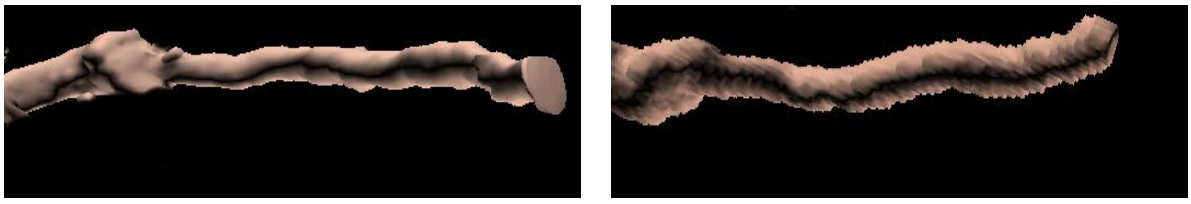


Figure 4.2: A generalized cylinder created with a circular cross section. The diameter of the cross section changes along the trajectory of the curve. The curve skeleton (black) of this generalized cylinder is the same curve as the trajectory used to create the generalized cylinder.



(a) (b)

Figure 4.3: A comparison of a portion of a volume (a) before masking using discrete segmentation data and (b) after. The aliased appearance after the masking is due to the discrete classification from segmentation, denoting whether or not a voxel is within the segmented region.

is shown in Figure 4.2.

4.1.2 Focal Subvolume Creation

We select only a subvolume of the original volume for deformation in order to create a more efficient deformation. A subvolume containing the focal structure is selected using a hexahedron, axis-aligned with the original volume. The hexahedron is placed in the 3D scene and used to mask the original volume. Other methods were not considered in order to limit the scope of this thesis.

The subvolume must contain only the specified structure for the deformation because other parts of the volume are not the focus. Thus, the final focal subvolume is created

by masking the original volume with the integer label for the designated structure (given through the segmentation data) within the specified subvolume. The label at each voxel in the subvolume is compared to the label of the structure of interest and if it is a match then the value of the original volume is copied into a new subvolume. If a match does not occur at a voxel then the new subvolume is assigned a default value.

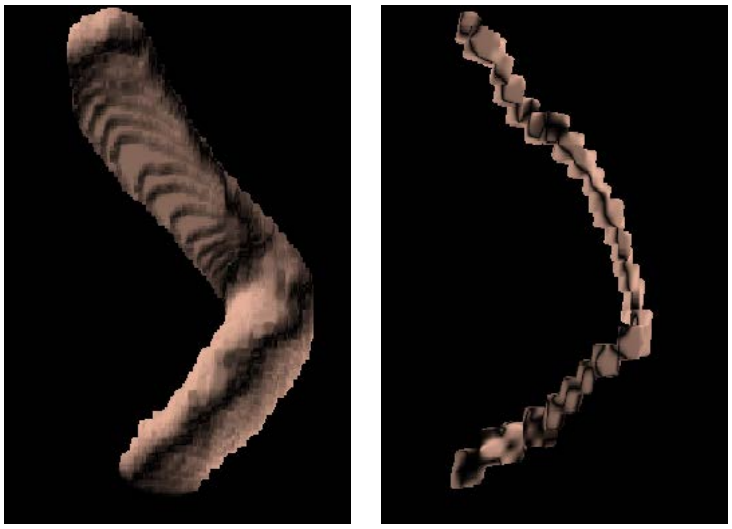
Figure 4.3 demonstrates that the newly created volume appears aliased compared to the original because of the discrete nature of the segmentation data. A fuzzy classification of voxels would have to be used to create a higher quality masked volume. The fuzzy classification gives a probability that a certain voxel belongs to a structure, and thus, voxels that would otherwise be classified as not belonging to a structure in a discrete model can be included. This would make the masked volume appear less aliased. However, because the masked volume is used in the next deformation steps, involving a resampling of the data, the aliasing is removed.

4.1.3 Curve Skeletons

A curve skeleton of a 3D shape is a "simplified 1D representation of the medial surface" [25]. To find curve skeletons, we use the medial axis thinning (MAT) of Lee et al. [54][1] that relies on decision trees due to its speed. Recall that thinning methods for finding the curve skeleton involve eroding the contour of an image until a single pixel/voxel wide curve skeleton is discovered as in Figure 4.4. The decision tree method uses look-up tables and chooses which voxels to remove based on a heuristic for how each voxel considered for removal will change the structure of the medial axis.

The curve skeleton resulting from the MAT is generally noisy because thinning is sensitive to surface perturbations [25]. This jagged curve skeleton is not ideal for the mapping and deformation that is later performed because the local frames (see 4.1.4) will replicate the jagged nature and cause a discontinuous mapping for the deformation. A smooth curve

[1]The implementation is from H.Homann [46], as an extension to VTK [51].



(a) (b)

Figure 4.4: A demonstration the curve skeleton found by MAT: (a) a volume and its (b) curve skeleton.

is created from the resulting skeleton by applying cubic B-spline reverse subdivision [64]. The smooth curve is a good approximation of the curve skeleton unless too many levels of reverse subdivision are applied. In our experience, three levels of reverse subdivison creates a smoothed curve skeleton without adversely affecting the overall shape. Figure 4.5 shows a jagged curve skeleton, resulting from MAT, shown above a curve skeleton smoothed by cubic b-spline reverse subdivision. Three levels of reverse subdivision were applied. Utilizing a potential field approach [41][20][27] for finding the curve skeleton may remove the need for smoothing, and thus, a change of shape of the curve skeleton from smoothing would no longer be a factor.

We have found that by applying three levels of reverse subdivision is sufficient to smooth the curve skeleton without adversely affecting the overall shape. If less are applied then the curve skeleton remains too jagged and when too many are applied the overall shape of the curve skeleton is affected. We have found that observing the smoothed curve skeleton along with the original structure while changing the number of levels of reverse subdivision enables enough control to create a smoothed curve skeleton that remains qualitatively accurate for

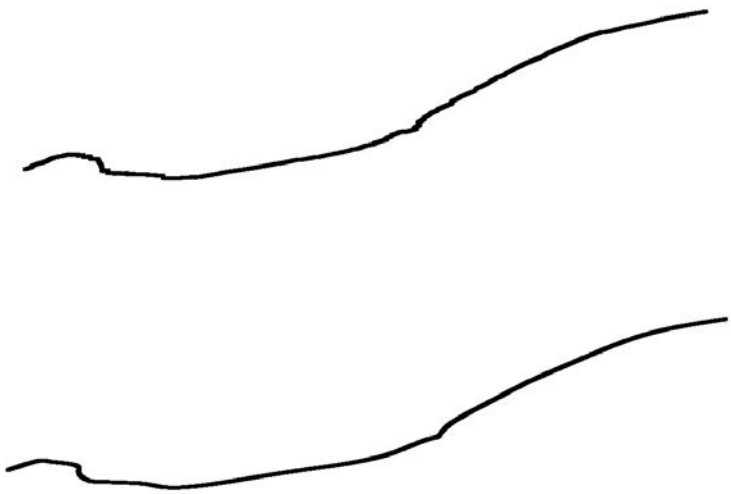


Figure 4.5: Jagged curve skeleton (top) compared to a curve skeleton smoothed by cubic b-spline reverse subdivision (bottom). The smoothed curve has three levels of reverse subdivision applied.

the purposes of the deformation.

4.1.4 Local Curve Frames and Mapping

The sweeping deformation requires a method of following both the curve skeleton and the snaking curve. At each point along each curve the sweeping template must be positioned and oriented appropriately. For this, we utilize local frames that represent position and orientation which must be computed from the curves. Finally, a mapping between the frame of the curve skeleton and the snaking curve must be found to define the transformation between the two curves.

Two ways of calculating curve frames include Frenet frames [10] and parallel transport frames [42]. Frenet frames are found by direct evaluation of the first and second derivatives of the curve. This results in undefined frames in areas of no curvature and frames that drastically change at inflection points. Thus, we do not consider Frenet frames and instead choose parallel transport frames because they lack these undesirable qualities.

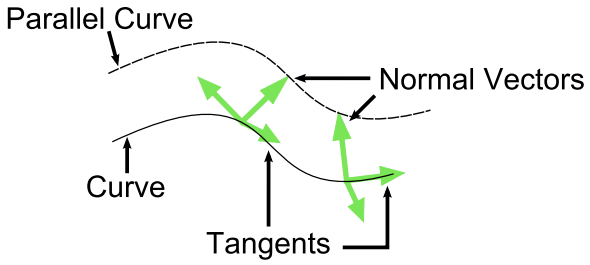


Figure 4.6: Parallel transport frames of a curve.

Algorithm 4.1 Finding a set of parallel transport frames using the approach of Hanson and Ma [42]. In our implementation, the initial parallel vector comes from computing the first Frenet frame but all frames after are parallel transported in the fashion presented here.

Require: A set of N input tangent vectors $\{T_i\}$ and an initial normal vector $V_0 \perp T_0$
Ensure: A set of vectors $\{V_i \mid V_i \perp T_i \; \forall \; i\}$

for $i \leftarrow 0 \rightarrow N-1$ **do**
  $A \leftarrow T_i \times T_{i+1}$
  if $||A|| = 0$ **then**
    $V_{i+1} \leftarrow V_i$
  else
    Normalize A
    $\theta \leftarrow \arccos(T_i \cdot T_{i+1})$
    $V_{i+1} \leftarrow V_i$ rotated around A by θ
  end if
end for
return $\{V_i \; \forall \; i\}$

Finding parallel transport frames involves using the tangent vectors for the curve skeleton and an initial normal vector [42]. The tangent vectors define the 'forward' vectors of each frame while the normal vector defines the 'right' vector[2]. These are pictured in Figure 4.6. The rotation between the tangent at the current point and the next point on the curve skeleton is used to rotate the normal vector. The axis of the rotation is the cross product between the two consecutive tangents. The rotations are accumulated and stored as quaternions to minimize numerical errors. The frame is computed as the current point on the curve and the orientation stored as the quaternion. We refer the reader to Hanson and Ma [42] for a detailed outline of the algorithm but we have reproduced the essential portions

[2]Which points to a curve that is parallel to the original curve, hence the term 'parallel' transport frames.

(a) (b)

Figure 4.7: An example of the sweeping deformation applied to (a) a structure containing a turn and (b) the resulting deformation to create an axis-aligned volume.

in Algorithm 4.1.

The mapping between the curve skeleton and the generated snaking curve is performed by mapping points between the beginning and ending of the curve skeleton using linear interpolation. Points are mapped along the two curves with respect to arc length.

4.1.5 Sweeping Deformation

In the sweeping deformation, a template for sampling the original volume is swept along the trajectory of the curve skeleton of the structure of interest. As the template is swept, the original volume is sampled and the positions of the samples are transformed to a corresponding position and orientation on the snaking curve using the curve frame mapping described in Section 4.1.4. An example result demonstrating the deformation is shown in Figure 4.7. The steps for the entire deformation, shown pictorially in Figure 4.1, are now listed as a summary.

1. Compute the curve skeleton for the structure within the region of interest. The skeletonization is performed on the subvolume containing the region of interest using MAT as described in Section 4.1.3.

2. Apply cubic B-spline reverse subdivision to the resulting curve skeleton to smooth the curve.

3. Create a cubic B-spline curve from the points that remain after the reverse subdivision in step 2. This is the curve skeleton that will be used to find the local frames.

4. Compute and store the parallel transport frames [42] of the curve skeleton created in step 3.

5. Map points on the curve skeleton found in step 4 to points along the generated snaking curve.

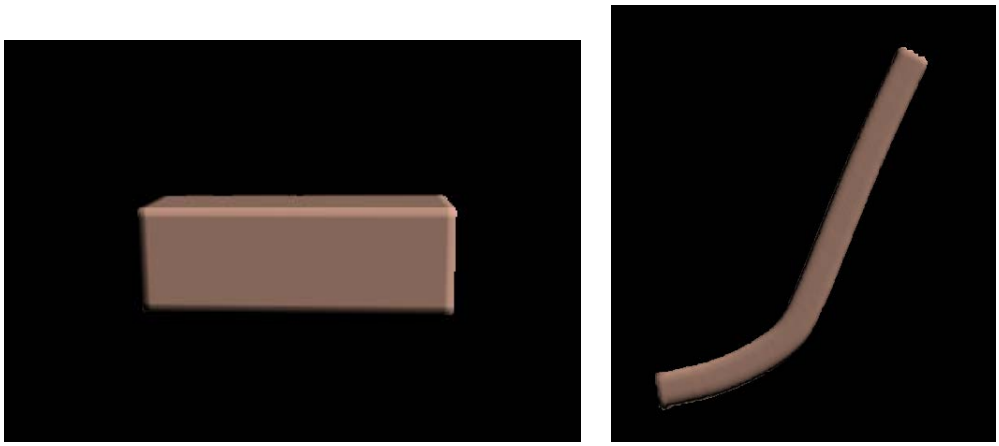
6. Use the mapping created in step 5 to transform the frames of the curve skeleton into the frames of the generated snaking curve. This also requires the parallel transport frames for the generated snaking curve.

7. Sample the original subvolume along the trajectory of the curve skeleton. Transform the position of the samples according to the mapping and transformation found in step 6.

8. Resample the deformed samples onto a uniform grid because they are currently scattered data.

Steps 1 through 4 involve processes that were discussed in previous sections. Thus, the rest of the steps are now described in more detail.

Steps 6 and 7 involve iterating over each frame in the trajectory of the curve skeleton and using the mapping onto a frame of the generated snaking curve from step 5. At each location along the curve skeleton a sweep template is oriented according to the frame at that location and is used to sample the original volume. The positions of the samples in the original volume are transformed according to the transformation between the current frame in the curve skeleton and its corresponding frame in the snaking curve. The sample values and their new positions are stored and are used to reconstruct the deformed volume.

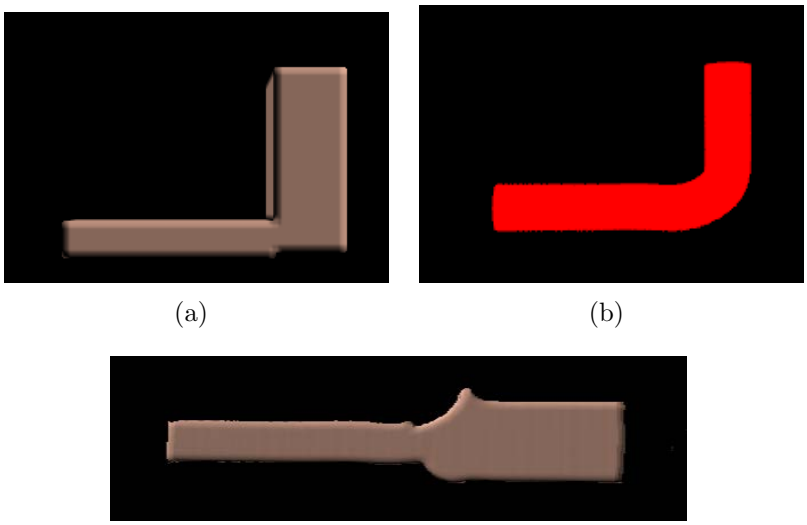


(a) (b)

Figure 4.8: A rectangular hexahedron (a) is deformed to follow an arbitrary curve (b) using the sweeping deformation approach for volumes described in this section. The deformed volume appears thinner than the original because the length of the structure has increased.

After the mapping of the original samples into their new position along the generated snaking curve, resampling onto a uniform grid is required in order to visualize the result utilizing volume rendering via texture mapping. Scattered data interpolation via the inverse distance weighting of Shepard's method [67], available in VTK [51], is used for the resampling in this thesis. It is critical for the scattered data interpolation that the deformed positions be scattered fairly evenly along the generated snaking curve so that no large empty areas affect the resampling. More local resampling methods may be more appropriate. Examples of local resampling methods include those designed for image-swept volumes in which resampling occurs after the mapping of each sweep segment [80] or for resampling volumes around skeletal points by growing spheres for the sampling areas [39], which are scan filled. In fact, the entire sweep deformation method described in this section is similar to the skeleton tree method for animating volumetric models [39] but we do not grow spheres around skeletal points and instead grow a sweep template to reconstruct the deformed volume.

A result of a sweeping deformation as presented in this section is shown in Figure 4.8. Since the rectangular hexahedron is a simple generalized cylinder the deformation is performed without distorting the original structure of the volume. In the image, the deformed



(a) (b)

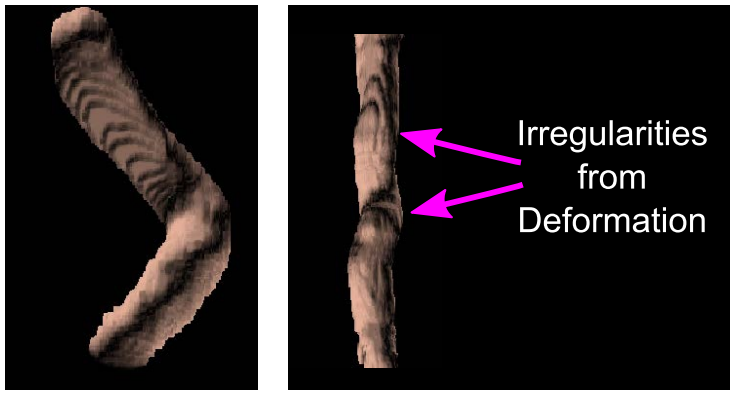
(c)

Figure 4.9: Deforming a simple volume with a semi-cylindrical nature. The (a) original volume, shown along with (b) its extruded curve skeleton, is (c) deformed to be axis-aligned. The errors in the deformation help us to determine what sort of artifacts may occur in the deformation of acquired volume data because of our assumption of cylindrical structure.

volume appears thinner than the original because the length of the structure has increased.

Figure 4.9 demonstrates the deformation described in this section on a structure with a partial generalized cylinder structure. The structure shown (Figure 4.9(a)) is made up of two rectangular hexahedrons placed perpendicular to each other to form a corner. The model is deformed to be aligned along the x-axis. Figure 4.9(b) shows its extruded, smoothed curve skeleton which appears to be perfectly smooth around the corner where the two rectanglar hexahedrons meet, however, this is due to the smoothing of the curve skeleton. The original curve skeleton contains a 90° corner. This leads to the deformation rounding out the corner when the thicker rectangular hexahedron is deformed into alignment with the thinner hexahedron. An example of an irregularity that arises when using acquired volumetric data is shown in Figure 4.10 and is due to the sharp, twisted curve of the original structure being deformed to fit a straight line.

The errors in this deformation help us to determine what sort of artifacts may occur in the deformation of acquired volume data because of our assumption of cylindrical structure.



(a) (b)

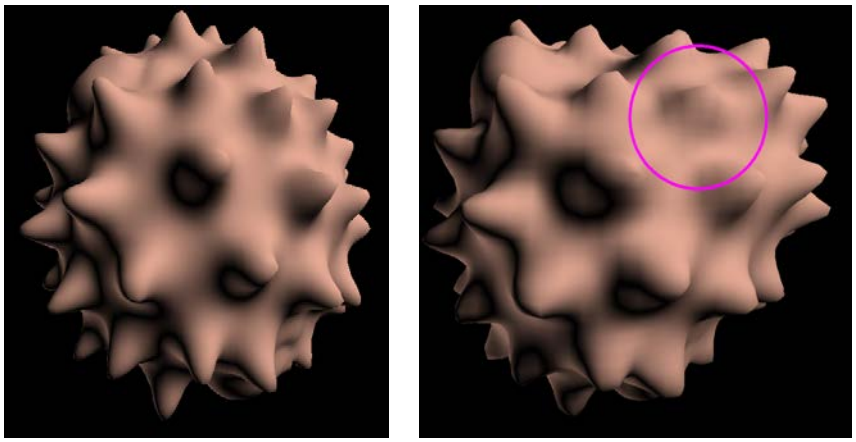
Figure 4.10: Irregularities resulting from deforming a sharp, twisted curve into a straight path.

However, for a structure such as the one in Figure 4.9, using the curve skeleton that is not smoothed would create a minor discontinuity around the corner due to the cusp. The rounding of the structure is unexpected but better than introducing a discontinuity.

The approach to deformation in this section is free-form deformation (FFD). Thus, its largest limitation is that surface details can be lost depending on the magnitude of the deformation. The structures are deformed and stretched along the snaking curve meaning that fine details can be lost, especially if the curve is long, which increased the magnitude of the deformation. For example, Figure 4.11 demonstrates that even a large surface perturbation can be slightly lost using FFD.

4.2 Multiple Levels of Focus+Context

The long, unobscured snaking path can be used for multiple areas of magnification or focal points, each terminating a portion of the snaking path. By observing the motivating images in Chapter 1, we observe that each section of the snaking path depicts structures at a different scale. The sections furthest from the viewpoint show structures on a large scale while those closest to the viewing plane show structures on a very small scale. For example, the motivating image in Figure 1.1 depicts muscle in the further portions of the curves as



(a) (b)

Figure 4.11: A deformation to demonstrate the loss of surface detail when structures are deformed. (b) The area of lost detail is circled. This example uses the FFD method of Sederberg et al. [66] rather than the sweeping approach presented in this section. However, the sweeping approach suffers from the same limitation when the deformation is large.

having smooth outer layers but the largest scale portion, closest to the viewing plane, shows the microscopic structures of muscle tissue.

To mimic this effect, the generated snaking path can be broken into sections. Each section is used for a different scale or focal level. In a sense, each previous focal level acts as the context for the next level. Thus, a final visualization utilizing the generated snaking curve can contain multiple levels of focus+context with a smooth transition between each level.

When deforming volume data along the generated snaking path, volume data from successively higher resolutions of the same dataset is used in each section of the path. The sections of the curve closest to the viewpoint use the highest resolution available and the resolution of each section decreases as the sections become further from the viewpoint. If the final curve only has one section then only one level of resolution is used.

Locating the correct regions of interest from each resolution requires a mapping between resolutions. Although, a mapping between resolutions does not guarantee the mapping will safely map the curve skeleton between resolutions. Figure 4.13 shows that the curve skeleton

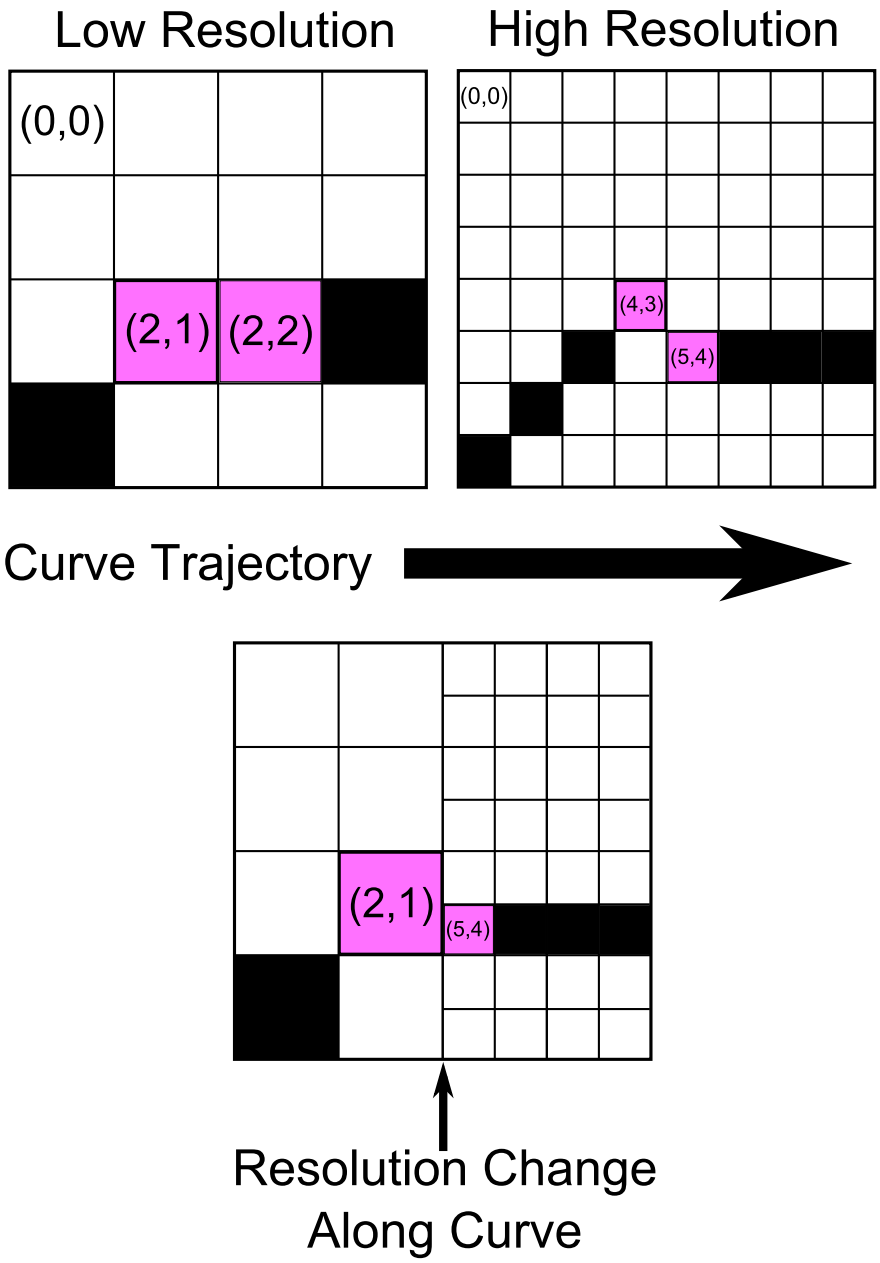


Figure 4.12: Colored blocks denote the curve skeleton. Locating the next point along the curve skeleton along the border between low and high resolution areas requires finding the nearest point in the curve skeleton in the higher resolution. In this case, (2,1) in the low resolution matches up with (5,4) in the high resolution which can not be located without a search in the neighborhood of the points in the high resolution that fill (2,1) in the low resolution (those are (4,2), (4,3), (5,2), and (5,3) in the high resolution).

will have the same overall shape between resolutions but direction changes of the curve skeleton have higher fidelity in the higher resolution. To solve this problem, the mapping between resolutions is used to locate the entire region of interest in all resolutions. The curve skeleton for the structure of interest is then found in each resolution separately.

The sweep deformation sweeps along the curve skeleton of the lower resolutions until the end of the first section of the snaking curve is met. The voxel location of the curve skeleton at this point is then mapped into a voxel location in the next higher resolution and the sweeping deformation continues along the curve skeleton in the higher resolution data. For example, when point (2,1) in Figure 4.13 is met, its location is mapped to (4,2), which is the upper left corner of the two-by-two block in the higher resolution that corresponds to the same location. Following the direction of the curve frames (to the right in this example), the proper point in the high resolution is (4,3) which borders (5,4). Finally, (5,4) is used as the location for the beginning of the next section of the sweeping deformation.

In this work, when different resolutions of acquired volume data were needed we obtained lower resolutions by downsampling the original volume data by half. Thus, when locating a voxel along the curve skeleton for the next higher resolution the voxel coordinates are multiplied by two. The unsmoothed curve skeletons are used for the voxel location to prevent any mapping errors. Once the voxel locations are found in the higher resolution, the nearest location on the smoothed curve skeleton is used as the location to resume the sweeping deformation.

It is also possible to use data obtained from different modalities (CT or Magnetic Resonance Imaging, for example) for different portions of the snaking path instead of only using different resolutions of the same data. The only requirement is that a mapping exists between the data and the original data used to locate the structure. The mapping will be one-to-one if the datasets are co-registered in the same resolution.

Enabling the use of different resolutions of data from possibly different datasets enables

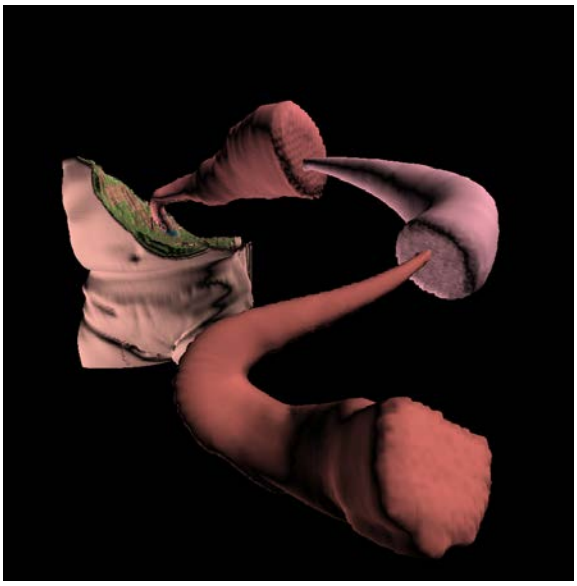


Figure 4.13: A result containing three sections along the curve. Each section receives its volume data from a consecutively higher resolution of the original volume data. Because the original volume data was already fairly low resolution, the differences are minor.

the creation of models and visualizations across structures whose size differ in several orders of magnitude. For example, an aorta in the human body is on the scale of centimetres but blood cells are on the scale of micrometres. A dataset of a human torso containing the aorta but captured at a resolution small enough to capture the structure of blood cells would be far too large to fully load into memory[3]. By splitting the visualization into sections along the path we enable the ability to load only the required portions of each volume. Some of these volumes may be captured at a very small scale while the others may be captured at a large scale. This enables our approach to be extended to extreme multiscale visualizations.

However, a problem with using acquired volume data, such as the CT data used in this thesis, is that sometimes it does not contain enough information to discern between interior layers. In the case of a human aorta, these could include things such as the layers of the vessel wall, the blood flowing through the vessel, or even at a microscopic level, the red blood cells the aorta is transporting.

[3]A volume with 8 bits per voxel of size 15cm-by-15cm-by-20cm of the human torso but captured at a resolution such that structures measuring only micrometres in length could be seen would be approximately 320 petabytes.

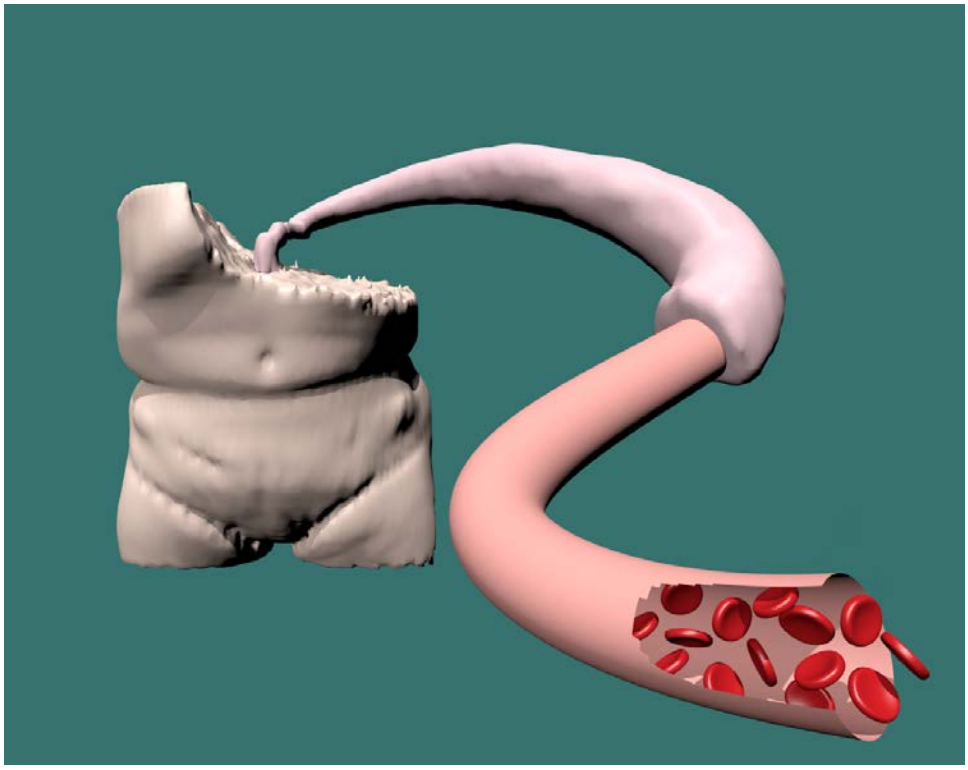


Figure 4.14: A visualization with a curve generated with our approach. The volume model of the torso is very low resolution while the aorta, found by sculpting and deformed over first half of the curve, is a higher resolution than the volume model in the background. The last portion of the curve is a surface model created along the generated curve. The surface model was exported and edited to be partially cut and contain red blood cells to illustrate the possibility of quick edits for depicting certain structures in higher detail or larger scale than the original acquired volume data allows.

Our approach allows a surface model to be created to follow the generated snaking path in any designated section in cases where the volume data is not sufficient to illustrate a structure in a larger scale or with greater detail. We generate a surface model of a generalized cylinder along the generated snaking path to provide an initial model that an artist can later edit or add details using modelling tools such as Blender [37]. This creates a hybrid visualization that mixes the benefits of exploring real acquired volume data with the imagination of an artist to give extra detail that the original data may lack.

The one downside of this hybrid approach is that the same volume sculpting techniques used to explore the volume data cannot be used to explore the geometric data of the surface

model. However, the information that is gained from the extra detail far outweighs the inability to utilize the same exploration tools. If our approach is used to create visualizations for which the intent is to view a final static image instead of being used as an interactive focus+context approach then this is definitely an added bonus.

Figure 4.14 shows a visualization that utilizes the volume-geometry hybrid apporach described previously. Constrained sculpting operations (described in Chapter 5) help to reveal the aorta which was then deformed to create the first half of the curve. The aorta structure was taken from a higher resolution version of the same stented aorta dataset and is rendered in a higher resolution than the original volume model in the background.

A surface model was created to follow the last portion of the curve. The surface model was exported and red blood cells to depict a microscopic view of oxygenated blood flowing through the aorta were added. This small amount of editing demonstrates that once a model is created to follow the snaking curve it can be changed to depict structures in higher detail than the original acquired volume data has the ability to. It is possible to further edit the surface model to produce a model that appears more like the structure it originates from (the aorta in this case), however, that is left to the artist. The visualization in Figure 4.14 serves to demonstrate that the approach presented in this thesis can be used as a quick method of preliminary modelling. Several easy modifications can be applied to the preliminary model and a final, polished visualization is created once a basic structure is generated in the form of the snaking path.

If volume data that accurately represents multiple scales of interior structures could be created then there would be no need for this hybrid volumetric-geometric approach. Wang et al. [76] present a method for modelling and rendering a form of volume data that contains multiscale embeddings in a vector form, thereby allowing for the representation of complex interior structures. Their volumes are modelled as a combined set of implicit surfaces, whereas our approach has been concerned with multiple resolutions of the same

implicit surface. That is, each resolution of the volumes used in this chapter, define the same implicit surface. In Wang et al., each implicit surface can be unique and the surfaces are combined together to form the multiscale embedding. However, the deformation presented in this chapter could be adapted to their volume format because they explicitly store the transformations of each object in the volume. Thus, a mapping exists that would allow them to align the required levels of detail.

Chapter 5

Constrained Sculpting via Point Radiation for Focus+Context

In order to create images similar to the motivating illustration shown in Figure 1.1 it is necessary to have some method that will remove occluding material without affecting the structure of interest. In the motivating illustration, the muscle tissue in a limb provides the context for the snaking curve illustration that shows several focal sections containing the structures of the muscle tissue. It depicts part of the muscle cut away to reveal a cross section of the muscle tissue. Exterior structures are also removed. However, some smaller interior structures at the point of the cut away remain to provide further context. In our illustrations, we aim to mimic the same effect of removing occluding structures from a volume model to provide a clear context of the location of the structure, while leaving the structure of interest intact.

Sculpting operations that deal with the removal of material, such as chiseling and drilling, are often difficult to control. It can be very easy to accidentally remove all material or destroy part of a structure that was to remain. Thus, we wish to constrain these types of sculpting operations so that they are able to preserve structures that should remain unaffected by their operation. In Section 5.1, a brief overview of the point radiation technique for sculpting, which forms the basis of our constrained sculpting, is given. The constrained sculpting method and the modifications made to point radiation to create constrained sculpting are then described in Section 5.2. Finally, we describe how we used constrained sculpting in this thesis, and the process we went through to create several of our results in Section 5.3.

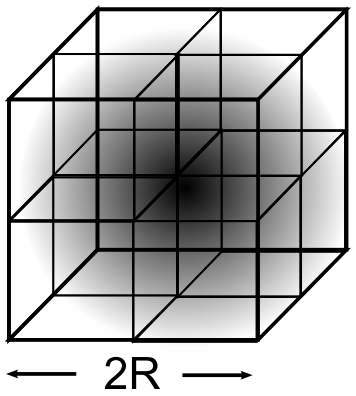


Figure 5.1: A 3D footprint of size $2R$ used for spreading energy radially using a Gaussian kernel of radius R.

5.1 A Brief Overview of Point Radiation

This section is meant as a brief overview of the point radiation technique since point radiation is described extensively in Chen et al. [19]. The technique enables the creation of high-quality, interactive sculpting tools that are anti-aliased by their nature, making it an attractive choice for sculpting.

A single point is radiated across a 3D footprint of size $2R$ where R denotes the radius of a 3D Gaussian kernel used for spreading energy radially as shown in Figure 5.1. Combining a set of overlapping 3D footprints implicitly defines a volume that is anti-aliased due to the smooth fall-off of the Gaussian kernel.

A set of 3D footprints are rendered to a *sculpting volume*, which contains a collection of radiated points created by tool-volume interactions. It is this collection of radiated points that defines the combined influence of the sculpting operations. The sculpting volume is used to mask the original volume during rendering. When the original volume is rendered the sculpting volume is sampled and if the value contained in the sculpting volume is below a certain threshold then the original volume value is rendered. Otherwise, the original volume is not rendered. This creates the effect of deleting material from the original volume without modifying the original volume and this formulation creates the basis for a number of deletion tools such as drills and lasers.

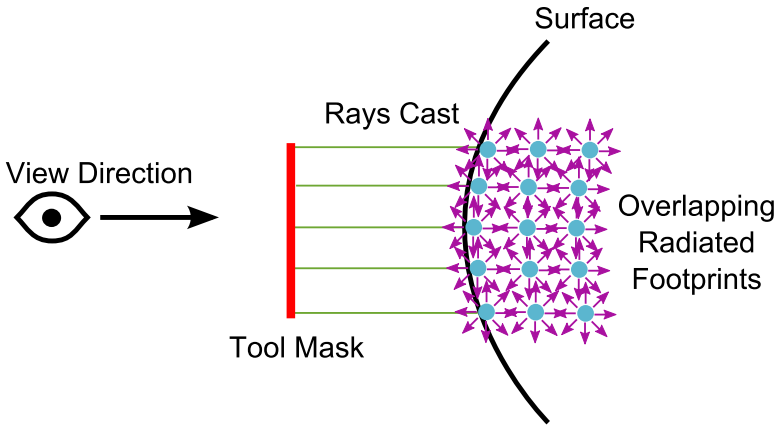


Figure 5.2: Rays are cast from a screen-space tool mask that defines a drilling tool towards the surface of the volume. Points, radiated in their 3D footprints, are spawned in layers according to the view direction.

Chen et al. [19] create sculpting tools from the point radiation technique utilizing a screen-space tool mask. The mask provides the initial points for radiation. When a sculpting action occurs, rays are cast from the 2D area of influence into the volume to find the affected points in 3D. The 3D positions are used as the initial points for radiation and more points (and their 3D footprints) are spawned in layers, either along the viewing direction or the surface normal at the original affected points (each creating different effects) to propagate the effect inwards. The number of layers and nature of the propagation is determined by a specified depth and the nature of a particular tool, respectively. Figure 5.2 gives a pictorial example of this process for a drilling tool that spawns interior layers of points according to the view direction.

5.2 Constrained Point Radiation

Constrained point radiation is an extension of point radiation that limits the removal of material to certain voxel labels. This creates the additional requirement for segmentation data in the form of an integer *label volume*. The label volume contains per-voxel labels that classify the voxels of the original volume. The label volume is stored as an additional 3D texture that is sampled when it is necessary to classify voxels.

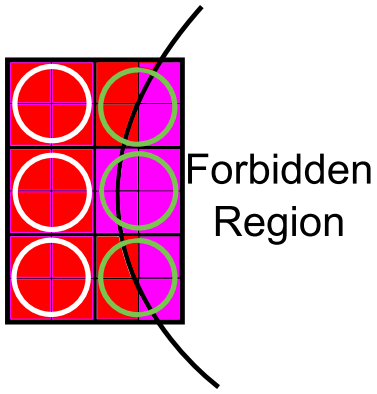


Figure 5.3: A simplifed 2D illustration of voxels that must be discarded, shaded red, during constrained point radiation. Footprints covering sets of voxels, designated by the circles, and a curve that denotes the surface of a forbidden region that should not be affected by the current interaction are shown. The magenta shaded voxels represent areas of that should not be sculpted because they lie within the forbidden region. Notice that some footprints cover areas that should be sculpted and areas that should not.

Recall that in the point radiation technique, the 3D Gaussian value for each footprint determines the value to write to the sculpting volume. If the calculated Gaussian is below a set cutoff value, then nothing is written to the sculpting volume. In order to constrain the point radiation, this process is modified. Not only must the Gaussian be computed but an additional test must be performed to classify the voxel at the current sculpting location. That is, does the current voxel lie within a region that should not be affected by the current sculpting action? We call the region that should not be affected the forbidden region.

However, all 3D footprints within a forbidden region cannot be fully excluded from the sculpting. The problem is that some footprints contain areas that do not lie within the forbidden region and some that lie within the region to be sculpted, as shown in Figure 5.3. In Figure 5.3 the footprints in the right column are contained within the forbidden region but if they are completely excluded from the sculpting then the surface resulting from the sculpting will be incorrect. Instead, it is necessary to compute which voxels within each footprint are in which region.

When considering voxels in the footprints for storage into the sculpting volume, voxels are excluded from the sculpting by sampling the label volume at the same coordinate. If the

label matches that of the label designating the forbidden region then the sculpting action is not performed. The result of constrained sculpting is shown in Figures 5.4 and 5.5.

It is important to note that the integer label volume is sampled using the nearest neighbor sampling to preserve the integer label. If the values were sampled using a method of interpolation it would be possible to retrieve invalid or unexpected labels. Using nearest neighbor sampling, however, has the side-effect of creating aliasing around structures that have all occluding material removed. Figure 5.6 demonstrates this aliasing effect (especially if zoomed in on a digital copy of this thesis) very clearly because of the nature of the folds in the brain. This problem also exists in rendering labelled volume data and not just in sculpting. In fact, it is possible to attribute the problem to the rendering method and not the sculpting method.

Since there is an existing method that deals with rendering integer labelled volumes in high-quality, we propose to solve the problem by employing the rendering method of Tiede et al. [69]. Their method mitigates the inability to interpolate between the labels by counting the number of labels surrounding a particular location. If an above average or median amount of labels for one particular structure lie in the neighborhood of the currently considered position, then the currently considered position is assumed to be in the particular structure. Thus, this method of rendering should be used when rendering the constrained sculpted data.

5.3 Constrained Sculpting and Snaking Curve Focus+Context

The constrained sculpting is a valuable asset to the creation of the snaking curve focus+context visualizations that are the goal of this thesis. Creating the constrained sculpting was driven by the motivating image shown in Figure 1.1. In this image, the muscle tissue of a limb provides the context for the snaking curve illustration that shows several focal sections containing the structures of the muscle tissue. The illustration depicts part of the muscle cut

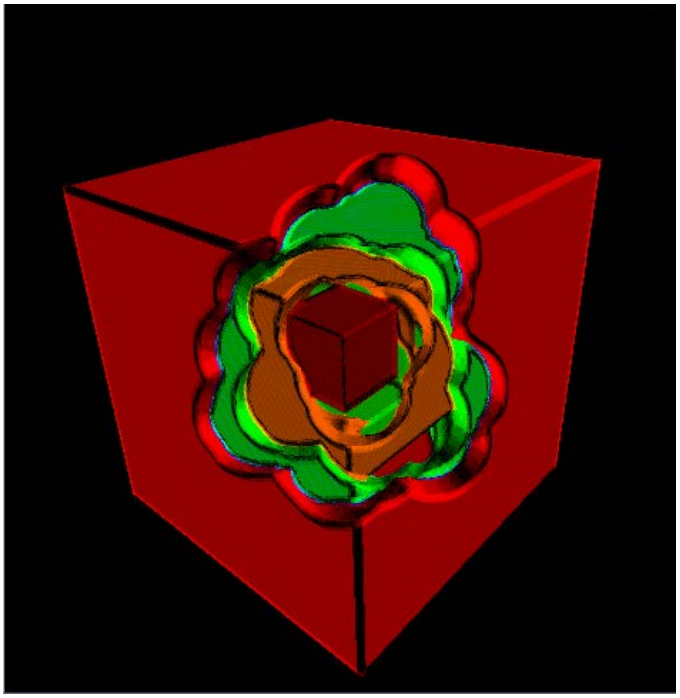


Figure 5.4: A smoothly sculpted dataset using constrained point radiation on multiple layers of a layered cube.

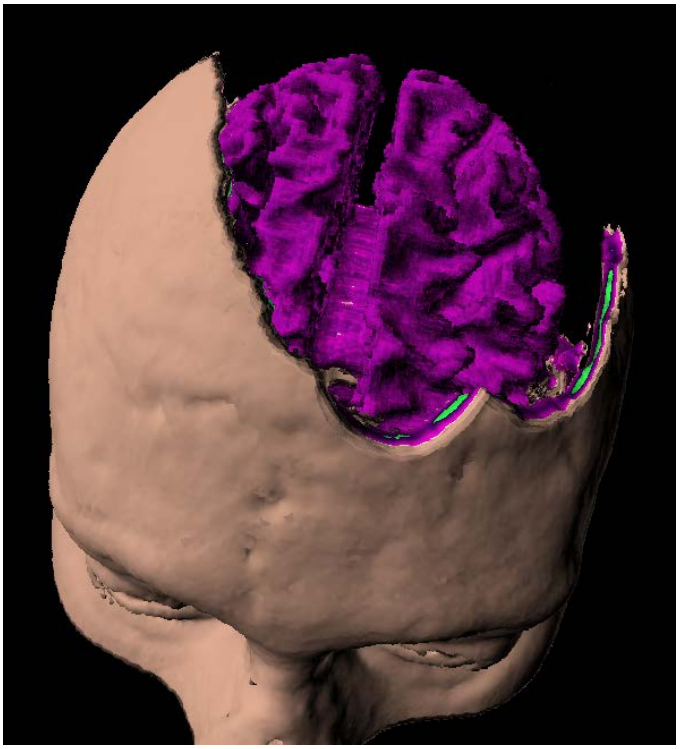


Figure 5.5: Sculpting the skull down to the segmented white matter. Dataset courtesy of BrainWeb: Simulated Brain Database (http://www.bic.mni.mcgill.ca/brainweb/).

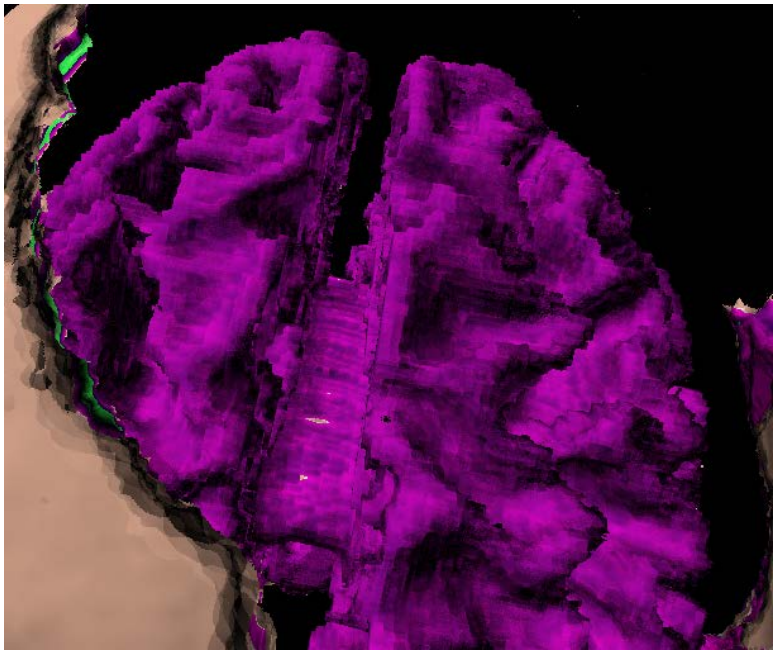


Figure 5.6: Aliased appearance due to sampling an integer label volume with nearest neighbor interpolation. The solution is to use an appropriate rendering method for integer labelled volumes.

away to reveal a cross section of the muscle tissue. Exterior structures of the limb are also removed.

In our illustrations, we aimed to mimic the same effect of removing occluding structures to provide a clear context of the location of the structure. This led to the creation of the constrained sculpting, which, along with the generation of snaking curves described in Chapter 3, has enabled the creation of the process we used to create the focus+context illustrations from acquired volume data in this thesis. Illustrations created following the steps described in this section are found in Chapter 6 of this thesis.

The steps we followed are shown pictorially in Figure 5.7, We assume the existence of a volume model and its corresponding label volume based on segmentation. The steps include the following:

1. Designate a structure that remains unaffected by the sculpting operation. This is the structure of interest.

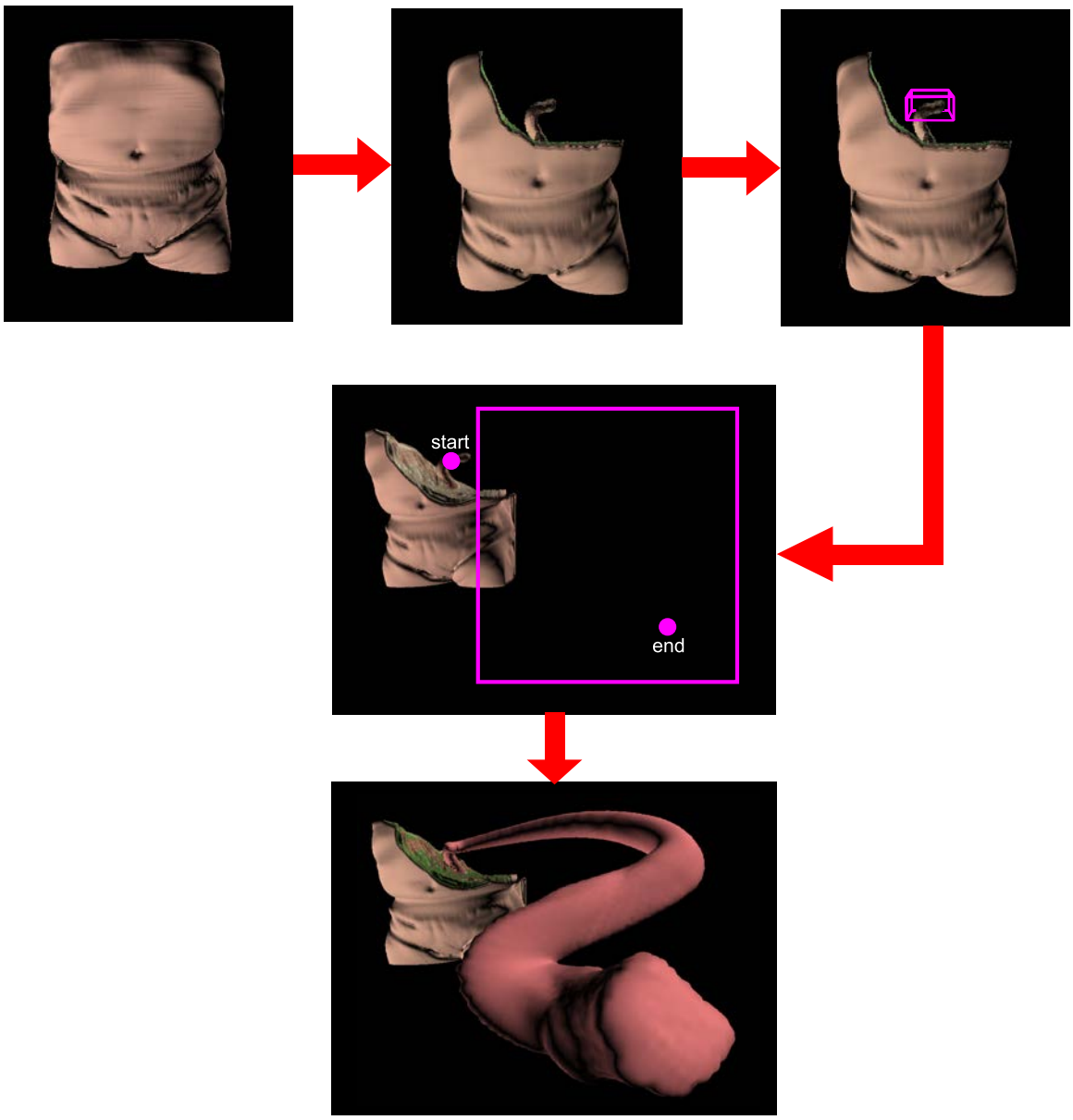


Figure 5.7: The interaction process we went through to generate illustrations in this thesis using constrained point radiation. In this example, the aorta is chosen as the structure of interest and sculpted towards using constrained point radiation. A bounding box, the start, and the end point for the snaking curve generation is specified and a final visualization is automatically created.

2. Sculpt in the area the structure is believed to be. Using the constrained sculpting, less care has to be taken because there is no worry of removing any portion of the volume containing the structure of interest.

3. Once the structure is revealed, utilize the snaking curve generation and specify the necessary information for the curve generation. This includes the 2D bounding area and the 2D start and end points. The 2D start point should match to the location of the structure of interest is located in 2D on the screen. From this location, the 3D position of the start point along the structure of interest can be found with a picking operation.

4. Specify any other information necessary for the snaking curve generation such as total depth change and number of resolutions of data. If certain curve portions should be created with a surface model instead, specify those. In our implementation this information is specified in a configuration file but would ideally be part of a user interface.

If changes are required then the snaking curve generation can be restarted. The final visualization can be saved or further explored. These steps enable exploratory discovery of structures and aid in revealing the structure of interest for the creation of focus+context illustrations. This is completed through the removal of occluding material around the structure of interest without affecting the structure.

Chapter 6

Results and Discussion

The discussion and presentation of results from our approach is split into several sections. A wide range of curve generation results are presented in Section 6.2. We then present a focus+context visualization created from texturing a result from the curve generation process (without the use of any acquired data) in Section 6.3. This provides a proof of concept for evaluating the effectiveness of our generated snaking curves in the desired focus+context style. Several visualizations created from objects such as a coaxial cable and fossilized wood are presented in Section 6.4.1. In Section 6.4.2, we present visualizations using pure volumetric data and those using our hybrid volumetric-geometric approach. Finally, discussion is summarized and the limitations of our approach are presented in Section 6.5.

6.1 Evaluation Criteria

All of the results and visualizations are evaluated qualitatively. There are three criteria we use in evaluating whether a curve is a success. The first and most important is whether or not it is a snaking path. Second, we determine whether or not the curve sweeps out a majority of the original bounding space. The last criteria is the shape and orientation of the curve as it travels through space toward the viewpoint. In particular, the tangents along the curve should be oriented mostly towards the viewpoint instead of away from the viewpoint. Curves that are mostly oriented towards the viewpoint are better because there are more spots along the curve where focal areas can be placed.

6.2 Curve Generation Results

In this section we present several different 3D curves whose paths are generated using our approach presented in Chapter 3. All examples of curves in this section are visualized in 3D and produced by using the sweep volume technique as described by Chen [18][1]. Several different shapes of paths can be generated by changing the bounding area, moving the 2D nodes, and excluding some nodes from the path finding process. We demonstrate that each of these seemingly minor modifications provide powerful customization mechanisms.

A simple curve is shown in Figure 6.1. The curve is split into four sections and each section terminates with a larger focal area, oriented slightly towards the viewpoint. Ideally, the shape of the curve will be such that anywhere these focal areas lie along the curve, they will always be oriented somewhat towards the viewpoint so that the focal area is visible. Figure 6.2 demonstrates that when moving the focal points along the curve there is only one small portion that has undesirable orientation.

The effect of changing the bounding box is demonstrated by comparing Figure 6.1 and Figure 6.3. Both curves travel from the same start and end points with the same curve configuration but figure 6.1 is generated with a bounding box that is taller than it is wide while Figure 6.3 has a bounding box that is wider than it is tall. The wider path enables the visualization of a greater surface area containing the outer layers of each section but Figure 6.4 shows that it contains many areas that are not oriented towards the viewpoint.

Next, we demonstrate how the 2D nodes can be interactively moved to change the overall snaking shape of the curve in 3D. An example of a curve created by manipulating the position of the nodes is shown in Figure 6.5. Figure 6.6 shows how the points are moved in 2D to produce the desired slanting shape. The automatic path finding is confined to the bounding box but the start and end points can be placed anywhere. In Figure 6.6, the last point is moved a little outside of the bounding box which enables the end of the curve to leave the

[1]This sweep volume technique uses point radiation to create an anti-aliased volume.

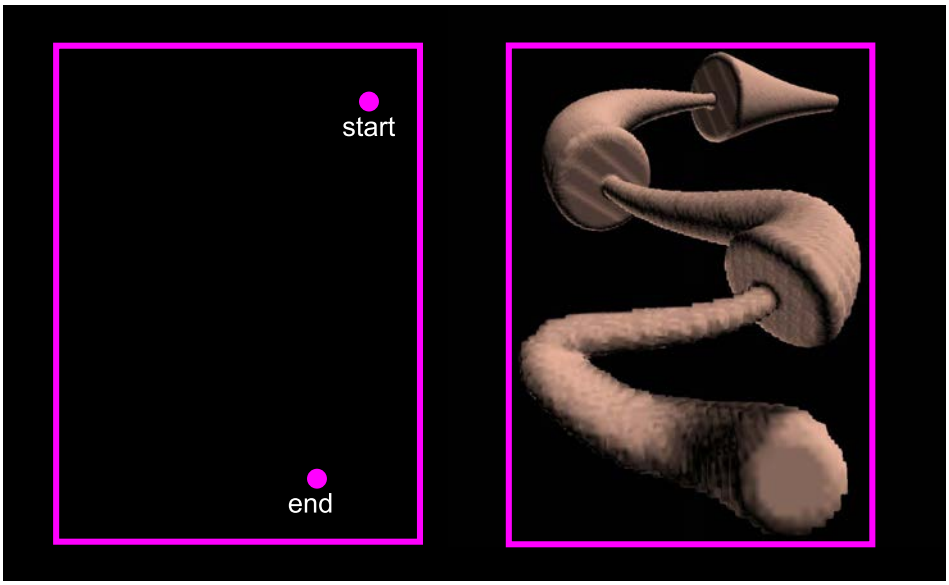


Figure 6.1: An example of a 3D curve created using our snaking curve generation approach. The curve is split into four sections each of which terminates with a larger area that is oriented towards the viewpoint to create an area of focus. The curve is generated with the input on the left and a three-by-four grid of nodes for the path finding (three direction changes).

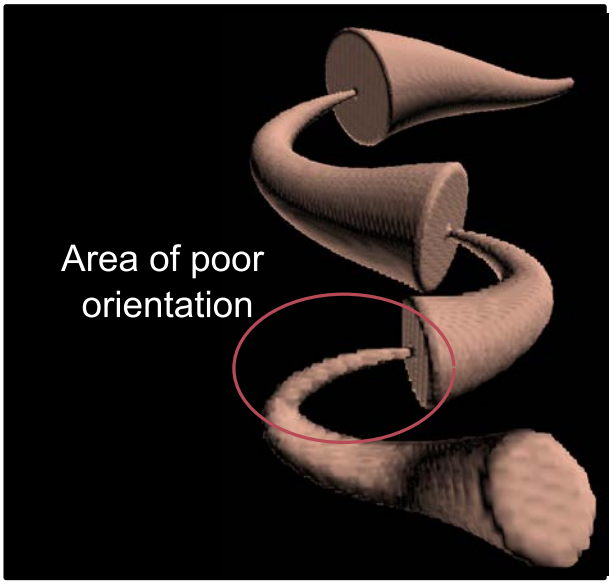


Figure 6.2: A demonstration of how the orientation of the focal points change as they move along the curve. They can not lie anywhere on the curve because the orientation may not be toward the viewpoint. However, in this curve that is only within the labelled area.

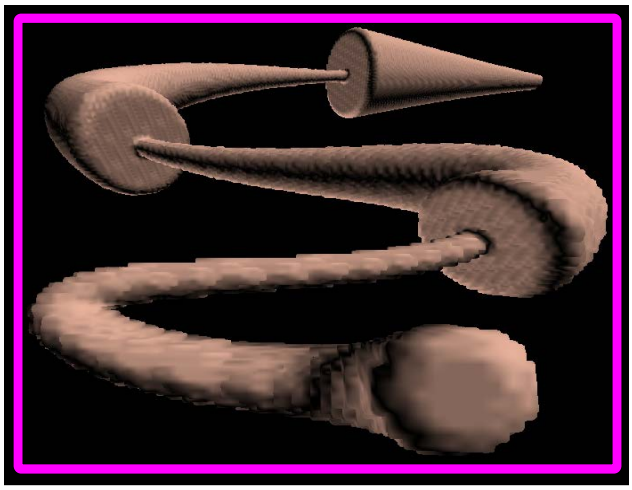


Figure 6.3: The curve shown here was generated using a bounding box wider than it is tall (magenta). Compared to Figure 6.1, this curve sweeps out a wider path, which changes the orientation of each of the focal areas.

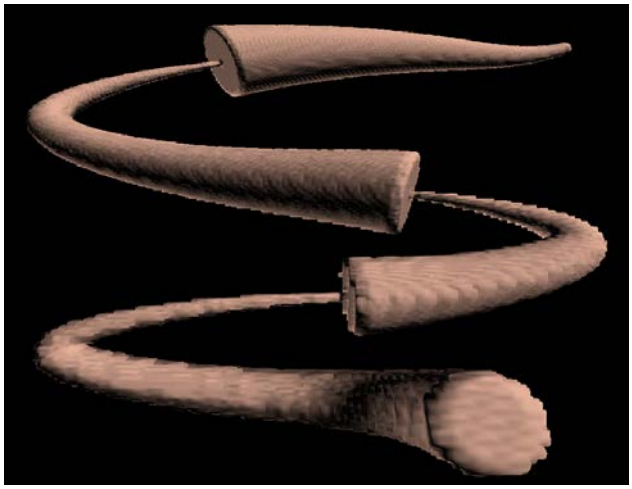
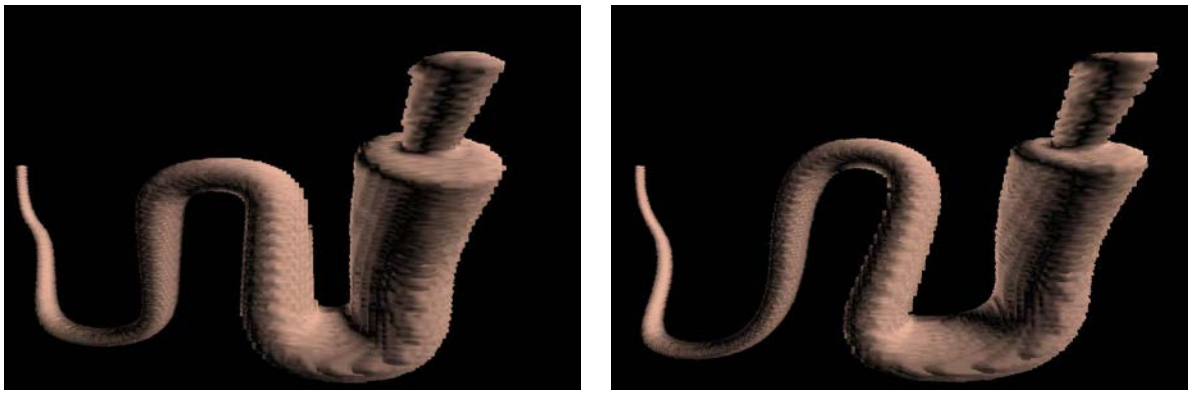


Figure 6.4: Curves generated with wide bounding boxes contain many areas of poor orientation.



(a) (b)

Figure 6.5: An example of how a curve can be customized by moving the 2D nodes. The original, automatically generated curve (a) has its control points moved in 2D to produce the resulting slanted curve (b). Both curves use the same depth assignment when generating the final 3D curve.

bounding box while maintaining the bounding constraint for the rest of the curve.

A particularly smooth, snaking curve is shown in Figure 6.7. Based on our criteria this curve is exemplar of a desirable curve. It (1) is a snaking path, (2) fully sweeps out the bounding area, and (3) has good orientation along the length of the curve. There are times when a similar path is desirable but modifications must be made to adhere to layout, for example, Figure 6.8. Using our approach, the path of the curve can be altered to accomodate labels or insets by excluding control points from the path finding process. This exclusion can be as effective as defining a more complicated bounding area. Figure 6.10 demonstrates how excluding a single control point in the lower right corner of a two-by-three grid changes the shape of the bounding area and resulting path and Figure 6.9 shows a visualization created using the same curve as in Figure 6.7 but with the path with the excluded point in Figure 6.10.

6.2.1 Curve Evaluation

Almost all of the presented curves in Section 6.2 and Section 3.4 demonstrate that our approach creates a variety of snaking curves based on different inputs. Each of these snaking

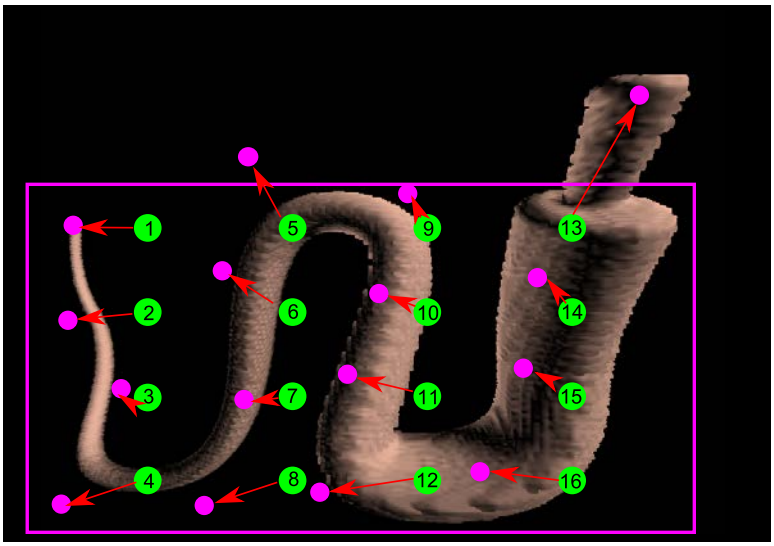


Figure 6.6: The positions of the original 16 control points (green) are moved in the direction of the arrows (red) to a final position (magenta) in order to produce the curve shown in Figure 6.5.

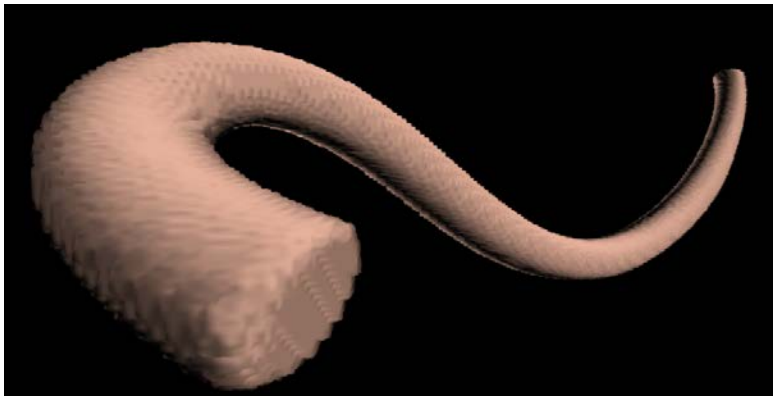


Figure 6.7: A particularly smooth, snaking curve produced using our approach.

Figure 6.8: A final visualization may require insets and labels in specific locations, thus, the generated curve may not be desirable for the purposes of final visualization layout.

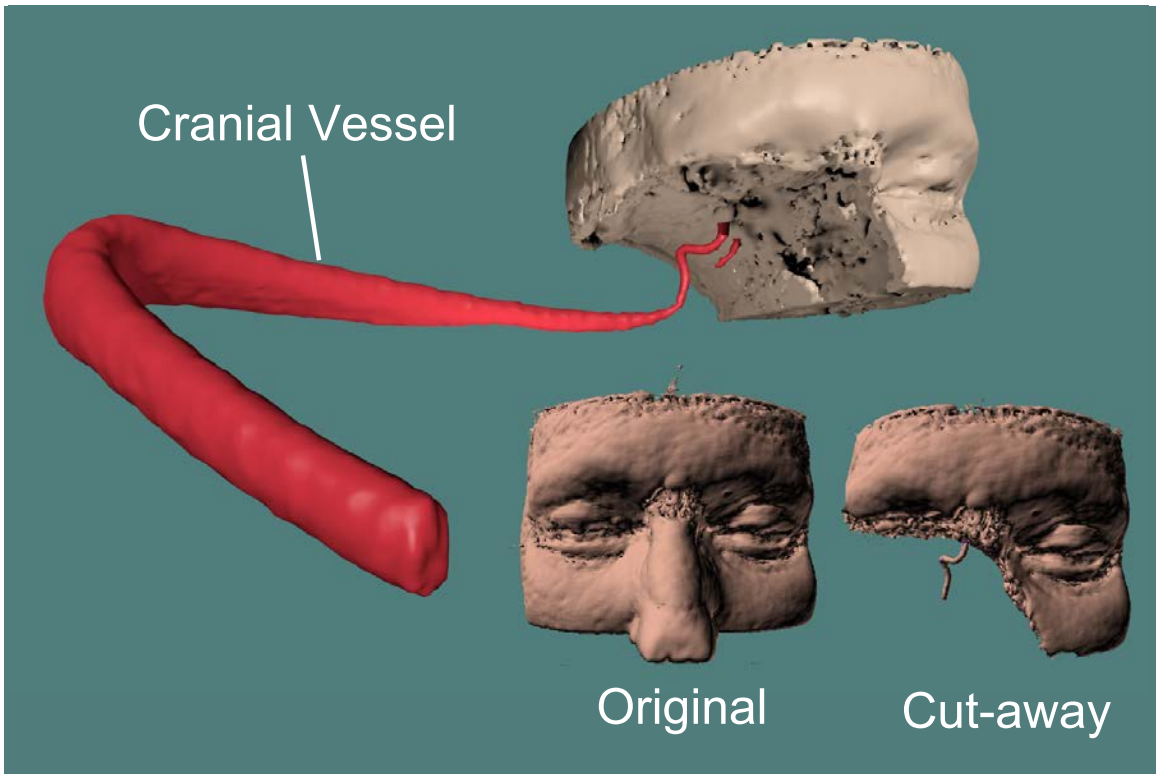


Figure 6.9: A fully-featured visualization showing a CT scan of a human head as the context, a segmented portion of an angiography of a cranial vessel as the focus, and the original dataset shown before and after a cut-away. The path of the cranial vessel towards the viewpoint is generated using the same configuration as in Figure 6.8 with some nodes excluded from the pathfinding. (See Figure 6.10.)

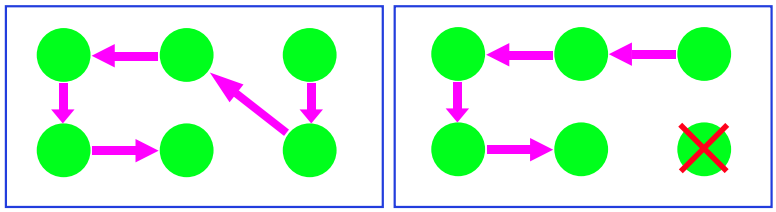


Figure 6.10: The difference between the automatic pathfinding when the lower right node is included (left) versus excluded (right). The curve generated before excluding the node is shown in Figure 6.7 while a visualization using a curve generated by excluding the lower right node is shown in Figure 6.9.

curves adheres to our evaluation criteria of sweeping out the designated bounding space. However, some of the curves have undesirable orientation, according to our criteria, and are not oriented towards the viewpoint along the majority of the curve. One such curve is shown in Figure 6.4.

Figure 6.9 demonstrates that if the path is modified by excluding some nodes from the automatic path finding that the resulting curve may not be a snaking curve. According to our criteria this is undesirable but it does enable extra control when it may be necessary to modify a curve to accomodate strict layouts including insets and/or labels.

6.3 A Visualization from a Textured Snaking Curve

A large portion of this thesis involves utilizing acquired volume data to create a model along the snaking curves for focus+context. However, we have created a visualization similar to a motivating image to demonstrate that it is possible to use only the curves themselves to create meaningful visualizations.

An example of a visualization created in the likeness of the motivating image in Figure 1.1 is shown in Figure 6.11. A generalized cylinder was used to create the 3D model that follows the generated snaking curve. Extra editing in the form of coloring or texturing the resulting model is required to produce informative visualizations because the generalized cylinders contain no information about any structures that would be present in volume data. However, an artist can use the generalized cylinder as a template for the path of the structure through space and create a model with more detail or add texture, which is demonstrated here.

6.4 Generated Visualizations

To demonstrate the versatility of our approach we have generated several visualizations from cross sections of objects (Section 6.4.1) using data created via image-swept volume [80] and from acquired volume data (Section 6.4.2) using our deformation approach described in

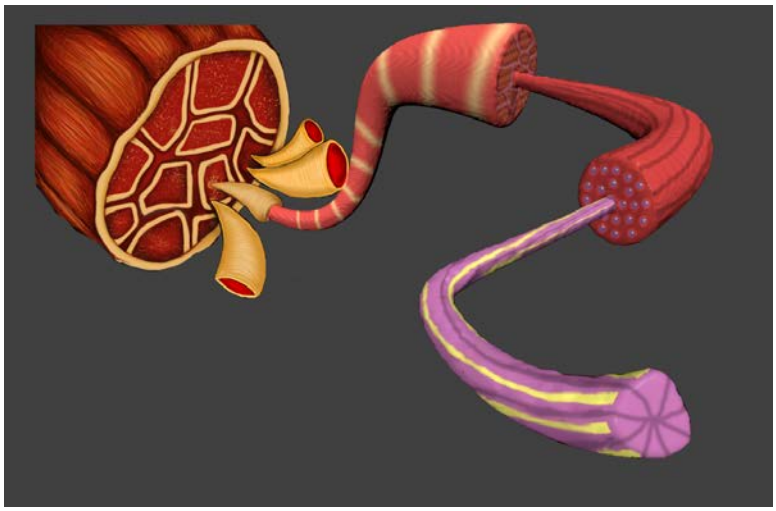


Figure 6.11: A visualization of the structure of muscle tissue created in likeness to the motivating image in Figure 1.1. The 3D model along the snaking curve was textured in Blender [37] after it was generated using our approach. The rendering is overlaid onto the muscle from the motivating image in Figure 1.1 for context.

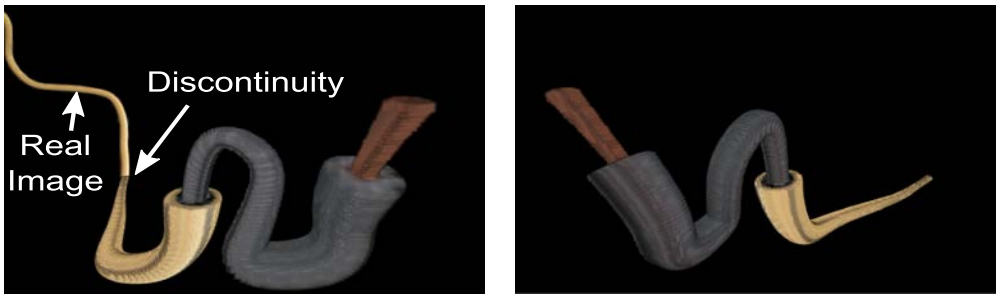
Chapter 4.

6.4.1 Visualizations of Objects Created via Swept Volume

The set of objects from which visualizations are created includes a coaxial cable, a stained slide of a cross section of a plant (Hypericum perforatum) stem, and fossilized wood. Each visualization in this section was generated by sweeping image templates to produce image-swept volumes [80] along the trajectory of the snaking curve.

A visualization of a coaxial cable, from two different viewpoints, is shown in Figure 6.12. The data for this volume was created from an image of the cross section of a coaxial cable as shown in Figure 6.14. Figure 6.12(a) shows the original view of the visualization overlaid onto a background containing an image of a real coaxial cable, while Figure 6.12(b) shows the same generated visualization rotated to view the side of the model, without the background image.

Figure 6.12 shows three different sections of the cable. From the left to right side of the image these are the exterior insulation, the interior insulation, and the copper wire.



(a) (b)

Figure 6.12: A visualization of a coaxial cable produced using our process. The original visualization with an image of a real coaxial cable is shown (a). A rotated view of the visulization is shown in (b), without the image of a real coaxial cable. The same process to generate Figure 6.5 were used to generate this result.

Each layer has its exterior portrayed along the snaking curve and the focal cross sections demonstrate the size of each successive layer compared to the previous. The side view of the coaxial cable (Figure 6.12(b)) reveals a noticeable change in how the curve travels through space near the copper wire. This is due to the specified end tangent to force the copper portion to be oriented more towards the viewpoint. The same curve without a forced end tangent is shown in Figure 6.5(b).

Figure 6.13 shows a different visualization of a coaxial cable. An additional section (black), corresponding to how the metallic shield appears in the original image of the cross section, is added to the visualization. The path of the curve in Figure 6.13 is the direct result of our automatic curve creation approach and required no user intervention to create; only the bounding box and positions of the terminating focal points of the sections along the generated snaking curve were specified.

Figure 6.15 shows a visualization created from a stained cross section of a plant stem overlaid on an image of the stem[2]. This visualization uses the example 2D path as shown in Figure 3.4(b), demonstrating how that path appears in a final visualization. An illustrative effect is created because a plant stem normally would not bend in such a way. The original

[2]Hypericum perforatum (St. John's Wort) cut stem courtesy of Bob Osborn `wbdpublications.co.uk`. Used with permission.

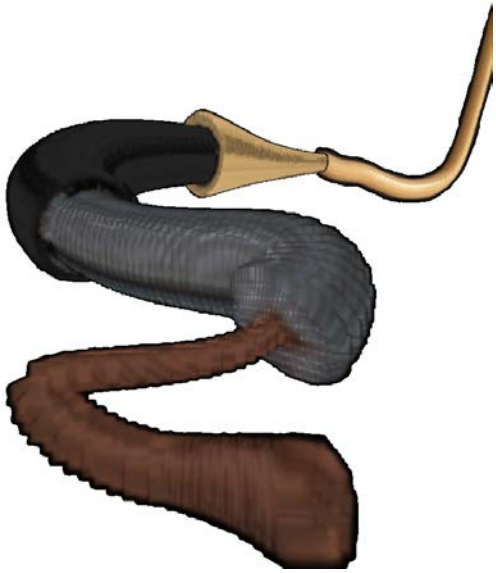


Figure 6.13: Another visualization of a coaxial cable created with our process, this time with more layers shown. Our model is overlaid on an image of a real coaxial cable (top-right). The volume data for this visualization was created via swept volume from an image taken by the author.

Figure 6.14: The image of the coaxial cable cross section used to create the volume data for the coaxial cable visualizations. We see the outer, tan segment at the rear of Figure 6.13, followed by the black segment, grey segment, and inner copper wire as we travel along the path towards the closest portion. Image created by the author.

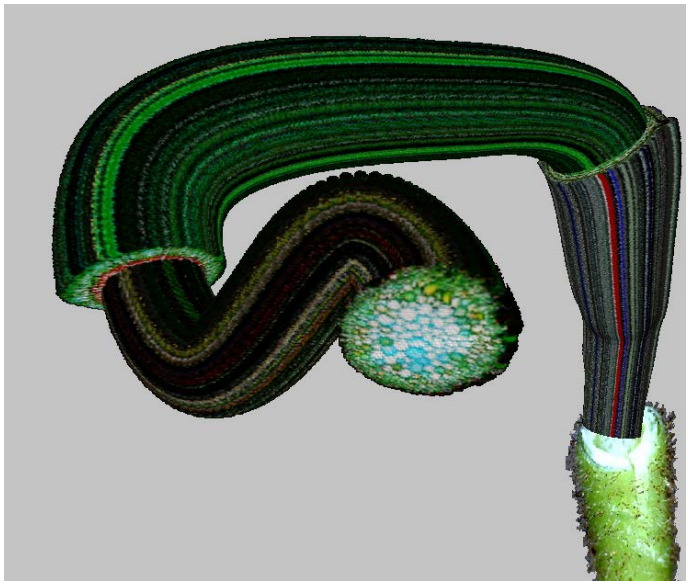


Figure 6.15: Visualization from a magnified, stained cross section of the stem of a Hypericum perforatum (St. John's Wort) overlaid on an image of a cut stem of the same plant. The snaking curve for this visualization was produced using the setup shown in Figure 3.4(b).

data for the visulization of the plant stem comes from an image of a stained cross section of a Hypericum perforatum (St. John's Wort)[3] which is shown in Figure 6.17.

A problem with this visualization is the way the closest portion of the snaking curve (center of the image) appears. The curve in that location contains too sharp of a turn. Most of the curves we have seen so far contain corners that are less sharp unless they were modified. Recall that the curves in this section have been generated with logarithmically spaced control points. A result of the logarithmic spacing is that points on the closest portion of the curves are closer together than on the rest of the curve. This is causing the sharp turn in Figure 6.15 which demonstrates a weakness of using logarithmic depth steps. The solution is to manually change the depth steps between points or use uniform spacing, as demonstrated in Figure 6.16. However, using uniform spacing causes the furthest focal point to lose its orientation towards the viewpoint.

The final visualization in this section is created from a cross section of a fossil of a

[3]Courtesy of RolfDieterMueller licensed under Creative Commons 3.0 `http://creativecommons.org/licenses/by/3.0`.

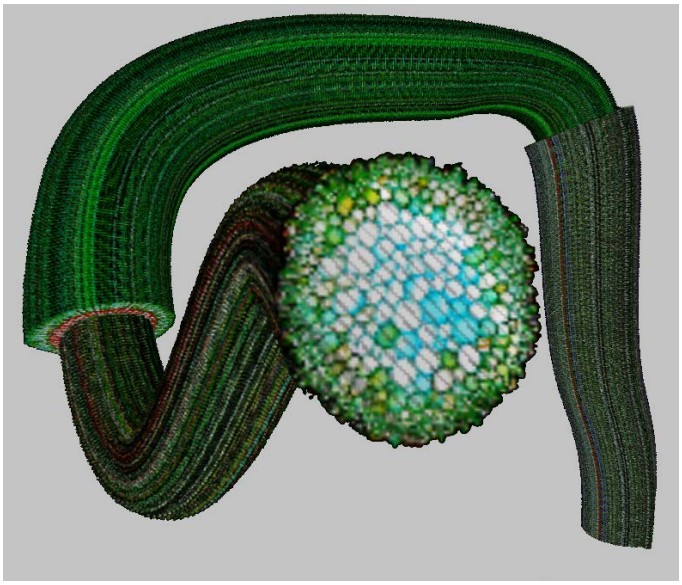
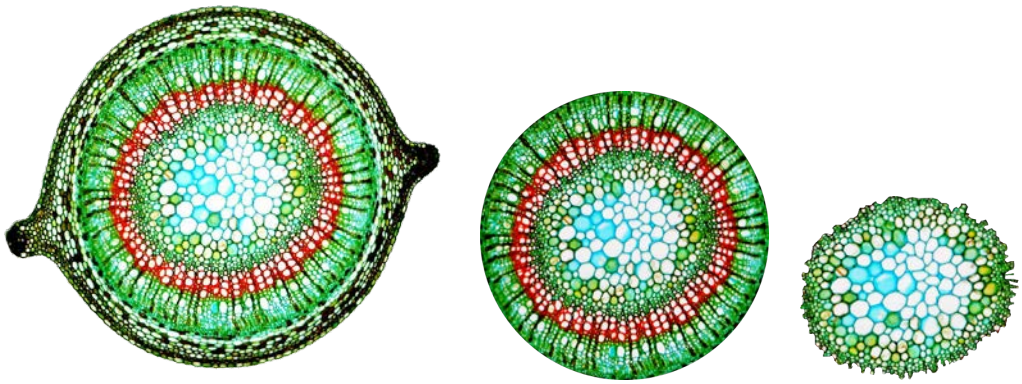


Figure 6.16: This visualization uses the same curve path as in Figure 6.15 but using uniformly spaced control points (in depth) instead of logarithmically spaced points. While the uniform spacing mitigates the sharp turn near the last focal point, the furthest focal point is no longer oriented towards the viewpoint.



(a) (b) (c)

Figure 6.17: The original stained cross sections of a Hypericum perforatum (St. John's Wort) stem that were swept to create the volumes used in Figure 6.15. Images used under the Creative Commons 3.0 `http://creativecommons.org/licenses/by/3.0`, courtesy of RolfDieterMueller.

Figure 6.18: A visualization created from a cross section of a fossil of a prehistoric tree (Hermanophyton). Cross section images can be found in Figure 6.20. The goal of this visualization is to bring the inner black portion of the fossil into focus in front of the viewpoint while providing a view of the context (original fossil) in the lower left corner.

prehistoric tree (Hermanophyton) and is shown in Figure 6.18. The images of the cross section used to generate the model for this visualization can be found in Figure 6.20. The goal of this visualization is to bring the inner black portion of the fossil into focus in front of the viewpoint.

The visualization in Figure 6.18 contains no manual modifications. The snaking curve as shown is exactly what is generated by the approach presented in this thesis. A large depth change was used to create this visualization. A smaller depth change of half the magnitude was first used to generate the snaking curve, as shown in Figure 6.19. However, the curve with smaller depth change does not appear to be travelling along a nice snaking path through space and instead appears twisted and contorted. Figure 6.19 provides an example of an visualization that does not turn out with a specified depth change, thereby requiring a larger depth change to produce a nicer visualization (Figure 6.18).

The visualization of the fossilized tree in Figure 6.18 is unrealistic; neither a tree or, especially, a fossilized tree, would bend in such a snaking curve as shown. It does, however,

Figure 6.19: An example of how a bad choice for the total depth change of the snaking curve can affect the final model. The curve does not appear to be travelling along a nice snaking path through space and instead appears contorted. The focal cross sections were moved slightly so that they did not lie on the twisted portion of the generated curve.

(a) (b) (c)

Figure 6.20: The original fossil cross sections used to create the volume used in Figure 6.18. Images courtesy of Mike Viney and the Virtual Petrified Wood Museum [74]. Used with permission.

aid in evaluating the idea of using a snaking curve for focus+context visualizations. The stretched, black portion of the fossil appears as though it is ready to be slid back into place inside of the previous layer. The same could be said of the previous layer. This visualization is very successful in demonstrating the effectiveness of focus+context via snaking paths because of its natural feeling and because it enables the visualization of the exact location of the origin of each layer.

In all of the visualizations presented in this section the generated snaking curve appears to travel a large distance towards the viewpoint, enhancing the effects of magnification. This property helps to achieve the desired illustrative effect as demonstrated by the motivating example images.

6.4.2 Visualizations of Acquired Volume Data

The visualizations produced in this section utilize acquired volume data, for both the context and along the generated snaking curve. The approach for volume deformation, presented in Chapter 4, is used to locate and deform the original volume data along the generated snaking curve. All visualizations presented in this section use logarithmic depth-steps for the automatic depth assignment to produce the final 3D curve as described in Section 3.3. Logarithmic spacing is chosen over uniform spacing because logarithmic spacing produces a curve that is oriented more towards the viewpoint along the entire length (see Section 6.2).

The datasets used to create the visualizations presented in this section include (1) the stented, human aorta dataset[4], and (2) an angiography of the human head[5]. Both datasets are pre-segmented. The goal is to create a visualization involving the aorta near the heart using the stented aorta dataset and a visualization of one of the cranial vessels near the base of the skull using the angiography dataset.

Figure 6.21 shows one level of focus+context created from the aorta dataset. To create

[4]The stented aorta dataset is courtesy of Michael Meißner, Viatronix Inc., USA via www.volvis.org.

[5]The angiography dataset is courtesy of Özlem Gürvit, Institute for Neuroradiology, Frankfurt, Germany and its segmentation courtesy of Dirk Bartz, VCM, University of Tübingen, Germany (via www.volvis.org).

this visualization our proof-of-concept interaction described in Section 5.3 was used. The result of sculpting using the constrained point radiation sculpting tools described in Chapter 5 is shown in Figure 6.22. The aorta was selected as the structure of interest and a portion near the heart was designated for use in the visualization.

The bounding box confining the snaking curve and a curve structure involving two 180° turns (three-by-three grid of nodes) were specified for the automatic snaking curve generation. The start tangent was set such that the resulting snaking curve follows the orginal dataset at its beginning. The end tangent of the snaking curve is a result of logarithmic-based depth steps between control points. This visualization uses only one focal section in the curve because data that differentiates structures within the aorta is not available. For an example of multiple focal sections with the same dataset, see the result in Figure 6.23, although that figure utilizes an edited geometric model for the last focal section.

There is a minor discontinuity in Figure 6.21 between the original volume and the generated curve due to two different volumes being rendered: the original, and the curve. This is a problem with rendering two overlapped, discontinuous volumes and not our approach to snaking curve generation or volume deformation process. However, it could be beneficial to explore a deformation technique that, instead of creating a separate deformed volume for the snaking curve, deforms the structure directly from the original volume model.

A major component that is missing from Figure 6.21 is the multiple levels of focus+context. The volume data does not contain the detail required to differentiate between interior structures of the aorta because the values corresponding to the aorta lie within a small range. To mitigate this problem, Figure 6.23 is generated using our hybrid approach involving part deformed volume data and part generated geometric models. This figure is created with a surface model along the last portion of the curve. The surface model was exported and edited in Blender [37] to edited to include red blood cells to depict a microscopic view of oxygenated blood flowing through the aorta.

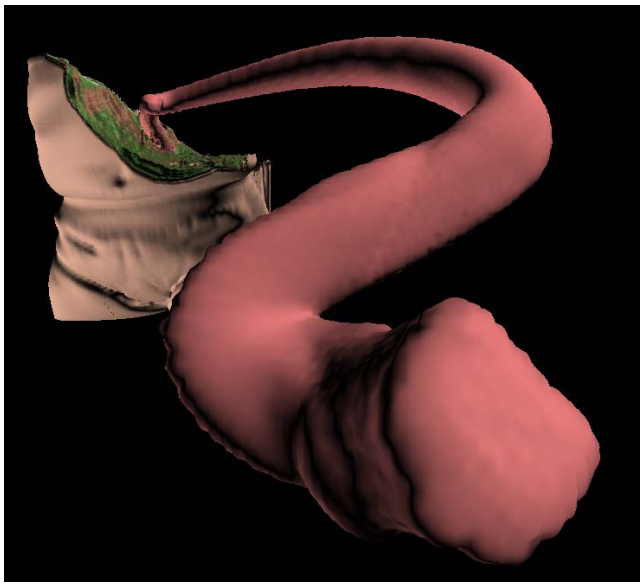


Figure 6.21: One level of focus+context created using the stented aorta dataset. The end of the aorta, which is closest to the heart in the original dataset, is deformed along the snaking curve causing it to stretch out. However, most of its shape is retained allowing a viewer to get a sense of the shape of the aorta at its position in the original context.

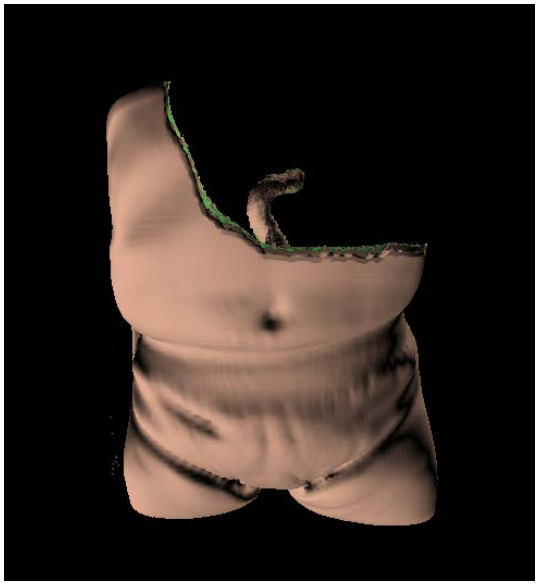


Figure 6.22: A close-up visualization of the original aorta dataset. Part of the dataset has been sculpted to remove the occluding material around the aorta so that this model can be used as the context for Figure 6.21.

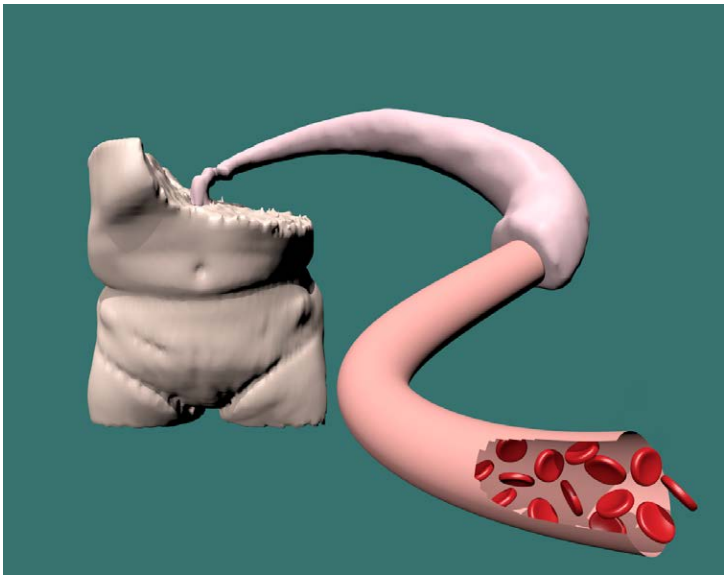


Figure 6.23: The volume model of the torso is of a lower resolution in this visualization, while the aorta is deformed over first half of the curve and is a higher resolution than the volume model in the background. A surface model is created along the last portion of the snaking curve, closest to the viewpoint. The red blood cells were added afterwards and the entire visualization was rendered in Blender [37].

Figure 6.23, containing the edited surface model proves that although the original volume model does not contain enough detail to differentiate microscopic structures, our appraoch enables quick preliminary modelling. Additional details and structures that can not be captured with volume data can be completed by an artist, using the generated surface model as a guide. This saves the artist the time of creating a model along the entire snaking path and they can instead focus on the creation of the structures.

The final visualization presented in this section is shown in Figure 6.24 and involves the angiography dataset. A cranial vessel from the angiography dataset is brought into focus along a snaking curve containing only one turn. The curve was generated using a two-by-three grid of nodes in the snaking curve generation but one of the nodes was excluded. To accomodate labels and insets in the final visualization, the snaking shape of the resulting curve was sacrificed (see Section 6.2 for more information). Isosurface models of the human head and the deformed vessel along the snaking path were rendered in Blender [37] to create a high-quality visualization with lighting.

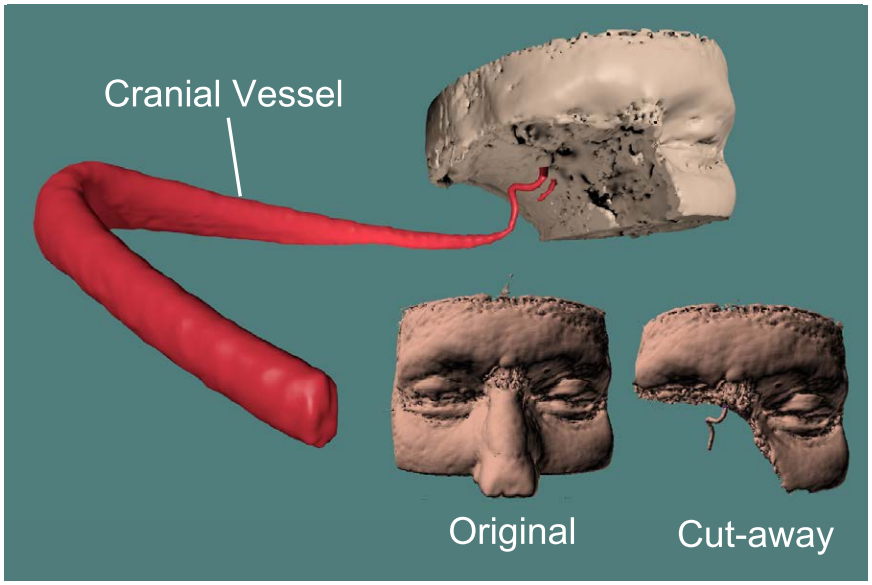


Figure 6.24: A cranial vessel from the angiography dataset is brought into focus along a snaking curve containing only one switchback. The path of the original generated curve has been altered via our methods to accomodate labels and insets in order to produce the final visualization (See Section 6.2).

At first, the visualization created from the angiography data seems lacking because there are no parts containing extra details such as blood cells or vascular structure. However, the visualization demonstrates a magnification of a single vessel. A magnification could be achieved using a lens technique, however, a lens technique would distort the surrounding context and only provide a view of the structure from one side. In Figure 6.24, all sides of the cranial vessel are visible because of the snaking curve and position of the viewpoint. That is, along the further portion of the snaking curve, nearest the head, the underside and front of the structure are visible. The top and back of the structure are visible in the nearest section of the snaking curve, after the corner. Thus, the visualization has successfully magnified the structure of interest and enabled the visualization of all its parts with no discontinuity. Partial distortion may result from the deformation as a result of stretching the cranial vessel along the snaking curve but overall shape is preserved.[6]

[6]It is important to note that our evaluation of successful relies on the qualitative criteria set out in Section 6.1 and the qualitative effect of the magnification and has not been evaluated by a medical specialist.

6.5 Limitations

Several limitations of the approach presented in this thesis have been made apparent in presenting the generated visualizations.

First, there is no control over the depth at which control points will appear because of the automatic depth assignment. As demonstrated by several examples in Section 6.4.1 this type of control would be the most helpful at the end of the snaking curve, closest to the viewpoint. Having better control over the depth assignment would mitigate the weaknesses of both the uniform and logarithmic spacing approaches. However, our approach was demonstrated to achieve the shapes of the snaking curves with desirable orientations along most of the curve despite that fine-grained control over the depth is missing.

A second limitation is the view-dependent nature of the resulting curve. This is apparent in Figure 6.12(b) where, when viewed from a different viewpoint than that used for the generation, the curve appears less desirable and it is not apparent that it is a snaking path. However, recall that our curve generation approach is view-dependent, thus, we do not assume that the snaking curve will appear as visually appealing from other viewpoints. This limits the aesthetics of our approach to static visualizations. But, there could still be a benefit in exploring the curve from alternate viewpoints such as for sculpting to further discover the interior structures.

The final limitation is that surface detail is lost due to the deformation. The approach works well for objects that have no fine texture or surface structure. However, those with more detailed surface textures would be stretched and deformed, especially if the length of the curve (thus, the amount of deformation) is large, thereby losing any information of the surface texture as shown in Figure 4.11. This is mitigated by using successively higher resolutions of the structure of interest along each focal section of the curve as mentioned in Section 4.2 but the amount of preserved detail is then limited by the highest resolution.

Chapter 7

Conclusion

In this thesis we have presented an approach to create continuous focus+context illustrations consisting of multiple-levels of focus+context. These depict a structure travelling along a snaking path, similar to Figure 1.1. Our approach enables the automatic creation of 3D snaking curves with a simplified 2D interface. A 2D bounding rectangle is automatically filled with a snaking curve. Two approaches to automatically extrude the resulting 2D curve into 3D are presented. Acquired volume data can be deformed to follow the generated snaking curve, thereby, enabling the creation of focus+context illustrations from data of real structures, such as acquired medical data. When acquired volume data lacks the necessary detail, we allow the creation of a geometric model of a cylinder to be created along a portion of the curve, which acts as a template for an artist to build upon.

Our results demonstrate that our apporach helps us to generate illustrations in likeness to the motivating images where a particular structure and several of its interior layers, bringing it into focus in front of the viewpoint. The resulting visualization and 3D model can be further explored, by sculpting or viewpoint changes, adding value to our approach. The resulting models can also be visualized using other focus+context methods.

We were unable to demonstrate an illustration involving a very large scale (highly detailed) generated from the acquired volume data that was availble to us. However, the models generated by sweeping image templates, combined with our visualizations of a human aorta using our hybrid volumetric-geometric approach, can mitigate the missing detail. Extra detail can be created after generating a template geometric model. Using our approach with volume data that contains well-differentiated layers would help to more closesly match the desired effect. Thus, our approach should work well in the future as data with the required

fidelity is made available.

We have demonstrated that we can also create geometric models that follow the trajectory of the generated snaking curve. However, if a geometric model already exists that an artist wishes to deform along the generated snaking curve then, although, not implemented in this thesis, our deformation process can be applied to geometric data. The deformation process would slightly change because geometry is being deformed instead of volume data. This process could involve finding the correct transformation matrices and simply transforming each vertex corresponding to a certain area in the model.

7.1 Snaking Curve Generation

Our automatic snaking curve generation approach creates snaking curves in likeness to the curves used for focus+context in the motivating illustrations in Figure 1.1 and 1.2. Most of the curves generated as part of the work in this thesis adhered to our three criteria of having a snaking path through space, sweeping out a provided bounding area, and having an tangents oriented mostly towards the viewpoint.

Generating the snaking curves automatically requires a method of automatically splitting a defined bounding area into a set of nodes through which a snaking path can be easily found. This constraint has the drawback of restricting the 2D bounding area to rectangular regions, or at the very most, an area built from smaller rectangular regions that can easily be tessellated into a regular grid-like structure. We were able to mitigate this problem by allowing some of the nodes controlling the automatic path finding to be excluded from the path finding. A large grid of control points can be created and specified points can be excluded until the desired shape or path are achieved.

The automatic 3D curve generation provides a method for modelling 3D curves by specifying 2D constraints. A weakness of the automatic method is the lack of depth control. A 2D snaking curve may at first appear to be the correct shape but the extra third dimension has

a profound effect on the overall shape of the curve. The automatic depth assignment produced desirably oriented curves under most conditions, which include varying bounding box shapes, using a moderately large total depth change. However, we observed cases where the automatic depth assignment failed to produce desirable results such as when the total depth change is too small. This is demonstrated by the curves sharp corners near the viewpoint such as in Figure 6.15 or contorted curves such as in Figure 6.19. Changing the overall depth change, or length of the curve through space, mitigated the problem (compare Figure 6.19 to Figure 6.18). In order to fix the problem in other visualizations (Figure 6.15), better control over the depth assignment would be required, or other automatic depth assignment methods should be investigated.

7.2 Focus+Context Visualizations via Snaking Paths

Where visualizations created from only volume data[1] lacked the fine detail necessary to visualize smaller structures, especially microscopic structures, the hybrid volumetric-geometric approach exceled. Although the hybrid approach also initially lacked the necessary detail, it enables an artist to focus on the creation of detailed structures without having to also create the initial curve. The preliminary model acts as a guide that an artist may modify and build upon to add the necessary detail.

Visualizations created from only acquired volume data still successfully mimicked the motivating illustrations, although, without multiple levels of focus+context. The effect created, instead, is a magnification of a structure of interest along a snaking path. Viewing the structure of interest along the snaking path enables the structure to be viewed from all sides while it travels along the snaking curve. Compared to a simpler lens-based magnification this gives a clearer sense of all parts of the structure, even though the model is generated in a view-dependent manner.

[1] That was available to us.

A limitation of our approach is that deformation of the structure of interest causes lost surface details. We have mitigated the loss of surface detail by deforming small parts of the structure of interest from higher resolution volume data as described in Section 4.2.

7.3 Future Work

We found that sometimes volume data does not contain enough detail to achieve results with many levels of focus+context. To mitigate this we added a hybrid approach to create a preliminary geometric model over a portion of the curve. The geometric model then acts as a template for an artist to create a more detailed model. An improvement to this is to provide a mechanism for geometric deformation so that our approach can also be used with existing geometric models in the same way it can be used with volume models.

Since we constrained our approach to generating snaking curves for use in visualization, we never explored the use of bounding structures more complicated than rectangular regions. One goal was to restrict the resulting visualization to an area on the screen to enable the placement of other elements such as labels around the final visualization. Rectangular bounding boxes were initially chosen because most illustrations are bound by a rectangular region and rectangular regions help provide a definition for a snaking curve to fill the designated space. In a rectangular region, the curve must travel between sides before switching back on itself in order to fill the entire space with a desirable curve (instead of filling the space with small zig-zags). With more complicated bounding structures this definition may have to change. It may be worth investigating a definition that would allow more complicated bounding structures to be specified and an associated automatic tessellation that will enable the automatic path finding to create snaking curves in the new tessellation.

An excellent further expansion to the idea of the snaking path and the visualization of structures along the path involves supporting branching structures. One can imagine that it would be beneficial to be able to create visualizations of the same style from branching

structures to illustrate each branch along a separate snaking path. For example, the motivating illustration in Figure 1.2 shows a nerve that contains branches at its endpoint. These branches could be further illustrated along snaking paths to enhance the visualization. Two problems would have to be investigated as part of this future work: (1) how to find branching snaking paths and lay them out in space, and (2) how to locate and deform branching structures of interest along the branching snaking path.

Our approach also enables 2D modelling of 3D curves because we have automated the move to 3D. A natural future work is to integrate our approach into a sketch-based modelling (SBM) system for volumetric data. An interesting investigation is whether or not our approach helps reduce the necessity of annotations that are frequently required in SBM.

Part of an investigation into SBM of curves involves gaining better overall control of the generated curve. This would include arbitrary placement of all control points in 2D but also requires full depth control. It would be very tempting to provide the placement of points from another viewpoint but existing sketch-based curve modelling systems [8][2][22] already allow this. Furthermore, sketching from two different viewpoints may not be necessary if a more semi-automatic approach akin to our depth-based assignment exists. Perhaps, after an initial automatic depth assignment, the control points can be moved in 3D, along the depth-dimension, using the same modelling methods available in modelling packages such as Blender [37] or Maya [7].

Bibliography

[1] Alok Aggarwal, Don Coppersmith, Sanjeev Khanna, Rajeev Motwani, and Baruch Schieber. The angular-metric traveling salesman problem. In *Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms*, SODA '97, pages 221–229, Philadelphia, PA, USA, 1997. Society for Industrial and Applied Mathematics.

[2] James Andrews, Pushkar Joshi, and Nathan Carr. A linear variational system for modelling from curves. *Computer Graphics Forum*, 30(6):1850–1861, 2011.

[3] David Applegate, Robert Bixby, Vasek Chvatal, and William Cook. On the solution of traveling salesman problems. In *Documenta Mathematica Extra Volume ICM III*, pages 645–656. Rheinische Friedrich-Wilhelms-Universität Bonn, 1998.

[4] Esther M. Arkin, Michael A. Bender, Erik D. Demaine, Sándor P. Fekete, Joseph S. B. Mitchell, and Saurabh Sethia. Optimal covering tours with turn costs. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, SODA '01, pages 138–147, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics.

[5] Arjang A. Assad and Bruce L. Golden. Chapter 5 arc routing methods and applications. In C.L. Monma M.O. Ball, T.L. Magnanti and G.L. Nemhauser, editors, *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, pages 375 – 483. Elsevier, 1995.

[6] Autodesk. 3ds max, 2012. http://www.autodesk.com/products/autodesk-3ds-max/overview.

[7] Autodesk. Maya, 2012. http://www.autodesk.com/products/autodesk-maya/overview.

[8] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. Ilovesketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st*

annual ACM symposium on User interface software and technology, UIST '08, pages 151–160, New York, NY, USA, 2008. ACM.

[9] Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *J. ACM*, 9(1):61–63, January 1962.

[10] Jules Bloomenthal. Graphics gems. chapter Calculation of reference frames along a space curve, pages 567–571. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[11] S. Bruckner and M.E. Groller. Volumeshop: an interactive system for direct volume illustration. In *Visualization, 2005. VIS 05. IEEE*, pages 671 – 678, oct. 2005.

[12] Stefan Bruckner, Sören Grimm, Armin Kanitsar, and Meister Eduard Gröller. Illustrative context-preserving volume rendering. In *Proceedings of EuroVis 2005*, pages 69–76, May 2005.

[13] Stefan Bruckner and M. Eduard Groller. Exploded views for volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1077–1084, September 2006.

[14] Thomas Brunet, K. Evan Nowak, and Michael Gleicher. Integrating dynamic deformations into interactive volume visualization. In *Proceedings of the Eurographics/IEEE Symposium on Visualization*. Eurographics, IEEE, 2006.

[15] Michael Burns, Martin Haidacher, Wolfgang Wein, Ivan Viola, and Eduard Groeller. Feature emphasis and contextual cutaways for multimodal medical visualization. In *Eurographics / IEEE VGTC Symposium on Visualization 2007*, pages 275–282, May 2007.

[16] M. S. T. Carpendale and Catherine Montagnese. A framework for unifying presentation space. In *Proceedings of the 14th annual ACM symposium on User interface software*

and technology, UIST '01, pages 61–70, New York, NY, USA, 2001. ACM.

[17] Haixin Chen, Jrgen Hesser, and Reinhard Mnner. Ray casting free-form deformed-volume objects. *The Journal of Visualization and Computer Animation*, 14(2):61–72, 2003.

[18] Hungi-Li Jason Chen. Interactive volume manipulation. Master's thesis, University of Calgary, 2008.

[19] Hungi-Li Jason Chen, Faramarz F. Samavati, and Mario Costa Sousa. Gpu-based point radiation for interactive volume sculpting and segmentation. *Vis. Comput.*, 24:689–698, July 2008.

[20] Jen-Hui Chuang, Chi-Hao Tsai, and Min-Chi Ko. Skeletonisation of three-dimensional object using generalized potential field. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1241 – 1251, nov 2000.

[21] Jean-Maurice Clochard and Denis Naddef. Using path inequalities in a branch and cut code for the symmetric traveling salesman problem. In Giovanni Rinaldi and Laurence A. Wolsey, editors, *IPCO*, pages 291–311. CIACO, 1993.

[22] Jonathan M. Cohen, Lee Markosian, Robert C. Zeleznik, John F. Hughes, and Ronen Barzel. An interface for sketching 3d curves. In *Proceedings of the 1999 symposium on Interactive 3D graphics*, I3D '99, pages 17–21, New York, NY, USA, 1999. ACM.

[23] M. Cohen. *Focus and Context for Volume Visualization*. PhD thesis, 2006.

[24] M. Cohen and K. Brodlie. Focus and context for volume visualization. In *Theory and Practice of Computer Graphics, 2004. Proceedings*, pages 32 –39, june 2004.

[25] N.D. Cornea, D. Silver, and P. Min. Curve-skeleton applications. In *Visualization, 2005. VIS 05. IEEE*, pages 95 – 102, oct. 2005.

[26] N.D. Cornea, D. Silver, and P. Min. Curve-skeleton properties, applications, and algorithms. *Visualization and Computer Graphics, IEEE Transactions on*, 13(3):530 –548, may-june 2007.

[27] Nicu D. Cornea, Deborah Silver, Xiaosong Yuan, and Raman Balasubramanian. Computing hierarchical curve-skeletons of 3d objects. *The Visual Computer*, 21:945–955, 2005. 10.1007/s00371-005-0308-0.

[28] Carlos Correa, Deborah Silver, and Min Chen. Feature aligned volume manipulation for illustration and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1069–1076, September 2006.

[29] Carlos D. Correa, Deborah Silver, and Min Chen. Discontinuous displacement mapping for volume graphics. In *Eurographics/IEEE VGTC Workshop on Volume Graphics*, pages 9–16. Eurographics Association, 2006.

[30] Carlos D. Correa, Deborah Silver, and Min Chen. Technical section: Constrained illustrative volume deformation. *Comput. Graph.*, 34(4):370–377, August 2010.

[31] C.D. Correa and D. Silver. Dataset traversal with motion-controlled transfer functions. In *Visualization, 2005. VIS 05. IEEE*, pages 359 – 366, oct. 2005.

[32] Balázs Csébfalvi, Lukas Mroz, Helwig Hauser, Andreas König, and Eduard Gröller. Fast visualization of object contours by non-photorealistic volume rendering. *Computer Graphics Forum*, 20(3):452–460, 2001.

[33] G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large-scale traveling-salesman problem. *Operations Research*, 2:393–410, 1954.

[34] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959. 10.1007/BF01386390.

[35] David Ebert and Penny Rheingans. Volume illustration: non-photorealistic rendering of volume models. In *Proceedings of the conference on Visualization '00*, VIS '00, pages 195–202, Los Alamitos, CA, USA, 2000. IEEE Computer Society Press.

[36] Eric Ferley, Marie-Paule Cani, and Jean-Dominique Gascuel. Practical volumetric sculpting. *The Visual Computer*, 16:469–480, 2000. 10.1007/PL00007216.

[37] Stichting Blender Foundation. Blender, 2012. http://www.blender.org/.

[38] Sam Fraser-Smith. Frangipani rust on Plumeria rubra. `http://en.wikipedia.org/wiki/File:Frangipani_rust_(caused_by_Coleosporium_plumeriae)_on_Plumeria_rubra.jpg`, March 16 2010.

[39] N. Gagvani, D. Kenchammana-Hosekote, and D. Silver. Volume animation using the skeleton tree. In *Volume Visualization, 1998. IEEE Symposium on*, pages 47 –53, oct. 1998.

[40] Tinsley A. Galyean and John F. Hughes. Sculpting: an interactive volumetric modeling technique. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '91, pages 267–274, New York, NY, USA, 1991. ACM.

[41] T. Grigorishin, G. Abdel-Hamid, and Y. Yang. Skeletonisation: An electrostatic field-based approach. *Pattern Analysis & Applications*, 1:163–177, 1998. 10.1007/BF01259366.

[42] Andrew J. Hanson and Hui Ma. Parallel transport approach to curve framing. Technical report, Indiana University, 1995.

[43] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems, Science, and Cybernetics*, SSC-4(2):100–107, 1968.

[44] Helwig Hauser, Lukas Mroz, Gian-Italo Bischi, and M. Eduard Gröller. Two-level volume rendering. *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS*, 7:242–252, 2001.

[45] Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. In *Proceedings of the 1961 16th ACM national meeting*, ACM '61, pages 71.201–71.204, New York, NY, USA, 1961. ACM.

[46] H. Homann. Implementation of a 3d thinning algorithm. Oct. 2007. `http://hdl.handle.net/1926/1292`.

[47] Wei-Hsien Hsu, Kwan-Liu Ma, and Carlos Correa. A rendering framework for multiscale views of 3d models. *ACM Trans. Graph.*, 30(6):131:1–131:10, December 2011. URL `http://doi.acm.org/10.1145/2070781.2024165`.

[48] Keiho Imanishi, Megumi Nakao, Masahiko Kioka, Masato Mori, Munehito Yoshida, Takashi Takahashi, and Kotaro Minato. Interactive bone drilling using a 2d pointing device to support microendoscopic discectomy planning. *International Journal of Computer Assisted Radiology and Surgery*, 5:461–469, 2010. 10.1007/s11548-010-0413-z.

[49] Pushkar Joshi. Curve-based shape modeling - a tutorial. *IEEE Comput. Graph. Appl.*, 31(6):18–23, November 2011.

[50] Richard Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[51] Kitware. Visualization toolkit (vtk), 2012. Version 5.10.1. http://www.vtk.org.

[52] J. Kruger, J. Schneider, and R. Westermann. Clearview: An interactive context preserving hotspot visualization technique. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):941 –948, sept.-oct. 2006.

[53] Yair Kurzion and Roni Yagel. Space deformation using ray deflectors. In *6th Eurographics Workshop on Rendering 95*, pages 21–32, 1995.

[54] 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, 1994.

[55] Michael J. McGuffin, Liviu Tancau, and Ravin Balakrishnan. Using deformations for browsing volumetric data. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, VIS '03, pages 53–, Washington, DC, USA, 2003. IEEE Computer Society.

[56] R.P. Paul. *Robot Manipulators: Mathematics, Programming, and Control: The Computer Control of Robot Manipulators*. The MIT Press Series in Artificial Intelligence. Mit Press, 1981.

[57] Joseph F. Pekny and Donald L. Miller. A parallel branch and bound algorithm for solving large asymmetric traveling salesman problems. In *Proceedings of the 1990 ACM annual conference on Cooperation*, CSC '90, pages 56–62, New York, NY, USA, 1990. ACM.

[58] Less A. Piegl and Wayne Tiller. *The Nurbs Book*. Monographs in Visual Communication Series. Springer-Verlag GmbH, 1997.

[59] Emmanuel Pietriga and Caroline Appert. Sigma lenses: focus-context transitions combining space, time and translucence. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1343–1352, New York, NY, USA, 2008. ACM.

[60] Cyprien Pindat, Emmanuel Pietriga, Olivier Chapuis, and Claude Puech. Jellylens: content-aware adaptive lenses. In *Proceedings of the 25th annual ACM symposium on*

User interface software and technology, UIST '12, pages 261–270, New York, NY, USA, 2012. ACM.

[61] C. Rezk-Salama, M. Scheuering, G. Soza, and G. Greiner. Fast volumetric deformation on general purpose hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, HWWS '01, pages 17–24, New York, NY, USA, 2001. ACM.

[62] Mike Roberts, Jeff Packer, Mario Costa Sousa, and Joseph Ross Mitchell. A work-efficient gpu algorithm for level set segmentation. In *Proceedings of the Conference on High Performance Graphics*, HPG '10, pages 123–132, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.

[63] George G. Robertson and Jock D. Mackinlay. The document lens. In *Proceedings of the 6th annual ACM symposium on User interface software and technology*, UIST '93, pages 101–108, New York, NY, USA, 1993. ACM.

[64] F.F. Samavati, R.H. Bartels, and L. Olsen. *Local B-spline Multiresolution with Examples in Iris Synthesis and Volumetric Rendering.* 2007.

[65] Ryan Schmidt, Azam Khan, Karan Singh, and Gord Kurtenbach. Analytic drawing of 3d scaffolds. In *ACM SIGGRAPH Asia 2009 papers*, SIGGRAPH Asia '09, pages 149:1–149:10, New York, NY, USA, 2009. ACM.

[66] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '86, pages 151–160, New York, NY, USA, 1986. ACM.

[67] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, ACM '68, pages 517–524, New York, NY, USA, 1968. ACM.

[68] T. Taerum, Mario Costa Sousa, Faramarz F. Samavati, S. Chan, and Joseph Ross Mitchell. Real-time super resolution contextual close-up of clinical volumetric data. In *Proceedings of the Joint Eurographics - IEEE VGTC Symposium on Visualization (EuroVis06)*, pages 347–354. Eurographics Association, May 8-10 2006.

[69] Ulf Tiede, Thomas Schiemann, and Karl Heinz Höhne. High quality rendering of attributed volume data. In *Proceedings of the conference on Visualization '98*, VIS '98, pages 255–262, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.

[70] User Tkgd2007. A cutaway diagram of a coaxial cable. `http://en.wikipedia.org/wiki/File:Coaxial_cable_cutaway.svg`, May 31 2008.

[71] Stefan Tschöke, Reinhard Lüling, and Burkhard Monien. Solving the traveling salesman problem with a distributed branch-and-bound algorithm on a 1024 processor network. In *Proceedings of the 9th International Symposium on Parallel Processing*, IPPS '95, pages 182–189, Washington, DC, USA, 1995. IEEE Computer Society.

[72] Maarten J.G.M. Van Emmerik. A direct manipulation technique for specifying 3d object transformations with a 2d input device. *Computer Graphics Forum*, 9(4):355–361, 1990.

[73] John Viega, Matthew J. Conway, George Williams, and Randy Pausch. 3d magic lenses. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, UIST '96, pages 51–58, New York, NY, USA, 1996. ACM.

[74] Mike Viney. The virtual petrified wood, 2008. http://petrifiedwoodmuseum.org.

[75] L. Wang, Y. Zhao, K. Mueller, and A. Kaufman. The magic volume lens: an interactive focus+context technique for volume rendering. In *Visualization, 2005. VIS 05. IEEE*, pages 367 – 374, oct. 2005.

[76] Lvdi Wang, Yizhou Yu, Kun Zhou, and Baining Guo. Multiscale vector volumes. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, SA '11, pages 167:1–167:8, New

York, NY, USA, 2011. ACM.

[77] Sidney W. Wang and Arie E. Kaufman. Volume sculpting. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, I3D '95, pages 151–ff., New York, NY, USA, 1995. ACM.

[78] Yu-Shuen Wang, Chaoli Wang, Tong-Yee Lee, and Kwan-Liu Ma. Feature-preserving volume data reduction and focus+context visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(2):171–181, February 2011.

[79] Rdiger Westermann and Christof Rezk-Salama. Real-time volume deformations. *Computer Graphics Forum*, 20(3):443–451, 2001.

[80] Andrew S. Winter and Min Chen. Image-swept volumes. *Comput. Graph. Forum*, 21(3):441–450, 2002.

[81] Seung-Hyun Yoon and Myung-Soo Kim. Sweep-based freeform deformations. *Computer Graphics Forum*, 25(3):487–496, 2006.

[82] Xin Zhao, Wei Zeng, Xianfeng David Gu, Arie E. Kaufman, Wei Xu, and Klaus Mueller. Conformal magnifier: A focus+context technique with local shape preservation. *IEEE Trans. Vis. Comput. Graph.*, 18(11):1928–1941, 2012.