

University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

8-2014

One-sided transparency : a revolution in visualization.

Robert Curtin

University of Louisville

Follow this and additional works at: <https://ir.library.louisville.edu/etd>



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Curtin, Robert, "One-sided transparency : a revolution in visualization." (2014). *Electronic Theses and Dissertations*. Paper 2279.

<https://doi.org/10.18297/etd/2279>

This Master's Thesis is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

ONE-SIDED TRANSPARENCY: A REVOLUTION IN VISUALIZATION

By

Robert Curtin
B.S. (ECE), University of Louisville, 2013

A Thesis
Submitted to the Faculty of the
University of Louisville
J. B. Speed School of Engineering
as Partial Fulfillment of the Requirements
for the Professional Degree

MASTER OF ENGINEERING

Department of Electrical and Computer Engineering

August 2014

ONE-SIDED TRANSPARENCY: A REVOLUTION IN VISUALIZATION

Submitted by: _____
Robert Curtin

A Thesis Approved on

(Date)

by the Following Reading and Examination Committee:

Dr. Aly A. Farag, Thesis Director

Dr. Bruce Alphenaar

Dr. Chris Richards

ACKNOWLEDGEMENTS

I would like to thank Dr. Farag for providing me the opportunity to research at the Computer Vision and Image Processing lab as well as all those at the CVIP lab who have supported me in my academic pursuits. In particular, I would like to thank Marwa Ismail and Chuck Sites for their invaluable guidance and support in research. Without those two, I could not have done any of this. I would also like to thank my committee members, Dr. Bruce Alphenaar and Dr. Chris Richards for their participation in my defense, and Dr. Rob Falk for his insight and suggestions as a medical professional. Finally, I would like to thank my parents, Dan and Ellen Curtin, as well as my girlfriend, Molly Connor. They have provided much their love and support for me during my time at the University of Louisville.

ABSTRACT

Robert Curtin

July 2014

Colorectal cancer is one of the leading causes of death in the world. Colonoscopy, the traditional procedure for detecting colorectal cancer, is very effective. It does have downsides, however - it is invasive, uncomfortable for the patient, and not available to some patients with certain conditions. Virtual colonoscopy has been developed in order to address these issues. A virtual colonoscopy (VC) is a non-invasive method for performing a colonoscopy by using medical imaging data to create a virtual representation of the colon.

Previous virtual colonoscopy methods include fly-through, fly-over, flattening, and the unfolded cube method. Fly-through moves the camera through the inside of the colon, following a centerline from the length of the colon. Fly-over splits the colon into halves longitudinally, and flies a camera over each half. Flattening reduces the 3D colon model to a 2D image. The unfolded cube method flies a set of cameras along the centerline as in flythrough, but where flythrough had one camera looking along the centerline, the unfolded cube method presents views from six cameras. The six camera views are positioned in the pattern of an unfolded cube, which gives rise to the method's name.

This thesis will present a new method called one-sided transparency (OST). This is a method for visualizing virtual objects so that the interior surfaces can be viewed from the outside. OST has numerous improvements over existing VC methods, particularly

when combined with fly-over methods. However, this thesis will also demonstrate that OST is not limited to fly-over nor even to VC, as it has applications in multiple fields.

For quantitative evaluation, this thesis focused on comparing specific scenarios that OST excels in visualizing. Fly-through navigation has difficulties with polyps between haustral folds, and prior fly-over work had visual artifacts that degraded the quality of the final visualization. These and other specific cases are visualized using OST in order to highlight the power of this new technique. Additionally, the previous FO method had some significant drawbacks that are solved by the application of OST. These problems and their origins will be addressed, along with the way that OST solves them.

This thesis will also explore potential applications for OST outside of VC. This will include a more general visualization of tubular objects. It will be shown that OST has the ability to highlight structural issues and deformities such as cracks and bumps. This has potential applications in medical fields outside of VC as well as in structural engineering. This will demonstrate OST's usefulness as a general technique, even outside the context of VC.

Finally, this thesis will present results regarding OST for VC. It will show that OST presents several advantages over previous VC methods. OST allows easier viewing of polyps in difficult locations and offers a more complete view of the colon. OST has a number of advantages over the existing fly-over method, including faster time-to-viewing, less sensitivity to centerline error, and improved accuracy in the separation of halves.

TABLE OF CONTENTS

| | |
|--|------|
| ACKNOWLEDGEMENTS | iv |
| ABSTRACT | v |
| TABLE OF CONTENTS | vii |
| LIST OF FIGURES | viii |
| CHAPTER 1 – INTRODUCTION | 1 |
| 1.A Colorectal Cancer | 1 |
| 1.B Colonoscopy | 2 |
| 1.C Virtual Colonoscopy | 4 |
| 1.D Contribution of This Thesis | 10 |
| CHAPTER 2 - LITERATURE REVIEW OF VISUALIZATION | 13 |
| 1.A 3D Surface Visualization | 13 |
| 1.B Marching Cubes | 14 |
| 1.C Back-Face Culling | 19 |
| 1.D Centerline | 21 |
| 1.E Summary | 21 |
| CHAPTER 3 - LITERATURE REVIEW OF VIRTUAL COLONOSCOPY | 23 |
| 3.A Introduction | 23 |
| 3.B Fly-through Visualization Method | 23 |
| 3.C Fly-over Visualization Method | 26 |
| 3.D Summary | 35 |
| CHAPTER 4 - ONE-SIDED TRANSPARENCY TECHNIQUE | 36 |
| 4.A Concept of One-Sided Transparency | 36 |
| 4.B Algorithm for One-Sided Transparency | 39 |
| 4.C Summary | 43 |
| CHAPTER 5 - ONE-SIDED TRANSPARENCY IN VIRTUAL COLONOSCOPY | 44 |
| 5.A One-Sided Transparency in Fly-over | 44 |
| 5.B One-Sided Transparency with the Clipping Plane Method | 48 |
| 5.C One-Sided Transparency without Navigation | 54 |
| 5.D GUI Interface for One-Sided Transparency in Fly-Over | 56 |
| 5.E Summary | 59 |
| CHAPTER 6 - VALIDATION AND RESULTS | 61 |
| 6.A Comparison to Fly-through | 61 |
| 6.B Comparison to Fly-over | 64 |
| 6.B.1 Polyps Located on the Cutting Plane | 64 |
| 6.B.2 Holes in Ring Generation | 67 |
| 6.B.3 Computation time | 68 |
| 6.C Summary | 70 |
| CHAPTER 7 - OTHER ONE-SIDED TRANSPARENCY APPLICATIONS | 72 |
| 7.A One-Sided Transparency Applied to Determining Structural Integrity | 72 |
| 7.B Further Medical Applications for One-Sided Transparency | 74 |
| 7.C Summary | 76 |
| CHAPTER 8 - CONCLUSIONS AND FUTURE WORK | 78 |
| REFERENCES | 79 |

LIST OF FIGURES

| | |
|---|----|
| Figure 1: Anatomy of the digestive system | 2 |
| Figure 2: Colonoscope | 3 |
| Figure 3: Diagram of fly-through navigation..... | 5 |
| Figure 4: Image obtained during fly-through navigation..... | 5 |
| Figure 5: Diagram of fly-over navigation..... | 6 |
| Figure 6: Image obtained during fly-over navigation | 7 |
| Figure 7: Diagram of flattening, with image obtained during flattening..... | 8 |
| Figure 8: Diagram of the cameras in the unfolded cube technique | 9 |
| Figure 9: Example of the unfolded cube technique | 10 |
| Figure 10: 3D surface broken down into its fundamental parts..... | 13 |
| Figure 11: Sphere with visible edges | 14 |
| Figure 12: Sample of marching cubes configurations | 15 |
| Figure 13: Result of marching cube algorithm, with edges highlighted..... | 16 |
| Figure 14: Surface obtained by marching cubes algorithm | 17 |
| Figure 15: Surface obtained by smoothing the surface generated by the marching cubes algorithm..... | 18 |
| Figure 16: Diagram of face orientation..... | 19 |
| Figure 17: Virtual Fly-through of a colon..... | 24 |
| Figure 18: Polyps in between folds missed by fly-through navigation | 25 |
| Figure 19: Diagram of fly-over splitting..... | 26 |
| Figure 20: Centerline with normal and tangent vectors..... | 28 |
| Figure 21: Pipeline of virtual fly-over | 31 |
| Figure 22: Diagram of incorrect ring assignment..... | 32 |
| Figure 23: Holes generated by the original fly-over method..... | 33 |
| Figure 24: Location of missing segments from holes..... | 34 |
| Figure 25: Box visualized normally and using OST..... | 36 |
| Figure 26: Textured tube visualized normally, with OST and a wire-frame, and using OST alone | 38 |
| Figure 27: Back-face culling used as intended | 40 |
| Figure 28: Klein bottle visualized normally, with OST and a wire-frame, and using OST alone..... | 41 |
| Figure 29: Klein bottle visualized from another angle | 42 |
| Figure 30: Mobius strip with visible discontinuity revealed by OST..... | 42 |
| Figure 31: Pipelines for original fly-over visualization and fly-over visualization with OST | 45 |
| Figure 32: OST-FO camera setup viewing the colon | 46 |
| Figure 33: Normal and view up vector computation | 48 |
| Figure 34: Diagram of near and far clipping planes | 49 |
| Figure 35: Diagram of clipping plane method..... | 50 |
| Figure 36: Clipping plane method applied to a section of the colon | 51 |
| Figure 37: Clipping plane method resolving occlusion issues..... | 52 |
| Figure 38: Clipping plane method combined with OST..... | 53 |
| Figure 39: Pipelines for visualization without navigation | 54 |
| Figure 40: Colon rendered normally..... | 55 |

| | |
|--|----|
| Figure 41: Colon rendered using OST without navigation | 56 |
| Figure 42: GUI interface developed in Python and Tkinter | 58 |
| Figure 43: View of a polyp as it rotates into view through use of the Roll commands...59 | |
| Figure 44: OST-FO compared to fly-through..... | 63 |
| Figure 45: Cross-section of a colon visualized using fly-over and OST | 65 |
| Figure 46: Three polyps located on the cutting plane, visualized using FO and OST-FO..... | 66 |
| Figure 47: OST-FO polyp visibility while rolling the cameras about the centerline..... | 67 |
| Figure 48: Section with holes in the original FO visualized using OST | 68 |
| Figure 49: First startup time of previous FO compared to OST-FO..... | 70 |
| Figure 50: Spiral object with crack shown normally and using OST | 72 |
| Figure 51: Spiral object with interior crack shown normally and using OST | 73 |
| Figure 52: Thick tube with an air pocket, shown normally and using OST | 73 |
| Figure 53: Tooth shown normally and using OST..... | 74 |
| Figure 54: OST applied to the spine | 75 |
| Figure 55: Lung visualized using OST | 76 |

I. INTRODUCTION

A. Colorectal Cancer

Colorectal cancer is a cancer that affects the colon and/or rectal region of the large intestine, figure 1. It is a leading cause of death in first world countries [1]. The main indicator of potential colorectal cancer is the existence of pre-cancerous growths along the inside of the large intestine. One of these growths is known as a polyp. Polyps usually are not dangerous, but this cannot be determined without taking a sample of the polyp for testing. A polyp 10 mm or larger in size has an 8% risk of developing into cancer within 10 years [1].

While colorectal cancer is deadly, it is also highly preventable. If the cancer is caught early enough, it may be curable with surgery. Surgery performed on colorectal cancer detected early has a 5-year survival of 95%, while surgery performed on colorectal cancer in its later stages has a 5-year survival of 5% [1]. Thus, screening is recommended beginning at age 50 and repeating every 10 years. The most common method of screening is a colonoscopy, while virtual colonoscopy currently exists as an alternate screening method.

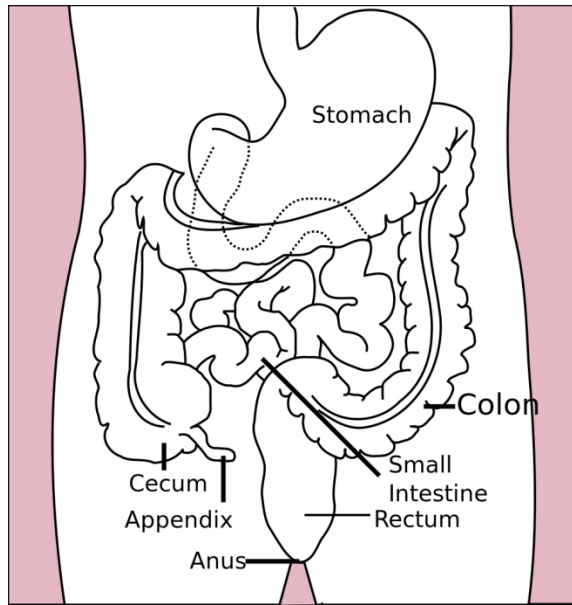


FIGURE 1 – Anatomy of the digestive system [2]

B. Colonoscopy

A colonoscopy is an examination of the large intestine with the intent of detecting polyps. To prepare, the patient must clear their colon of solid matter by sticking to a liquid diet or low-fiber diet several days prior to the procedure. This will prevent false identification of fecal matter as polyps, while also preventing polyps from being occluded by fecal matter. The day before the procedure, the patient will be given a laxative preparation or will undergo whole bowel irrigation in order to ensure that no fecal matter remains. [1]

Before the colonoscopy, the patient undergoes sedation. The colonoscopy begins when a colonoscope (a flexible, lighted tube with a video camera on the end, figure 2) is inserted into the patient's anus. Simultaneously, air is puffed into the colon to allow

greater visibility. If a polyp of significant size is found, a sample of it will be removed for testing.



FIGURE 2 – Colonoscope [3]

While colonoscopy is the current gold standard for colorectal cancer screening, it does have disadvantages. First of all, many patients find it uncomfortable and some will elect to forgo the procedure on those grounds alone. Secondly, some patients may not be able to undergo a colonoscopy due to medical conditions [4]. Thirdly, sedation is often used and thus someone must accompany the patient home. Finally, bleeding or tearing of the colon due to instrument complications can occur, although this is rare [1].

C. Virtual Colonoscopy

Virtual Colonoscopy (VC) is a non-invasive alternative to the traditional colonoscopy. Here is a typical pipeline for a generic VC system: First, CT or MRI scans of the patient are acquired. Next those scans are segmented to isolate the colon. The isolated colon is then visualized in some manner, in most cases as a 3D model. Finally, the physician examines this final visualization. This thesis will focus on improvements in the visualization step, but will outline the preceding steps as well.

VC requires the patient to clear their colon, much like a traditional colonoscopy. In order to acquire the scans, the patient lies in the supine position and a small tube is inserted into the anus to inflate the colon. The patient is then scanned with either CT or MRI. The patient repeats this procedure in the prone position. Both prone and supine positions are necessary, as polyps may be located in residual fecal matter in one orientation. Typically, the scans are low-dose to minimize exposure to radiation. Once these scans are acquired, they can be visualized by any number of methods. The two methods that this thesis will focus on are fly-through and fly-over, but here we will describe several methods for a more comprehensive overview of the field.

Fly-through navigation involves virtually flying the camera through the colon, mimicking the behavior of a colonoscope. Figure 3 shows this process, and figure 4 shows one view obtained during a fly-through navigation. It was originally developed by [5] as the first proposed method for VC. The first VC fly-through video was presented in February 1994, with the musical accompaniment of Wagner's "Ride of the Valkyries" [4]. [6] details a more modern implementation of fly-through navigation. This method has

the benefit of being familiar to physicians and has been approved for use as a screening method.

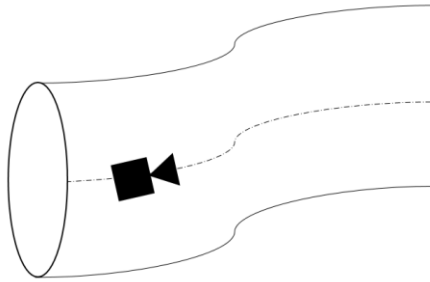


FIGURE 3 - Diagram of fly-through navigation, where the camera follows the centerline through the colon.



FIGURE 4 – Image obtained during fly-through navigation

The CVIP lab first introduced the fly-over visualization in [7] and has obtained a patent on the method. In order to advance previous work at the CVIP lab, this thesis focuses on improving the existing fly-over method. In the fly-over visualization, the colon is split into halves and each half is viewed from above, as outlined in figure 5.

Figure 6 shows what a fly-over looks like. This technique provides 20% greater visibility than fly-through, but it has not yet been FDA approved.

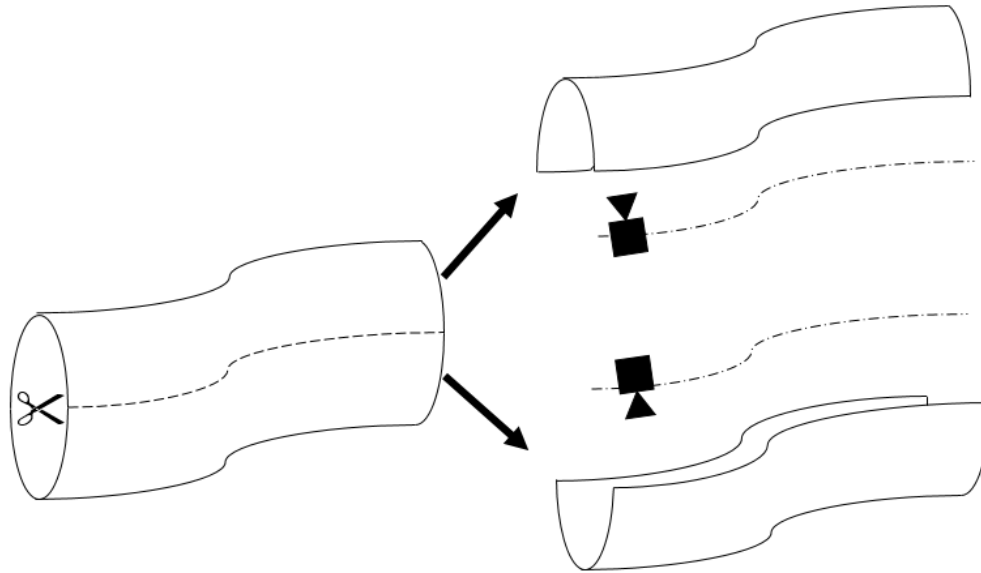


FIGURE 5 - In fly-over navigation, the colon is cut into halves and a camera flies over each half

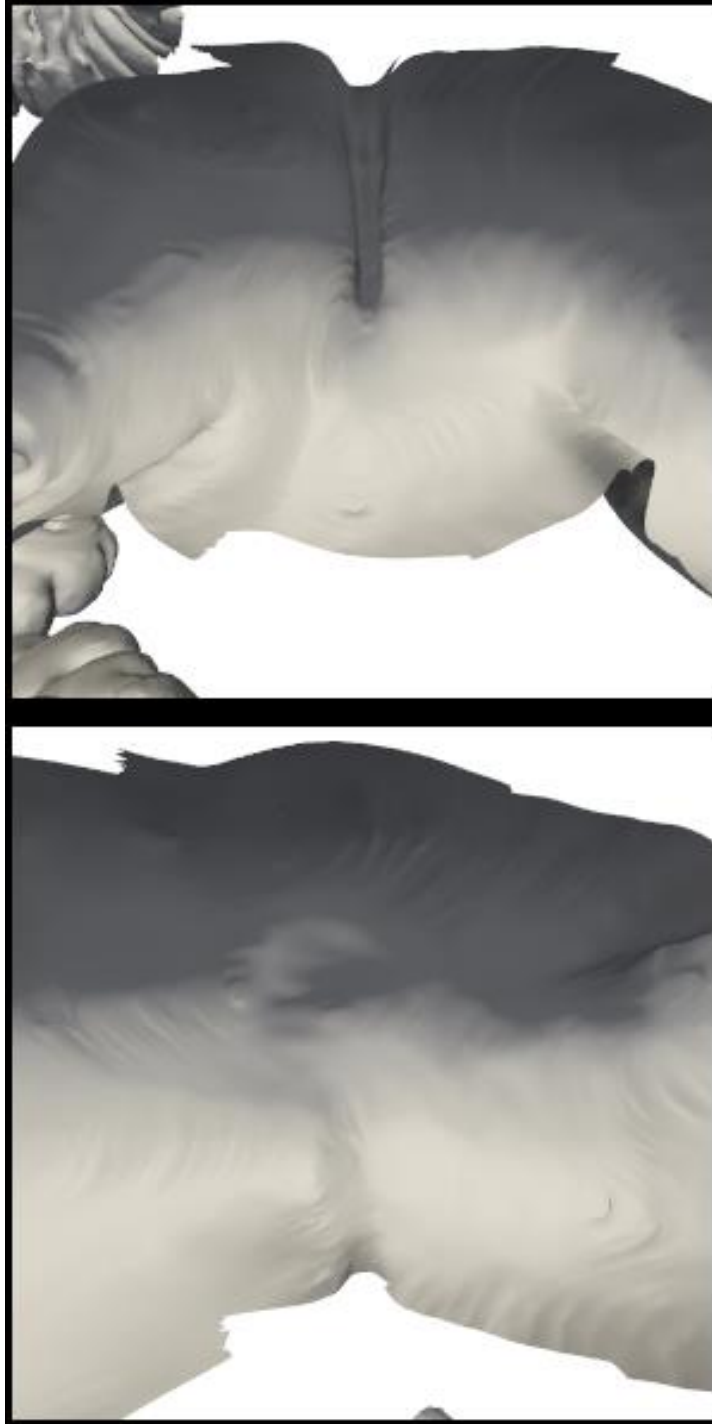


FIGURE 6 - Image obtained during fly-over navigation. The top view flies over one half of the colon, while the bottom view flies over the other half.

Flattening, shown in figure 7, is a method that converts the colon surface into a 2D image. This was originally proposed in [8] and has been further developed by [9] and

[10]. There are multiple methods for flattening the colon, depending on whether it is topologically treated as a genus 0 or genus 1 surface. [9] and [10] achieve flattening by reducing the colon to a perfectly tubular object with a textured surface. They then unfold that tubular object to obtain the final view, as figure 7 illustrates.

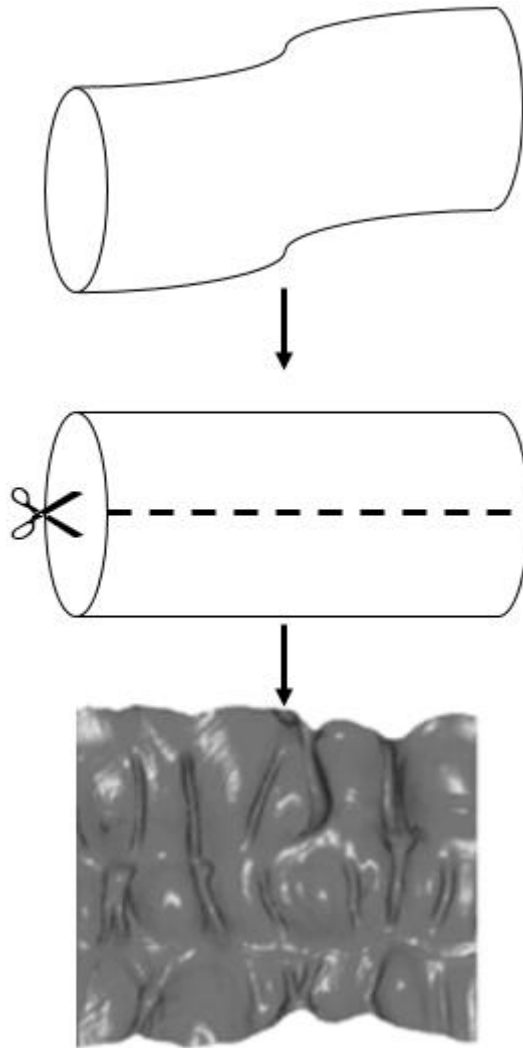


FIGURE 7 - In flattening techniques, the colon is converted to a 2D object. Here, the colon is mapped to a cylinder and then that cylinder is unrolled to create a 2D image. 2D image courtesy of [11]

[12] gives one example of the unfolded cube method. The unfolded cube method is similar to the fly-through method, but it uses six cameras instead of one, figure 8. In

this way, sideways views of the colon can be visualized alongside the forward and reverse views that fly-through could offer. These six views are stitched together as if they were part of an unfolded cube. Figure 9 shows how each view corresponds to the cube, as well as the view resulting from projecting the unfolded cube onto a 3D cube.

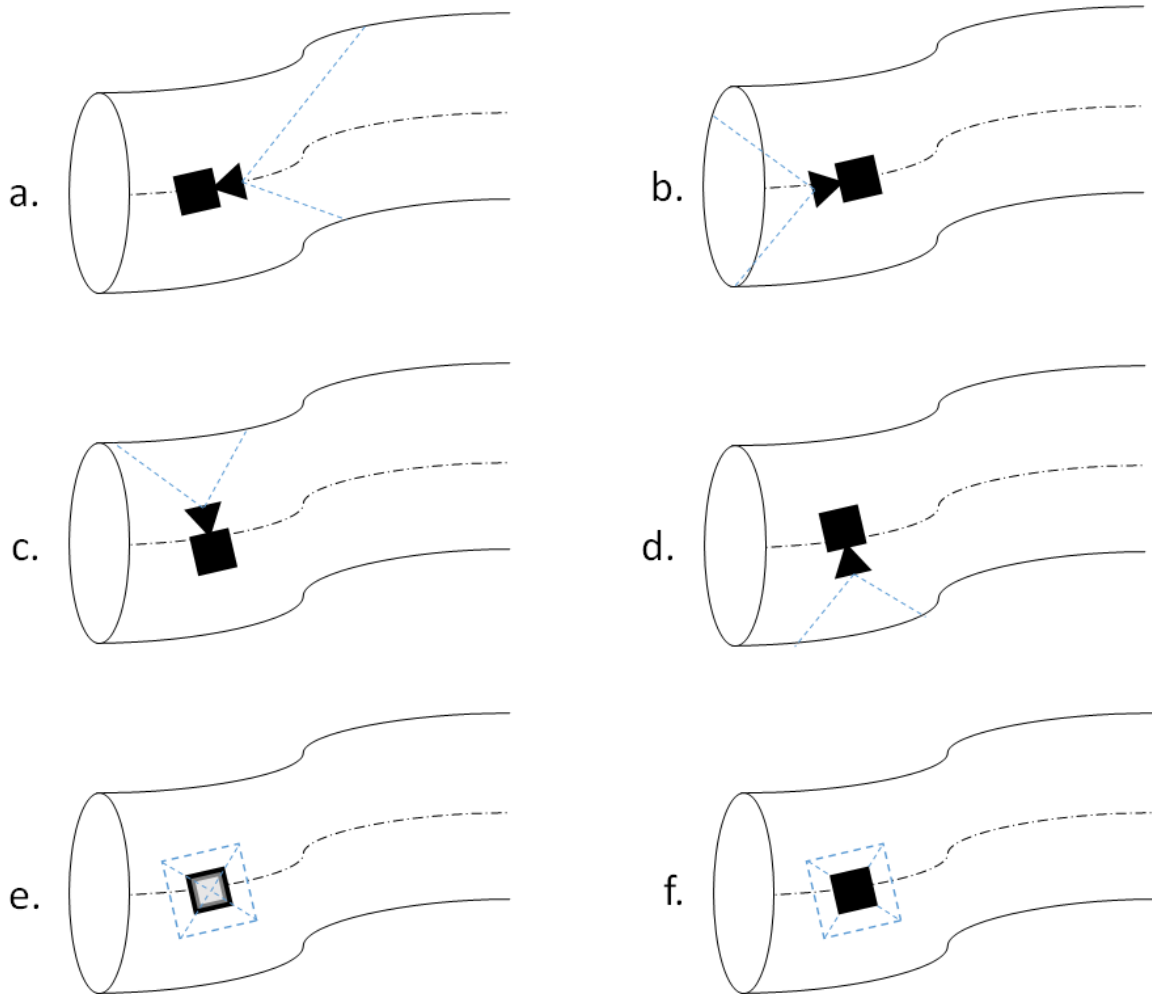


FIGURE 8 - Six views of the unfolded cube technique. a,b view forward and backward as in fly-through navigation, while c,d,e,f all view the sides of the colon similar to fly-over navigation.

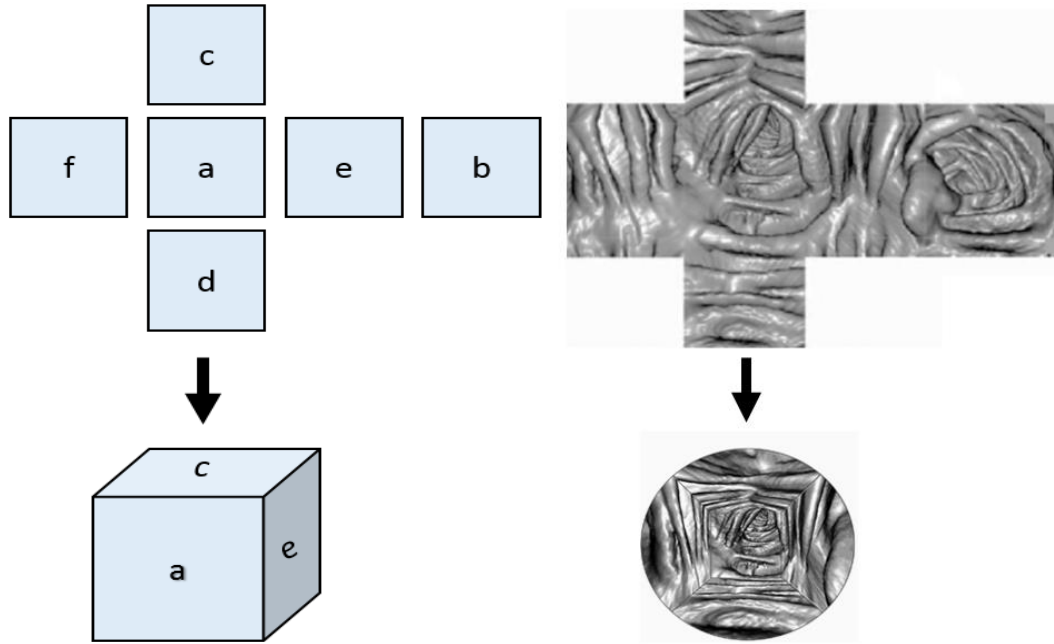


Figure 9 - The six views are stitched together into an unfolded cube. This can also be refolded into a cube and visualized. Images courtesy of [11]

Virtual colonoscopy has a number of advantages over traditional colonoscopy that make it worth considering. It is minimally invasive, as it does not require the insertion of a colonoscope. It does not require sedation. Moreover, VC techniques compare favorably with traditional colonoscopy techniques [13].

D. Contribution of This Thesis

This thesis will describe a novel method called One-Sided Transparency (OST) used for visualizing the interior of 3D surfaces. It will provide the background necessary to understand the method and some of its applications. It will also explain the method

both conceptually and mathematically. Finally, it will provide several applications for the method and demonstrate results obtained through using the method.

Chapter II will give a background of 3D visualization techniques that involve visualizing the surface of an object. These techniques include the marching cubes algorithm, back-face culling, and centerline generation. The marching cubes algorithm generates a surface for a set of points. Back-face culling removes faces on the surface that aren't currently visible, which speeds up visualization. Back-face culling is also key to achieving OST. Finally, centerline generation creates a line that passes through the center of an object.

Chapter III will provide a review of existing virtual colonoscopy techniques. Virtual colonoscopy is the application that OST was developed for, and which is the main focus of this thesis's research. This chapter will focus on two virtual colonoscopy techniques: fly-through visualization and fly-over visualization. Fly-through visualization is the current gold standard for virtual colonoscopy, and is already seeing clinical use. It involves flying a camera through the inside of the colon. Fly-over visualization is a method developed by the CVIP lab that is enhanced by OST. It involves splitting the colon into halves and viewing each half from above.

Chapter IV will detail the OST technique. It will explain the concept behind the technique as well as the algorithm for it. It will also expand on the OST concept by introducing the clipping plane method used to improve OST. The clipping plane method covers some of the weak points of OST, including occlusion.

Chapter V will introduce applications for OST within virtual colonoscopy. The primary application of OST is augmenting fly-over navigation, but several other

applications will be discussed. It will introduce the clipping plane method, which enhances OST within fly-over. It will describe a method for using OST without navigation, and explain some uses for this method. This chapter will also showcase the GUI developed to combine OST and fly-over navigation.

Chapter VI will demonstrate how OST's practicality has been validated using both qualitative and quantitative analysis. It will cover a qualitative comparison to both fly-through and fly-over navigation techniques. Notably, it will explain how OST covers several of the drawbacks previously associated with fly-over navigation.

Chapter VII will showcase some applications for OST outside of virtual colonoscopy. These other applications include determination of structural integrity through visualizing weaknesses such as cracks and air pockets. Additionally, several potential medical applications will be proposed, including dental, spinal, and respiratory visualizations.

Finally, chapter VIII will discuss the conclusions of this thesis's work, and explain what further work can be done. This will include further enhancement of OST within virtual colonoscopy as well as exploration of OST's possibilities outside of virtual colonoscopy.

II. LITERATURE REVIEW OF VISUALIZATION

A. 3D Surface Visualization

In most applications, it is neither necessary nor practicable to visualize a 3D object as a solid object. As the surface is the only part of the object that is visible, most applications only represent the surface. In computational geometry, the surface is approximated using discrete 2D elements known as polygons.

Each polygon can be defined by its vertices, its edges, or its faces, figure 10. Its vertices are the angular points at the edge of the polygon. Its edges are the lines resulting from connecting these vertices. Finally, its faces are the 2D space that the lines surround. All points within a polygon are coplanar.

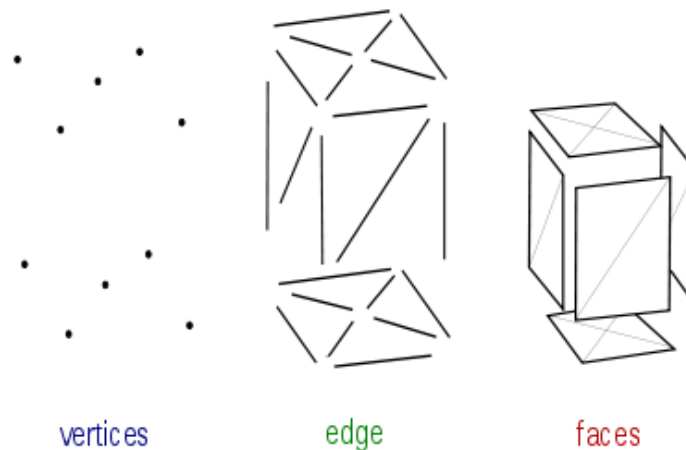


FIGURE 10 - 3D surface broken down into its fundamental parts. Image adapted from [14]

This surface is represented as a polygon mesh, which is a set of connected polygons. Figure 11 shows a sphere with the edges of its polygonal mesh outlined.

Usually, a polygon mesh consists only of triangles, in which case it is called a triangle mesh.

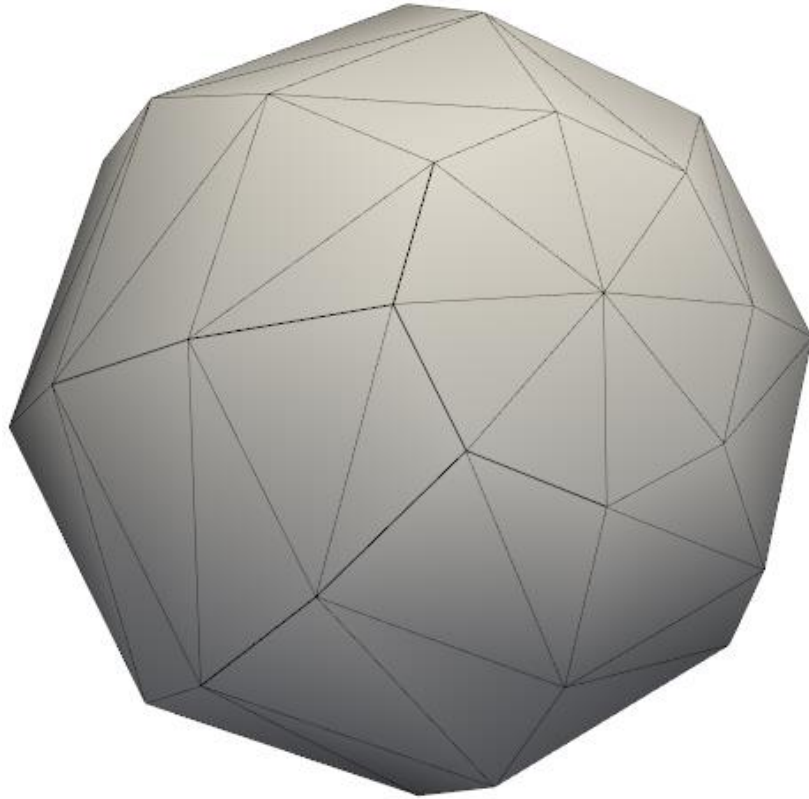


FIGURE 11 - Sphere with visible edges

B. Marching Cubes

Originally presented in [14] in 1987, Marching cubes is an algorithm for extracting the polygonal mesh of a set of points. Since its patent expired in 2005, it has been widely used for CT and MRI scans. This thesis uses an implementation of this

method to generate 3D models of segmented CT scans. The implementation used here is provided in the Visualization Toolkit (VTK) [15].

Marching cubes proceeds through the data, considering eight neighbor locations at a time. Each neighbor is treated as one vertex of a cube. Then, the algorithm determines whether each neighbor is located inside the cube, or outside of the cube. The algorithm then matches the configuration of points to a set of existing templates, seen in figure 12. These templates are designed to be rotationally and reflectively symmetric and deliver a continuous surface between templates.

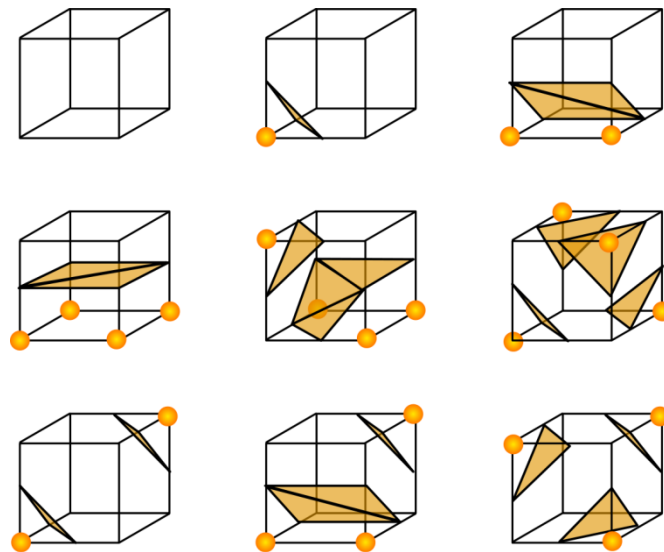


FIGURE 12 – Sample of marching cubes configurations. Points at vertices indicate points that are inside the surface, triangles indicate new faces of the generated surface. Image adapted from [16].

Figure 13 shows a sample output of this algorithm. The resulting surface can be very rough depending on the original resolution of the data set provided. In this case, the face has many sharp transitions, while the side of the head is very flat. In order to resolve this, the output surface is smoothed to create a cleaner looking image. Figure 14 shows the original output of the marching cubes algorithm compared to a smoothed version. The

smoothed version of figure 15 is the final result after applying several algorithms that enhance visualization.

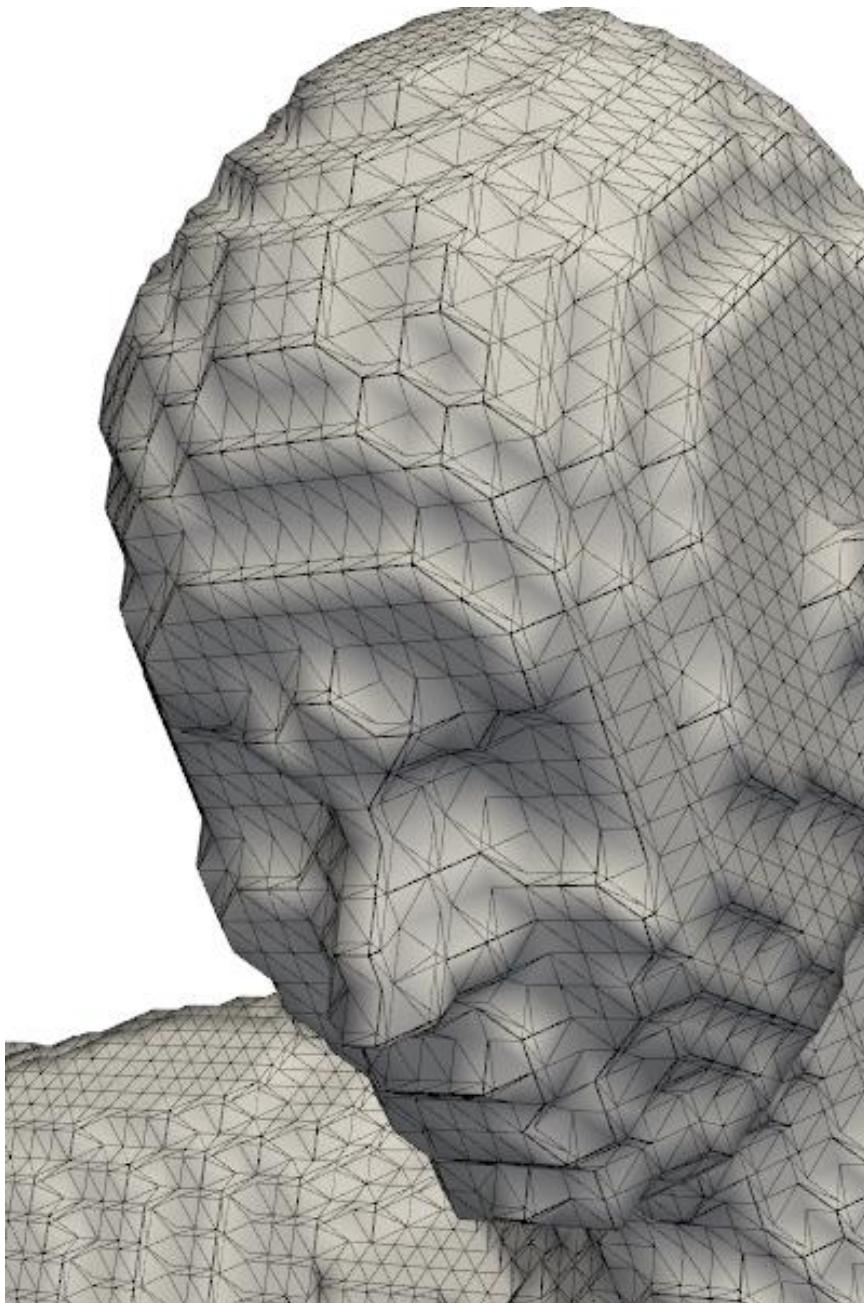


FIGURE 13 - Result of marching cube algorithm, with edges highlighted.



FIGURE 14 - Surface obtained by marching cubes algorithm. Note the jagged transitions.

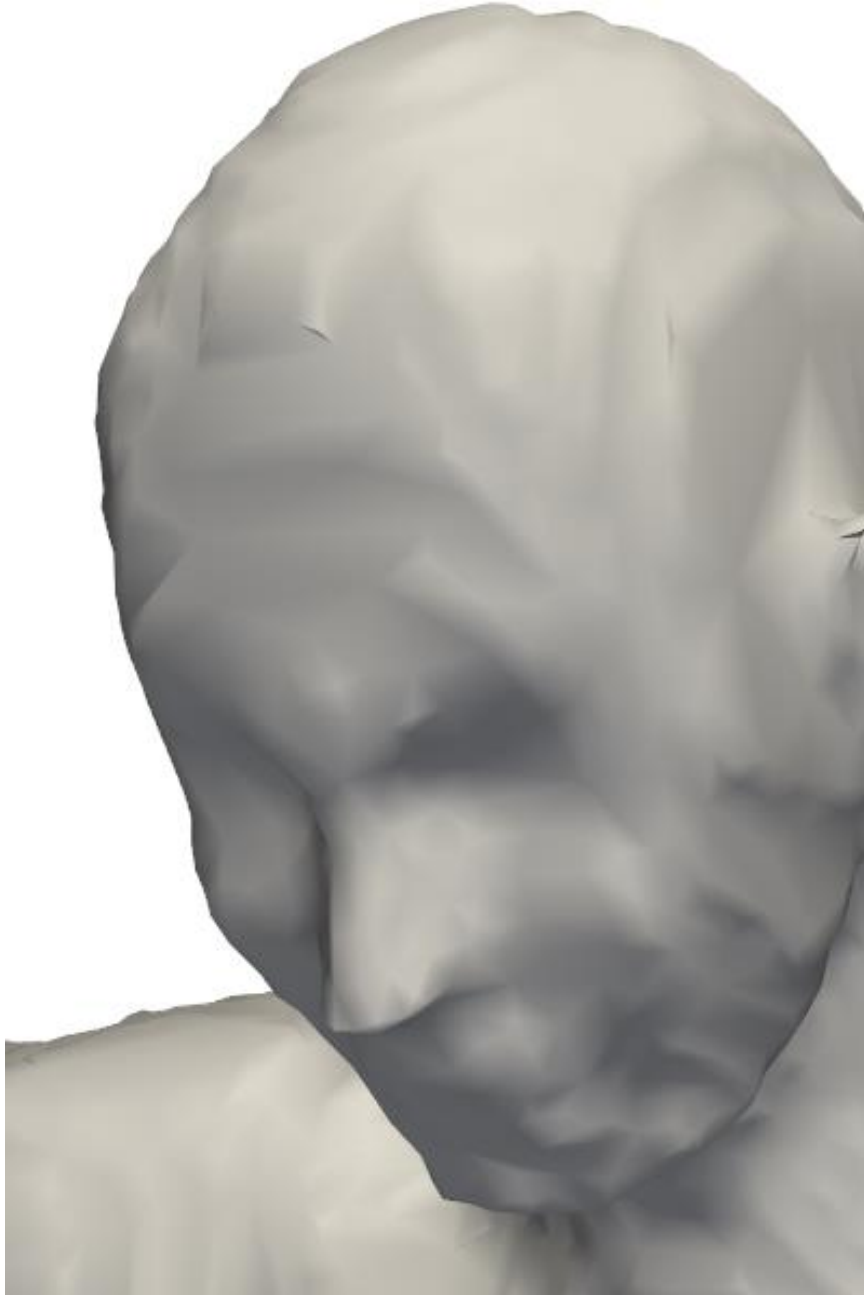


FIGURE 15 - Surface obtained by smoothing the surface generated by the marching cubes algorithm

C. Back-Face Culling

In order to understand OST, we must first understand back-face culling. Back-face culling is a computer graphics technique that is most commonly used to optimize rendering with the goal of improving frame rate [17]. It disables rendering for faces of polygons that are not normally visible, which allows the graphics engine to ignore those faces while rendering. For example, the interior of a ball wouldn't normally be visible, so all faces within the sphere can be ignored.

In practice, back-face culling is handled by a graphics library such as VTK [18]. However, it is necessary to understand the process in order to comprehend OST's implementation of it. First, consider a triangular face defined by points $p1, p2, p3$. Back-face culling will decide whether to discard the face or not based on the order of these points. If the points $p1, p2, p3$ appear clockwise with respect to the camera, as in figure 16(a), the face will be discarded. But if the points appear counterclockwise with respect to the camera, as in figure 16(b), they will not be discarded.

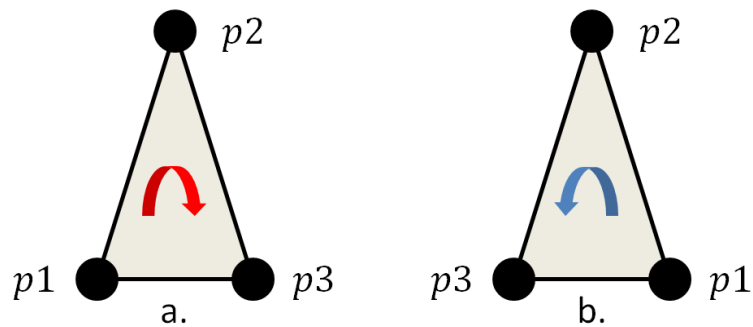


FIGURE 16 - a) Points oriented clockwise with respect to the camera, this face will be discarded b) Points oriented counterclockwise to the camera, this face may be visualized

Back-face culling requires a normal vector for each face. For a triangle defined by points p_1, p_2, p_3 ; we assign vectors $A = p_2 - p_1$ and $B = p_3 - p_1$. Then we calculate the normal vector $N = (N_x, N_y, N_z)$:

$$N_x = A_y B_z - A_z B_y$$

$$N_y = A_z B_x - A_x B_z$$

$$N_z = A_x B_y - A_y B_x$$

This same computation can be performed on non-triangular faces by simply using three arbitrary points. The normal vector is calculated once and stored for future use. This is because the back-face culling that uses these vectors is computed every time the scene is updated, and it would be excessive to calculate this every time.

Back-face culling decides whether or not to discard a face based on its orientation. First, it gets the camera point p_c and a point on the face p_f . Next, it assigns a view vector $V = p_f - p_c$ from the camera point to the face point. Once it has this vector as well as the normal, it can compute a scalar s :

$$s = V \cdot N$$

If this scalar is positive, then the normal is pointed away from the camera and the face can be discarded. If this scalar is negative, then the normal is pointed towards the camera and the face should be considered for visualization.

D. Centerline

The centerline, also known as a curve skeleton, is a key concept for VC and is the primary means of navigating through the complex structure of the colon [19]. A centerline is a path through the center of a 3D model. [20] explains the CVIP lab's method for generating curve skeletons. That paper describes a curve skeleton as a compact representation of a shape that consists of one-dimensional curves. When a centerline is created for a complex 3D objects, the centerline will contain multiple paths and branches. Within VC however, the curve skeleton will result in a single line that flows from one end of the colon to the other.

E. Summary

Objects in 3D space are typically visualized using a polygon mesh wrapped around the object's surface. Marching cubes is one algorithm for generating a polygon mesh from a set of points. It traverses the data, and at each step it compares eight neighbor points and places triangles based on their values.

Back-face culling is a method for speeding up visualization, by discarding faces that are not currently visible to the camera. This is done by examining the face's normal vector with respect to the camera's current position. If the normal points in the direction of the camera, the face will be considered for visualization. If the normal points away from the camera, the face can be discarded.

The centerline is the path through the center of an object. The CVIP lab has devised a method for generating a centerline for 3D objects. One of the many applications for the centerline is in virtual colonoscopy, where it is used to generate a path to navigate the colon.

III. LITERATURE REVIEW OF VIRTUAL COLONOSCOPY

A. Introduction

There are two methods that have strong relevance to this thesis. Fly-through is the existing gold standard, and it is important that any new technique is compared to it. Fly-over is the method patented by the CVIP lab, and this thesis builds upon this method. This chapter will provide an overview of both visualization methods, along with current implementations of both at the CVIP lab. It will also detail existing problems with these methods, in order to highlight issues that this thesis aims to solve.

B. Fly-through Visualization Method

Virtual fly-through is the first VC method developed. It was first presented in [5] in 1994 and has since become an accepted method for colorectal cancer screening. It sees widespread use in industry, and is an easy method for physicians used to a colonoscopy. The general idea of fly-through is to mimic a colonoscopy by flying a virtual camera along the centerline through the 3D colon model. This method is the gold standard for VC because it is familiar to traditional colonoscopy and thus does not require much acclimation from the physician.

As fly-through performed in one direction could potentially miss polyps located on the opposite side of a fold, it must be performed in both directions [6]. Thus, a complete fly-through screening consists of two supine traversals and two prone traversals, for a total of four traversals.

This thesis implemented a simple version of fly-through, using the following method: First, acquire the colon's 3D model and centerline through some means. Then place the camera at the start of the centerline and move it along the centerline until it reaches the end, keeping the camera focused on the next point in the centerline. Next, traverse the opposite direction by focusing the camera on the previous point in the centerline, move the camera from the end to the start. Figure 17 shows this thesis's implementation of fly-through.

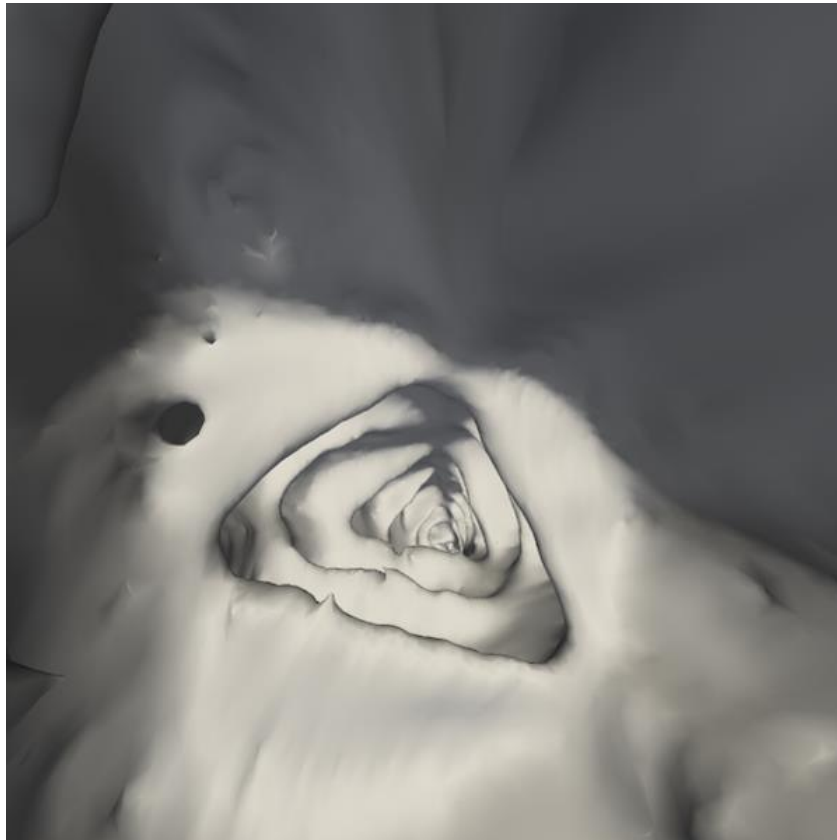


FIGURE 17 - Virtual Fly-through of a colon.

Fly-through is an effective method, but it does have some issues. First of all, it requires four traversals of the colon, which is difficult for the physician mentally and is also time-consuming [21]. Secondly, it can potentially miss polyps located between haustral folds [6]. Figure 18 shows how a polyp on one side of a haustra will only be visible in one direction, which is why two traversals are necessary. It also shows how a polyp located between folds could be missed entirely.

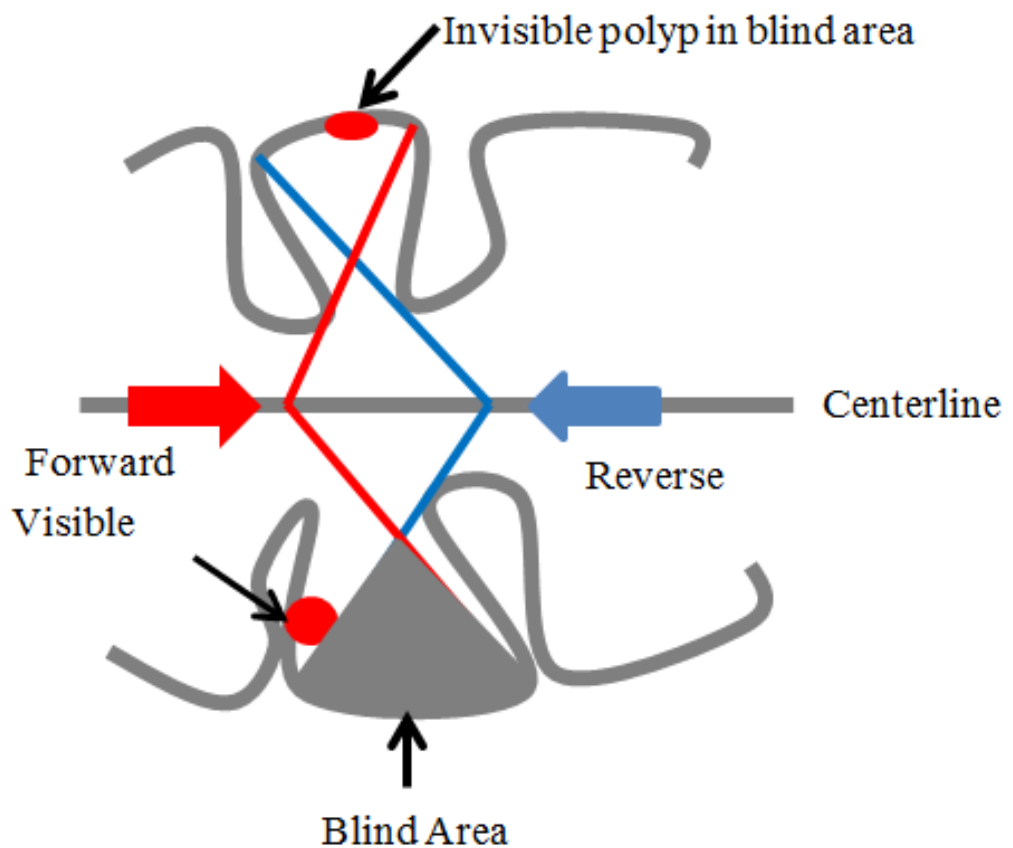


FIGURE 18 - Polyps in between folds may be missed by fly-through navigation. Additionally, a polyp on one side of a fold is only visible in one direction of fly-through. Image courtesy of [11]

C. Fly-over Visualization Method

Virtual Fly-over is a VC method patented by the CVIP lab. It was first proposed in [7] in 2006. This method aims to split the colon in half along its length, then view each half individually, figure 19. The method begins with a 3D colon surface obtained through segmentation. It splits the colon into smaller rings, cuts each ring individually in half, and then combines the half-rings to create the final half-colon.

The colon must be split along its length, which is a challenging issue. Previous work achieved this by dividing the colon surface into continuous rings before cutting. These rings are selected so that a single cutting plane can divide the ring into two halves.

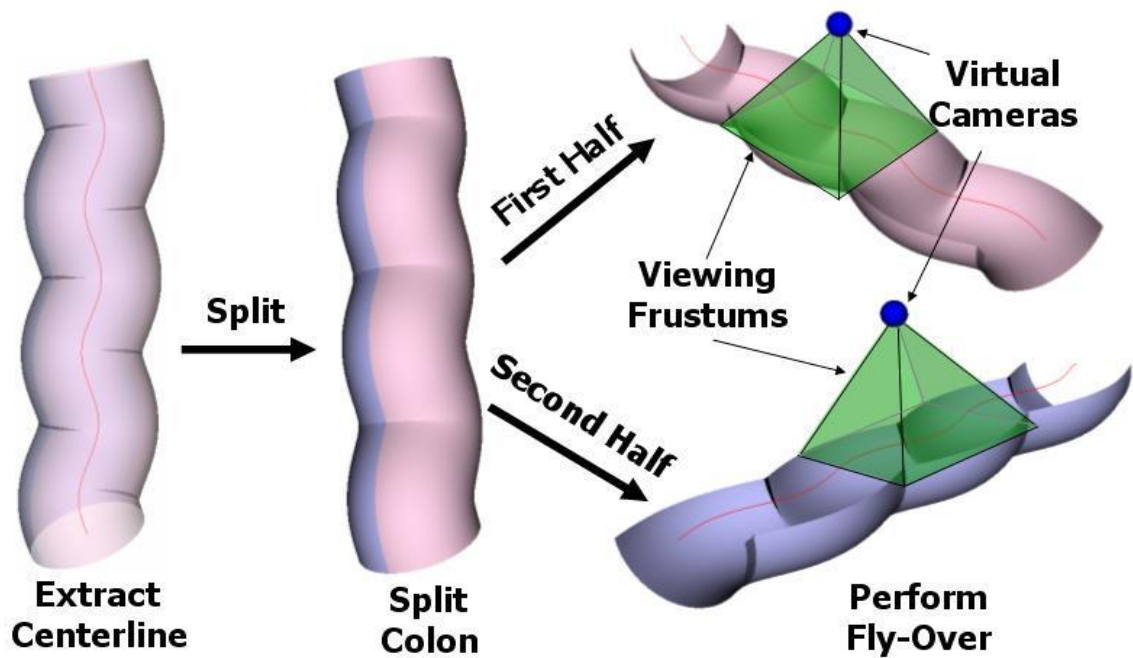


FIGURE 19 – In fly-over, the colon is split into two halves and then a camera flies over each half of the colon. Image courtesy of [7]

Let C be a smoothed version of the colon's centerline, which is sampled uniformly to N_p points, each is labeled P_i as shown in figure 20. The centerline C is also

divided into N_s segments l_j , where $1 \leq j \leq N_s$. The points of each segment l_j are assigned a specific label based on some *label* function.

$$\forall(P_i) \in l_j, label(P_i) = j$$

Once all triangles are labeled, the set of surface triangles with the same label form one possible ring. Let c_k be the geometric center of each surface triangle r_k , where $1 \leq k \leq N_r$, and N_r is the total number of surface triangles. Points within a certain ring \hat{P}_l , are found and labeled by the associated ring r_k :

$$\hat{P}_l = \arg \min_{P_i \in C} \|c_k - P_i\|^2$$

$$label(r_k) = label(\hat{P}_l)$$

Vectors normal to the centerline are used at each ring in order to split that ring. In order for the ring halves to be attached later, these normals must be aligned to each other. This will ensure that the colon is continuous between rings. Thus, the normal generation method must relate each normal to the previous normal. The method given here was presented for use in flyover in [7].

Each ring R_j spans multiple centerline points with starting point q_j and ending point q_{j+1} . Figure 20 shows \vec{u}_j , p_j , and \vec{t}_j with respect to the centerline, where \vec{t}_j is a vector tangent to the centerline. In order to split R_j into two halves, it is clipped using an infinite plane π_j with center q_j and normal vector \vec{u}_j given by

$$\vec{u}_j = (q_{j+1} - q_j) \times \vec{n}_j$$

The normal vector \vec{n}_j at each centerline point p_1 is given by

$$\vec{w}_j = \vec{u}_j \times \vec{t}_j$$

$$\vec{n}_l = \vec{t}_l \times \vec{w}_l$$

Where $\vec{u\bar{p}}$ is an arbitrary vector, chosen at each point to be any of the Cartesian basis vectors provided that it is not parallel to \vec{t}_i .

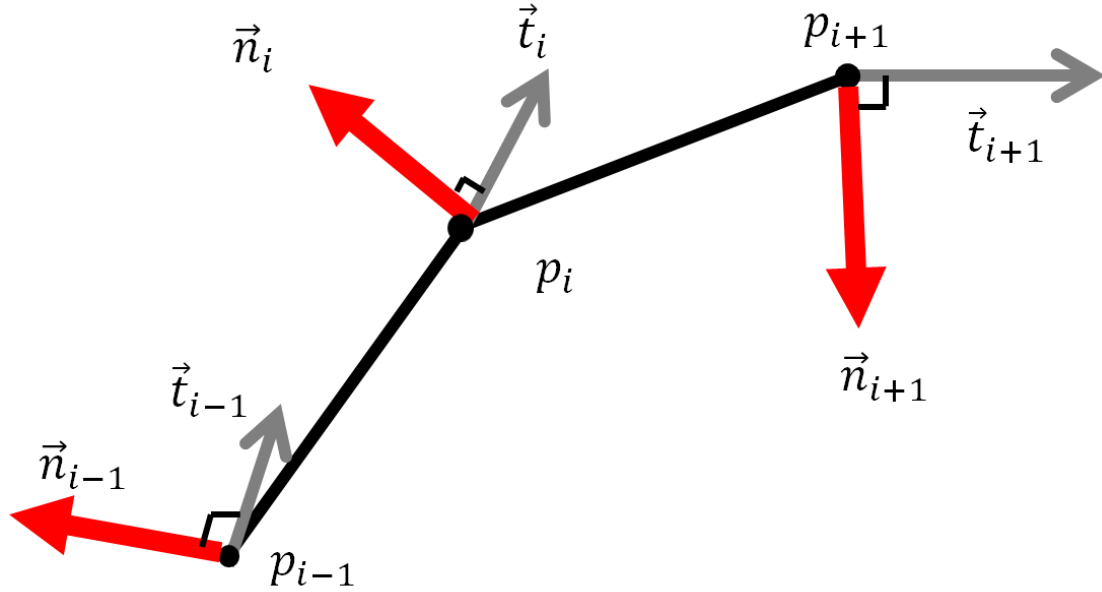


FIGURE 20 – The centerline C with points p , normal vectors \vec{n} , and tangent vectors \vec{t} .

Next, the vectors must be aligned to each other. The first normal can be assigned arbitrarily, and then each proceeding normal will be aligned with respect to the normal before it. This ensures correct concatenation of rings. The following algorithm is applied to each normal vector after the first in order to maintain rotational similarity.

First, define the previous normal vector \vec{n}_{j-1} in terms of its end points p_{i-1} and f_{i-1} . Then the current normal vector \vec{n}_j that needs to be aligned is similarly defined by its end points p_i and f_i . p_i is the current centerline point, p_{i-1} is the previous centerline point, and p_{i+1} is the next centerline point. Conceptually, \vec{n}_{j-1} is normal to the line segment from p_{i-1} to p_i and \vec{n}_j is normal to the line segment from p_i to p_{i+1} . The point

f_i is rotated around its corresponding line segment until the rotation angle $\hat{\theta}$ between \vec{n}_{j-1} and \vec{n}_j is minimized:

$$\hat{\theta} = \arg \min_{\theta} \|f_i - R(\theta)f_{i+1}\|^2$$

Thus, each ring is clipped using the infinite plane π_j with center q_j and aligned normal vector \vec{u}_j . Then, each ring half is joined with all other corresponding ring halves to create a half-colon surface. Finally, these surfaces must be viewed. This is accomplished by assigning a camera to each half-colon. For each point p_i along the centerline, the camera focuses on it from a location determined by the position vector \vec{v}_{pos} . As there are two halves to be viewed, there must also be two cameras. Thus, we will differentiate between position vector $\vec{v}_{pos,1}$ for the vector corresponding to the first half and position vector $\vec{v}_{pos,2}$ corresponding to the second half.

The direction of projection vectors $\vec{v}_{dop,1}$ for the first half and $\vec{v}_{dop,2}$ for the second half are determined by

$$\vec{v}_{dop,1} = \vec{t}_i \times \vec{n}_i$$

$$\vec{v}_{dop,2} = -\vec{v}_{dop,1}$$

Next, the camera position vectors $\vec{v}_{pos,1}$, $\vec{v}_{pos,2}$ and view-up vectors $\vec{v}_{up,1}$, $\vec{v}_{up,2}$ are calculated by

$$\vec{v}_{pos} = p_i + \frac{h_i(\vec{v}_{dop})}{\|\vec{v}_{dop}\|}$$

$$\vec{v}_{up} = \vec{t}_i$$

Where h_i is the desired camera elevation from each centerline point p_i . This elevation is calculated based off the desired FOV angle θ . The equation dictating this is

$$h_i = \frac{D(p_i)}{\tan\left(\frac{\theta}{2}\right)}$$

Where $D(p_i)$ is the distance of a given point to the centerline. Thus at $\theta = 90^\circ$, $h_i = D(p_i)$ and the camera will be located at the edge of the colon opposite the visible half.

Figure 21 shows a visual overview of the method. First, the centerline is divided into segments, figure 21(a). Next, each point on the colon's surface is assigned to a centerline segment, forming rings, figure 21(b). Then arbitrary normals are created for each centerline point, figure 21(c). These normals are aligned with each other, figure 21(d). Finally, the surface is split into two halves, figure 21(e).

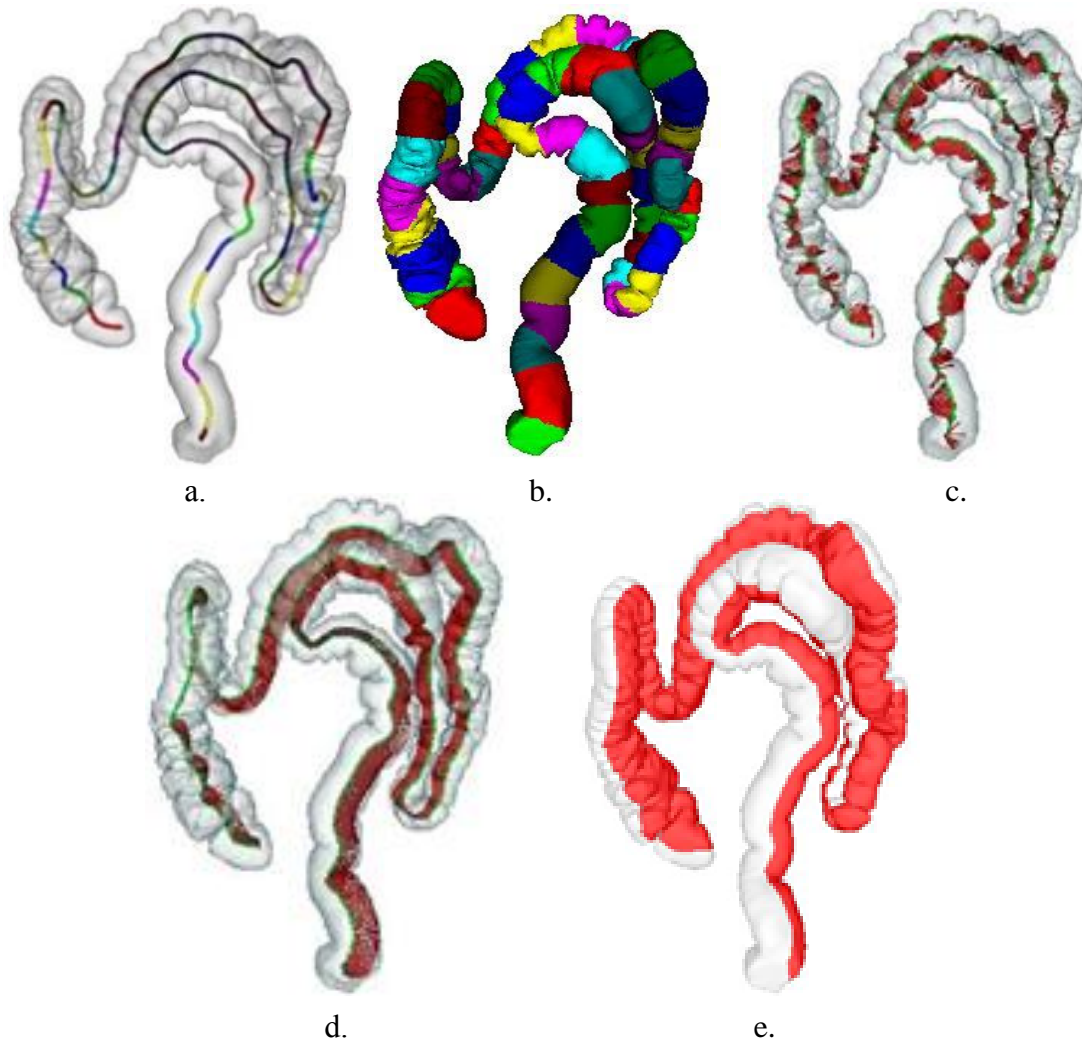


FIGURE 21 - Pipeline of virtual fly-over: (a) Division of the colon centerline into finite line segments. (b) Division of the colon surface into consecutive rings using the surface skinning method. (c) The generated normal vectors along the colon centerline may change their directions abruptly several times. (d) Aligned normal vectors along the colon centerline by the proposed aligning technique, providing a good approximation of the polygonal clipping surface. (e) Splitting colon surface into exactly two halves.

While this method has many advantages, it has several pitfalls that hinder complete visualization. These drawbacks are important to note, as they will be solved through the work of this thesis. The first drawback occurs when a polyp exists along the

cutting plane. The view of the polyp will be split between the two colon halves, and may no longer be recognizable as a polyp.

The next drawback involves the ring generation process. When rings are generated, points are assigned to each ring depending on the nearest centerline. However, in certain sections of the colon that are particularly close together, a point may be assigned to an incorrect ring. Figure 22(a) shows a bend in the colon with an ambiguous protrusion. This protrusion should be assigned to the red ring, but it actually lies closer to the blue centerline and is assigned to the blue ring, figure 22(b).

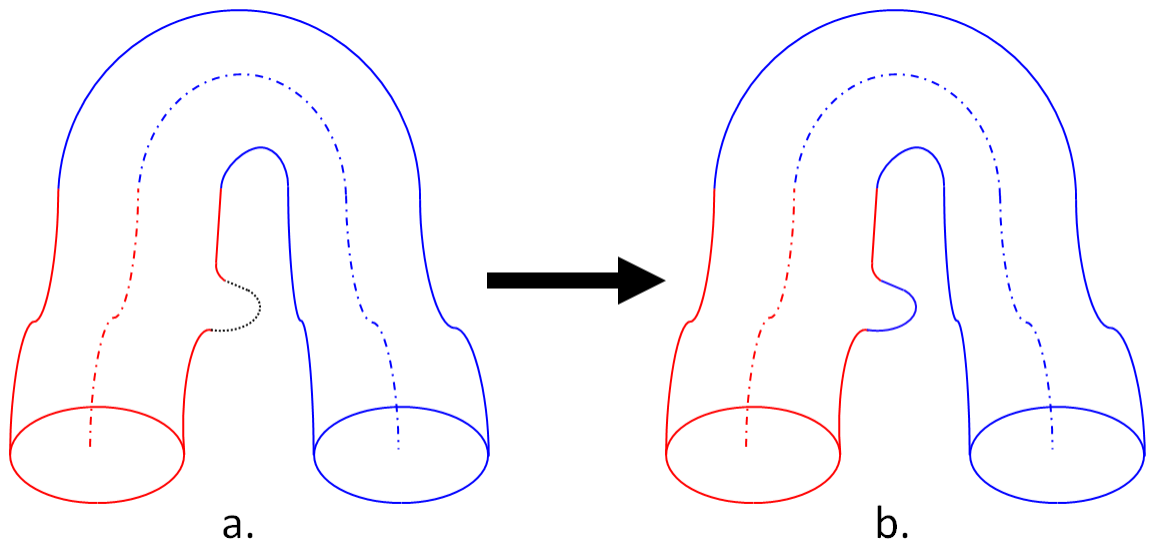


FIGURE 22 - a. The colon is being separated into red and blue rings. One protrusion is ambiguously placed between the red and blue centerlines. b. The protrusion has been mistakenly assigned to the blue ring, as it lies closer to the blue centerline segment.

When the rings are split, this mistaken section can be assigned to the wrong half of the colon. The result is that one half has a piece of the other half, so the wrong camera visualizes that section. The top view in figure 23 has several holes caused by this issue, and the bottom view of figure 23 shows that they have been associated with the wrong

view. These holes are confusing and visually distracting, and a polyp located in one of these places would be missed entirely.

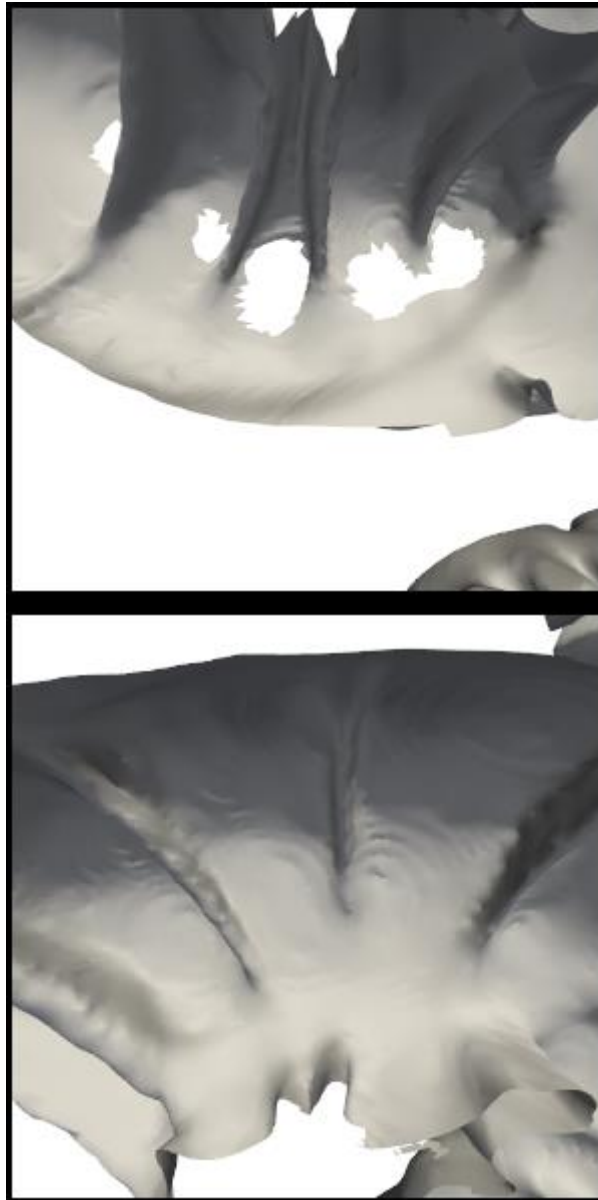


FIGURE 23 - Holes generated by the original fly-over method, where parts of the colon were assigned to the wrong colon half during ring generation.



FIGURE 24 - In the bottom view, the missing segments are visible. As they were closer to the nearby section of colon, they were incorrectly assigned to that section during ring generation.

Finally, there is computation speed. This method takes several minutes to generate the colon halves, often taking five minutes or more. The majority of this computation time occurs during the splitting, as calculations are performed for every point on the colon and every centerline point. There are typically hundreds of thousands

of colon points in these data sets as well as thousands of centerline points, so these calculations come at a significant cost in time.

D. Summary

Fly-through and fly-over visualizations are both essential for understanding the contribution of OST. Fly-through works by moving a camera through the colon, following the centerline. This method has been widely accepted in part due to its similarity to a typical colonoscopy. This familiarity makes fly-through navigation an easy transition for physicians new to virtual colonoscopy. It does have visibility issues with polyps between folds, however. Its overall visibility is around 80%. Despite these issues, it remains an effective technique.

Fly-over navigation was developed by the CVIP lab. It consists of separating the colon into two halves and flying a camera over each half. This has a greater visibility coverage than fly-through navigation, but it is a new technique and is less familiar to physicians. Additionally, previous implementations have had issues with computational complexity as well as incomplete visualizations.

IV. ONE-SIDED TRANSPARENCY TECHNIQUE

A. Concept of One-Sided Transparency

One-sided transparency (OST) is a way to exploit the representation of 3D objects. It allows the inside of an object to be seen from outside, as if the outside were peeled away. The method makes the exterior transparent while keeping the surface interior opaque, figure 25. Through this, OST allows accurate visualization of the inside volume with excellent visibility coverage using a virtual camera.

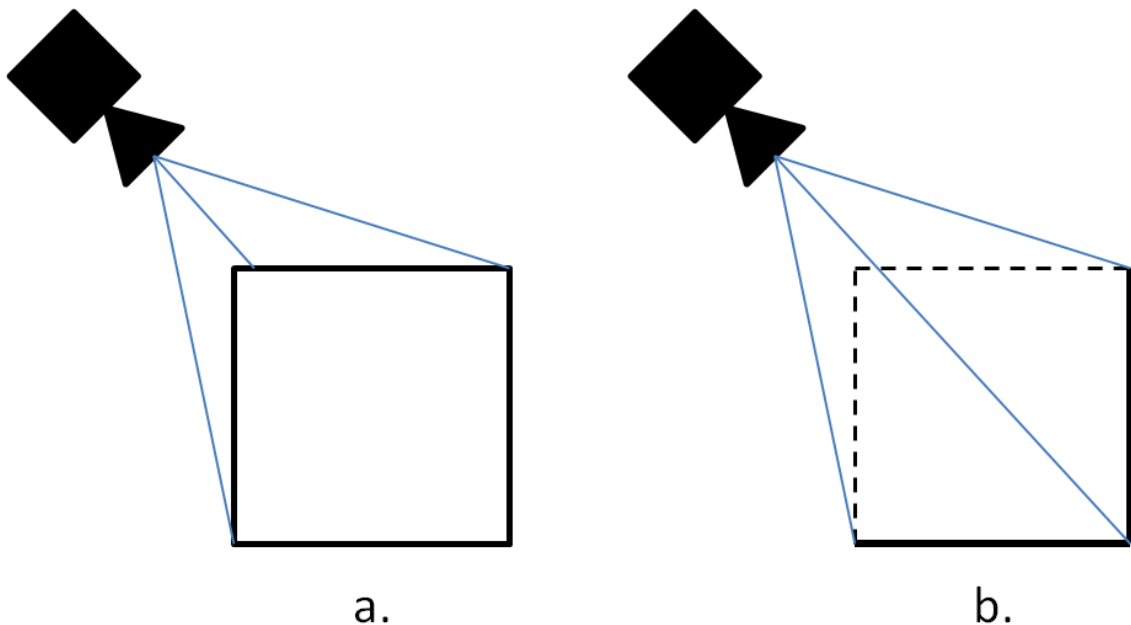
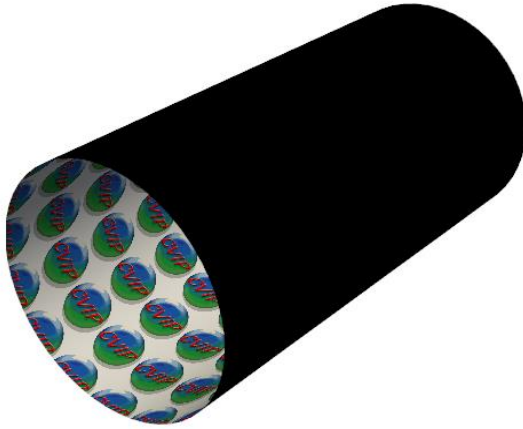


FIGURE 25 - Box visualized a. normally and b. using OST. With OST, exterior faces are not rendered, allowing interior faces to be seen.

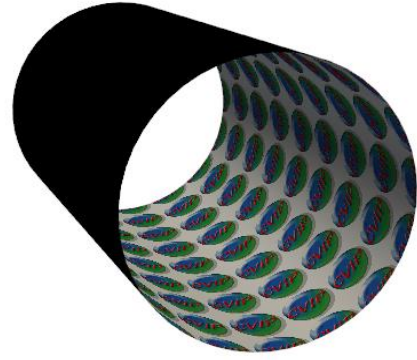
In order to fully understand this technique, it is helpful to see some examples.

Figure 26(a) and (b) show a tube as it would normally appear. The exterior is colored

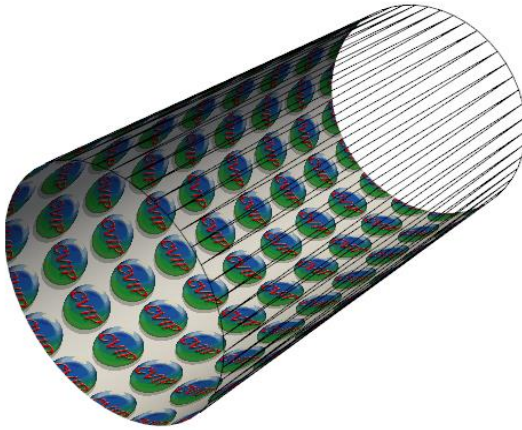
black while the interior is textured with a repeating image. Figure 26(c) and (d) show the same tube with exterior faces culled using OST, and a wireframe placed over the surface to show where the surface would be. Finally, figure 26(e) and (f) show the tube visualized using OST entirely. Note that the exterior black face is invisible, and only the interior texture is visible.



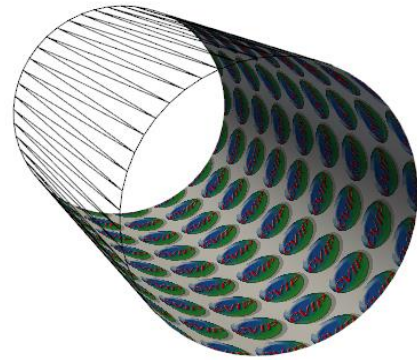
a.



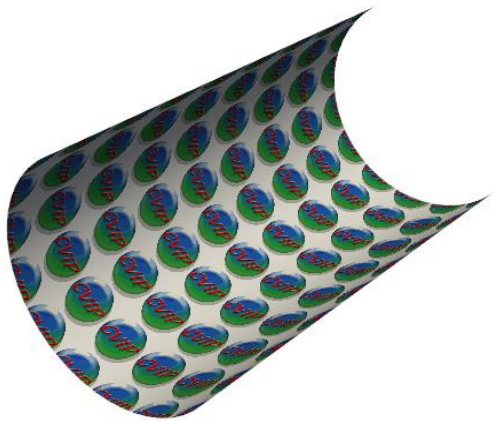
b.



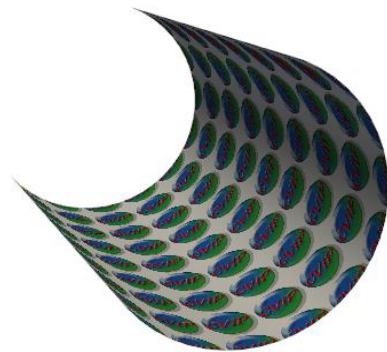
c.



d.



e.



f.

FIGURE 26 - Tube visualized a,b) Normally c,d) Using OST, with a wire mesh showing where the surface would be e,f) Using OST alone

B. Algorithm for One-Sided Transparency

OST exploits back-face culling in order to make the outside faces become invisible and the interior faces become visible. This is done in three steps. First, the model is created such that the normal for each face points towards the inside of the model. Secondly, back-face culling is applied to the model using VTK [18]. Finally, the model is viewed from the outside. Once the model is viewed from the outside, the faces whose normals point away from the camera are culled, leaving only faces whose normals point toward the camera. Unlike normal applications of back-face culling, this will intentionally cull faces that would otherwise be visible. OST reverses the intended application of back-face culling to achieve something fundamentally distinct.

Figure 27 shows a sample scene. In this scene, a front face has clockwise winding and a back face has counter-clockwise face. Thus, the GPU culls faces with counter-clockwise windings, as they are assumed to be not visible to the camera, Figure 27(a). However, when we reverse the normals, we reverse the windings with respect to the camera. This results in the exact opposite effect, making faces that would normally be visible invisible and making faces that would normally be invisible visible, Figure 27(b).

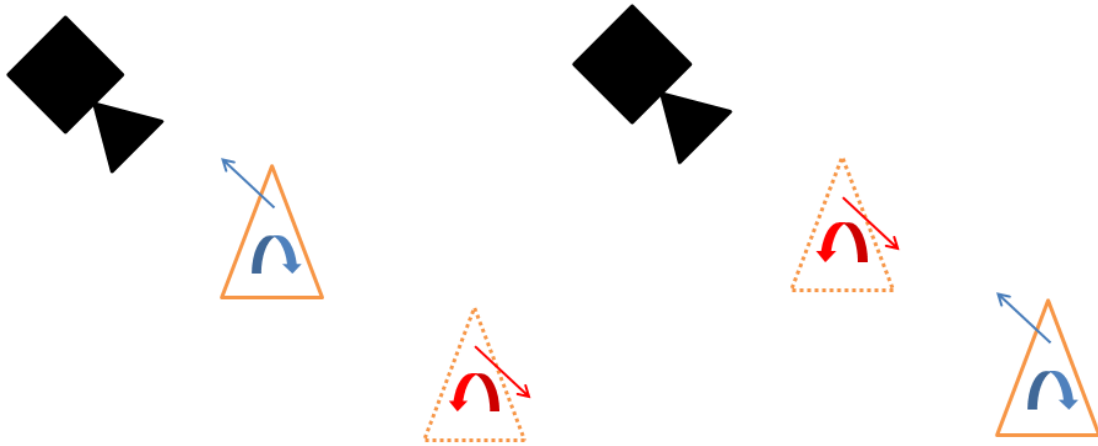


FIGURE 27 - Back-face culling used as intended, to remove faces pointed away from the camera. The clockwise triangle has a normal pointed towards the camera, so it is visible. This represents an exterior face. The counter-clockwise triangle has a normal pointed away from the camera, so it is culled. This represents an interior face, which would not normally be visible. b. Effect of back-face culling after normal orientation is reversed. Now that the normals point toward the interior of the object, exterior faces are transparent and interior faces are opaque.

Figure 28 shows an interesting result of this implementation, where gold should be the interior and blue should be the exterior. A Klein bottle has only one surface, as there are no edges. Thus, there should be neither an “interior” nor “exterior” face. However, applying OST reveals that this model of a Klein bottle does not follow that definition. Practically speaking, there must always be two sides to any visualized object. Without intersecting surfaces, a Klein bottle cannot be visualized in three dimensions [22]. This is because interior and exterior faces are not related to the object as a whole, but rather are locally oriented.

Examining figure 29 gives us an insight into how this Klein bottle was generated. There is a clear break surrounding the hole of the Klein bottle between the blue and golden regions. This indicates that the faces reverse direction along that line. Thus, there is a discontinuity at that point. This is how the Klein bottle is rendered despite being

logically incompatible with the visualization system. Similarly, the Mobius strip in figure 30 has a discontinuity, as the concept of a single continuous one-sided surface is nonsensical in this visualization system.

These examples should make clear that the intuitive meaning of “interior” and “exterior” will not always line up with the implementation of these ideas. Instead, the orientation of each individual face with respect to the camera dictates its visibility. However, in the majority of cases this will not be an issue. Thus, the understanding of OST as rendering the exterior invisible and interior visible is sufficient to realize its applications.

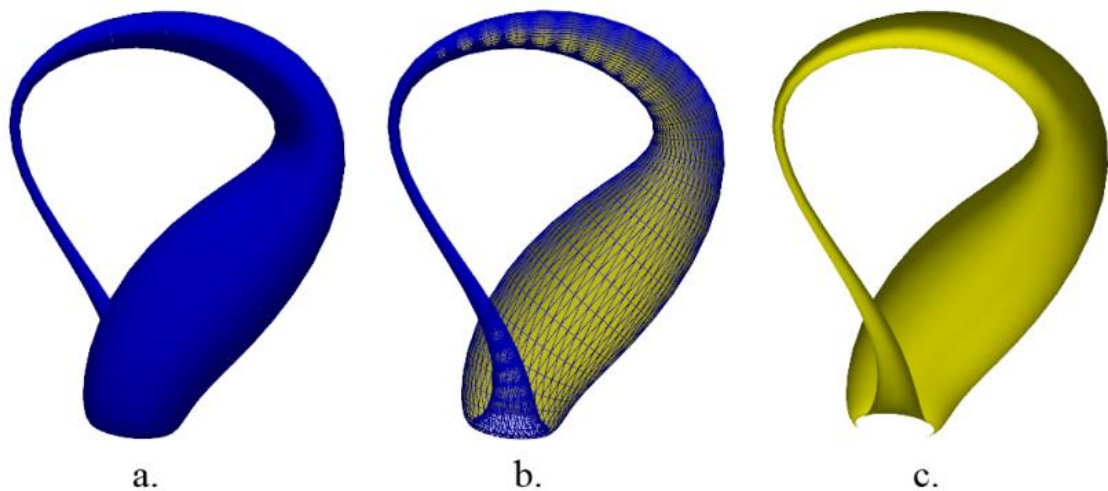


FIGURE 28 - Klein bottle visualized a) Normally b) Using OST, with a wire-frame showing where the surface would normally be c) Using OST alone.

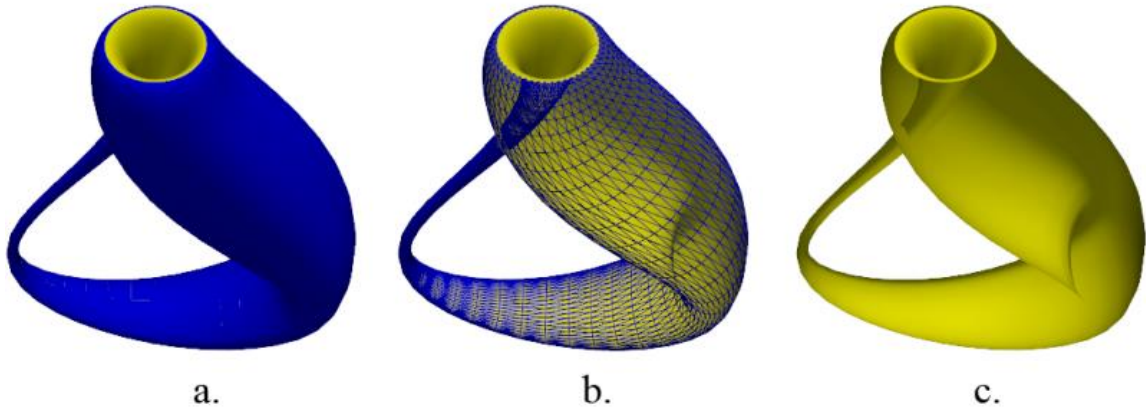


FIGURE 29 - Klein bottle visualized from another angle

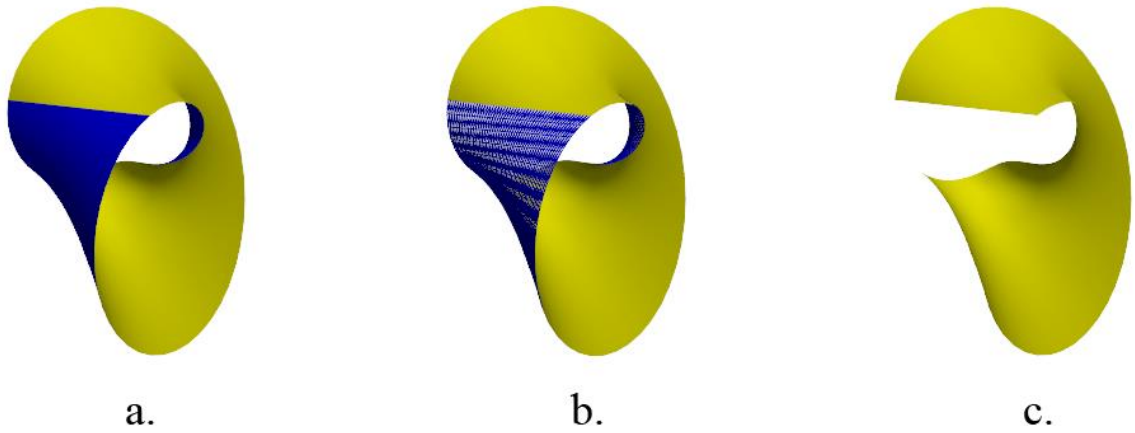


FIGURE 30 - Möbius strip with visible discontinuity revealed by OST

C. Summary

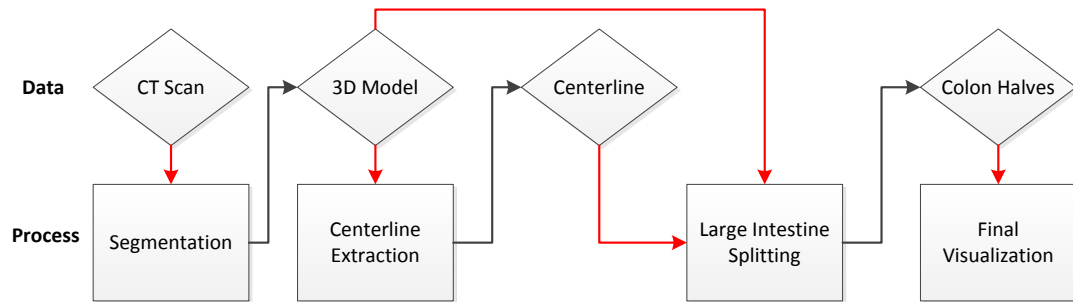
One-sided transparency is a method for exploiting the visualization of 3D objects. It allows the inside surface of an object to be seen from the outside of that object, by using a technique derived from back-face culling. Whether a particular face is interior or exterior can be determined by the orientation of the normal of that face. In order to achieve OST, the normal of each face is pointed toward the interior of the object and then back-face culling is applied.

V. ONE-SIDED TRANSPARENCY IN VIRTUAL COLONOSCOPY

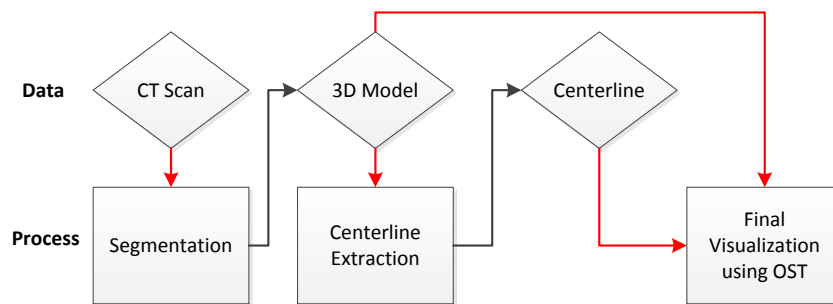
A. One-Sided Transparency in Fly-over

The One-Sided Transparency (OST) method was developed as an improvement over the existing virtual colonoscopy (VC) software used by the University of Louisville's Computer Vision and Image Processing (CVIP) lab. In particular, it was invented to address the complexity and computation time associated with the previous FO method.

As OST was originally developed for VC, there should be no surprise that it finds its most immediate use there. Including OST allows a major portion of the existing FO pipeline to be skipped, figure 31. There is no need to split into halves, so that step can be removed entirely. The work of this thesis adapts the existing fly-over method, with some key alterations. The main difference lies in the normal generation, which is different from the normal generation method previously outlined in [7]. This new method is simpler, equally effective, and provides a smoother viewing by aligning view up vectors to be normal to the centerline. Aligning the view up vectors also allows for easy rotation about the centerline. As this was not useful in the original fly-over, it had not been implemented in previous work.



a.



b.

FIGURE 31 - Pipelines for visualization, beginning with the input CT scans and ending with visualization. Red lines denote input, black lines denote output. a) Original fly-over pipeline. First, the CT scans are segmented. Next, the centerline of the resulting 3D model is extracted. Then that centerline is used to split the model into halves. Finally, each half is individually visualized. b) New pipeline resulting from use of OST. The large intestine splitting is skipped entirely, removing a tremendous amount of computation.

This method is different from what was proposed originally in [7] and proved to be faster while providing the same accuracy. It calculates normal vectors by relating each normal to the previous normal. Simultaneously, the camera setup is prepared for navigation, parts of the camera setup are used in our normal vector computation. In order to take advantage of OST, two cameras are positioned opposite each other with respect to the colon. One camera views each side of the colon, ensuring that the entire colon is visualized, figure 32.

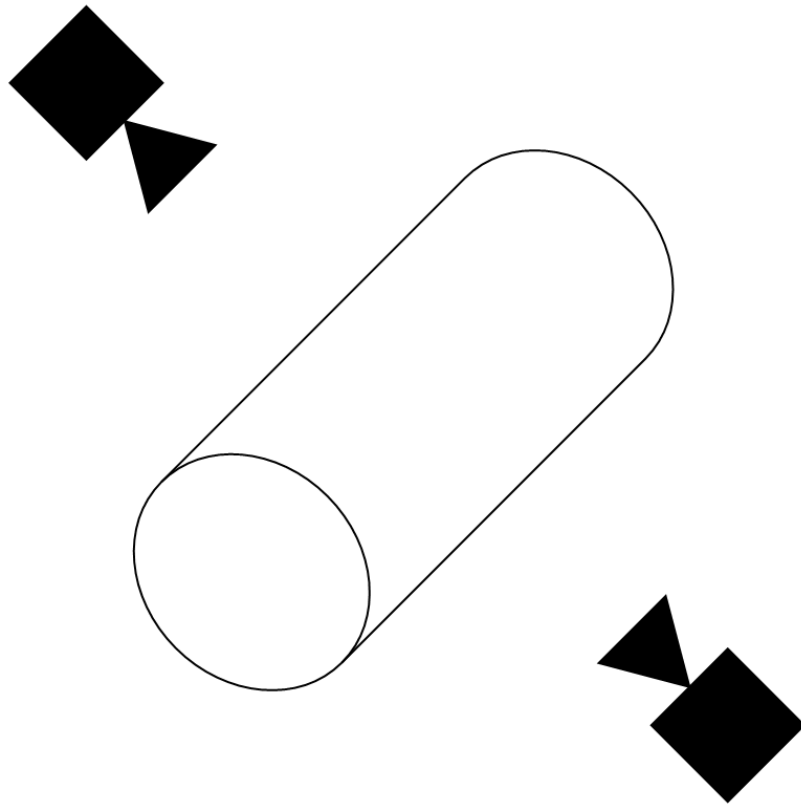


FIGURE 32 - OST-FO camera setup viewing the colon. The cameras are placed opposite each other and each view one half of the colon.

The camera must be located perpendicular to the centerline in order to provide consistent navigation without wild spinning around the centerline. To accomplish this, the normal vector is calculated at each point of the centerline and aligned to the previous normal vector. The first normal vector is determined by finding the vector between the first two points p_1 and p_2 :

$$v_{\Delta 1} = p_2 - p_1.$$

If this vector is not along the Z axis, then an arbitrary normal can be determined by:

$$v_{N_1}(x, y, z) = (v_{\Delta}(y), -v_{\Delta}(x), 0)$$

If it is along the Z axis, then there is a normal vector at

$$v_{N_1}(x, y, z) = (1, 0, 0)$$

The first camera point is then

$$p_{cam_1} = p_1 + v_{N_1}$$

For each normal vector after the first, we must make sure that it is aligned with the previous vector. Figure 33 visualizes this process. First, we use the previous normal vector and the vector from the previous point p_{i-1} to the current point p_i to determine O_i , the offset we need to move the point by:

$$v_{\Delta i} = p_i - p_{i-1}$$

$$O_i = v_{\Delta i} \cdot (p_i - p_{cam_{i-1}})$$

Next, we use this distance to find the new camera point p_{cam_i} by offsetting the previous camera point $p_{cam_{i-1}}$:

$$p_{cam_i} = p_{cam_{i-1}} + O_i \hat{v}_{\Delta i}$$

With this new camera point we can now find the new normal vector as well as the reverse fly-over point:

$$v_{N_i} = p_{cam_i} - p_i$$

As there must be two cameras, the second one is positioned at the reverse camera point $p_{cam_{i,R}}$:

$$p_{cam_{i,R}} = p_i - v_{N_i}$$

In order to allow for rotation around the centerline, each camera point must be aligned so that it is perpendicular to the centerline. Once this is achieved, the camera will be able to rotate in a perfect circle about the centerline. It also keeps a consistent left-to-right traversal of the colon, which provides a stable frame of reference for the viewer. To align each vector to the centerline, a vector that is normal to the centerline is chosen. The

view up vector for point p_{i-1} is shown as V_{up} in Fig. 5. The view up vector V_{up} for point p_i is determined by:

$$v_{\Delta i+1} = p_{i+1} - p_i$$

$$V_{up} = v_{N_i} \times v_{\Delta i+1}$$

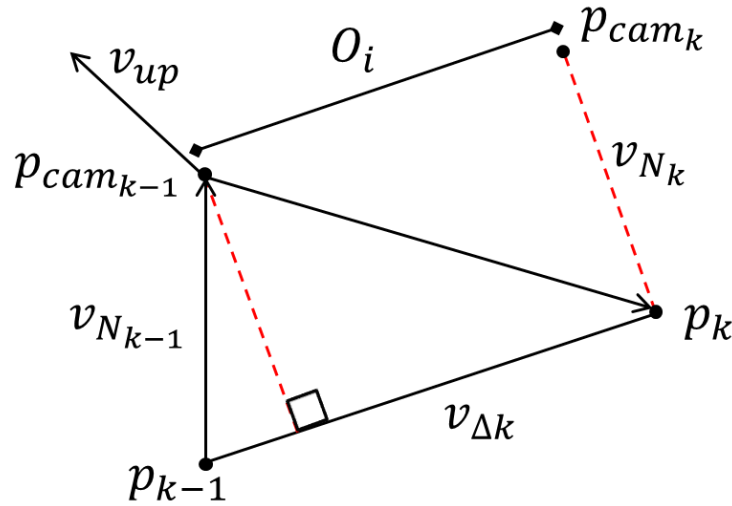


FIGURE 33 - Normal and view up vector computation. p_i and p_{i-1} are points along the centerline. p_{cam_i} and $p_{cam_{i-1}}$ are the camera points above their respective points. V_{up} is the camera view up vector. $v_{N_{i-1}}$ is the previous normal vector, v_{n_i} is the normal vector currently being computed. O_i is the offset distance used to find p_{cam_i} from $p_{cam_{i-1}}$.

B. One-Sided Transparency with the Clipping Plane Method

The clipping plane method is a new method devised to accomplish a similar result to the original FO but without as much computation. It works by modifying the clipping planes of each camera. In VTK, a camera has both a near and far clipping plane [23]. Faces between the camera and the near clipping plane will not be visualized, and faces beyond the far clipping plane will not be visualized. Through VTK, the distance to each

of these clipping planes can be modified. Typically, this would be used to avoid visualizing objects that are very far away, thus speeding up the visualization.

Figure 34 shows a usual clipping plane setup with a camera and several objects. The near clipping plane has been placed almost at the camera, while the far clipping plane is set some distance away. The cube behind the camera is behind the near clipping plane, and thus would not be visualized. The cube in the middle is within the visible range, so it will be fully visualized. The cube in the right straddles the far clipping plane, so part of it will be visualized and part of it will not.

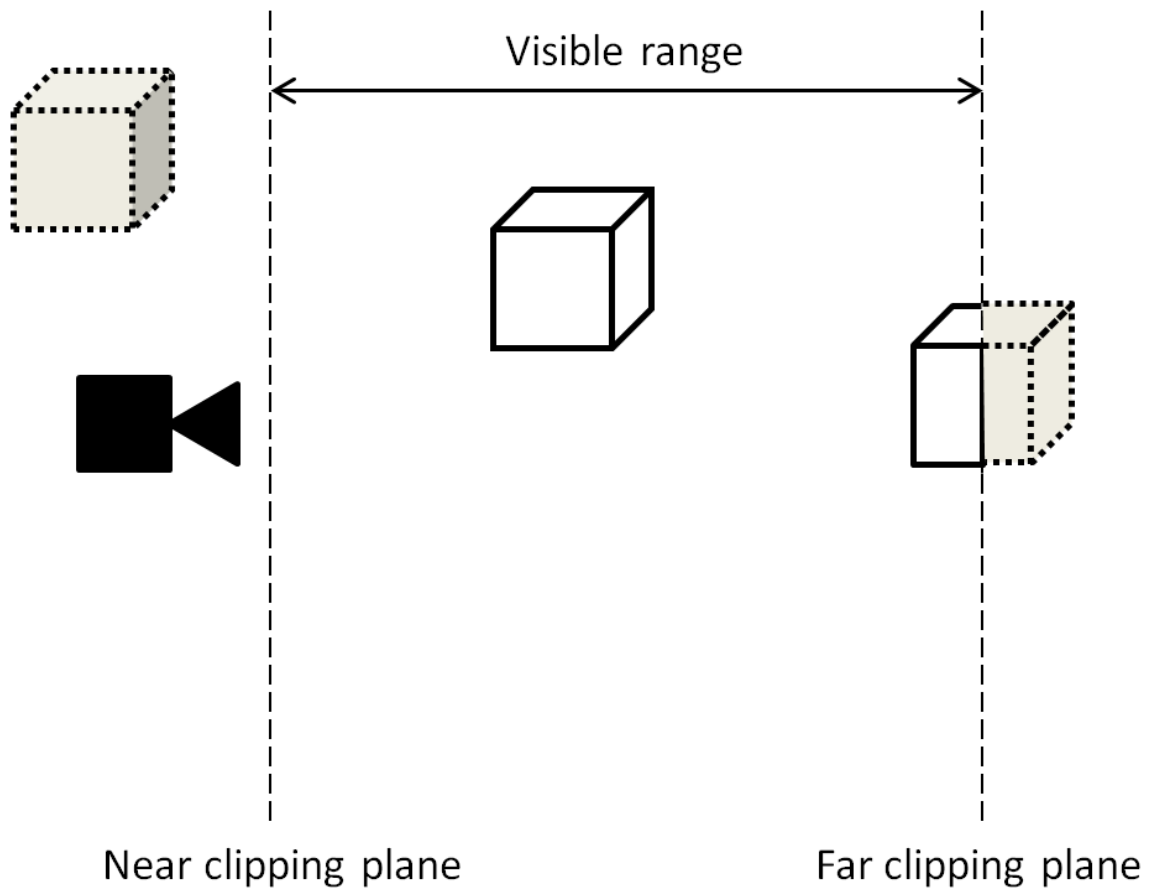


FIGURE 34 - The near and far clipping planes define the visible range of the camera. Anything outside of that range will not be visualized.

Figure 35 shows how the clipping plane method uses an atypical setup to achieve a similar result as the original FO. By placing the near clipping plane at the centerline of the colon, only parts of the colon beyond the centerline will be visible. This is similar to cutting the colon in half and viewing one of the halves, as in the original FO.

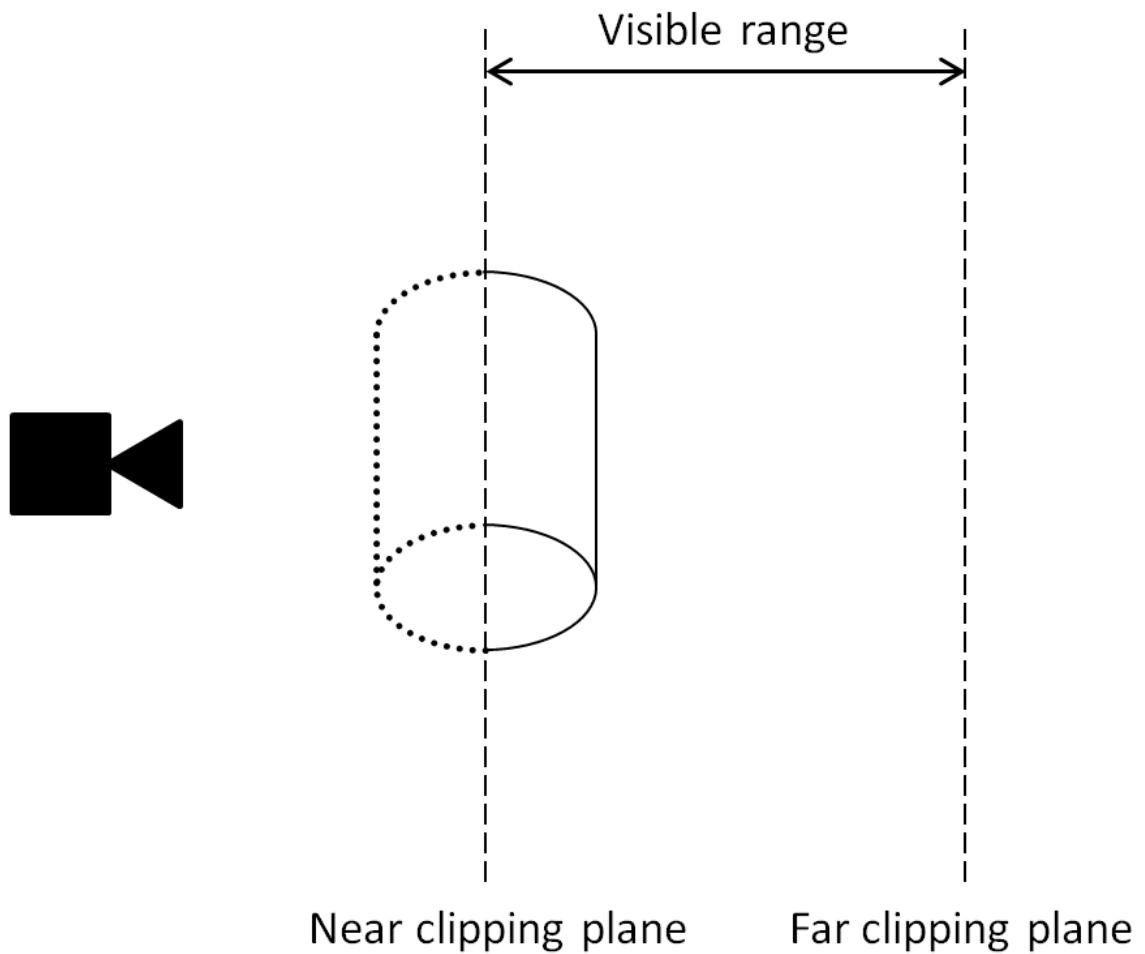


FIGURE 35 - When the near clipping plane is placed inside of the colon, only the opposite side of the colon is visualized. This is functionally similar to the original FO approach.

In order to move the near clipping plane to the point on the centerline that the camera is currently focused on, we must first find a distance d to move it by. This

distance d is calculated using the camera point p_{cam_i} and the corresponding centerline point p_i :

$$d = \sqrt{(p_{cam_i_x} - p_{i_x})^2 + (p_{cam_i_y} - p_{i_y})^2 + (p_{cam_i_z} - p_{i_z})^2}$$

The near clipping plane is placed at that distance from the camera, which locates it directly on the centerline. The result of this is a visualization comparable to that of FO but without additional computation, figure 36. This method shares the computational speed of OST and is somewhat easier to navigate. The best results are obtained when the clipping plane is set closer to the camera than d while OST is applied. This merges the benefits of OST and the clipping plane method while reducing their issues.

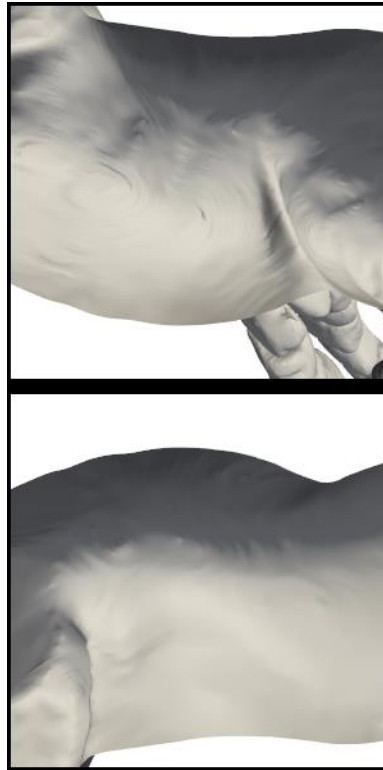


FIGURE 36 - Clipping plane method applied to a section of the colon.

During evaluation of OST, one issue that emerged was occlusion. In the event that the camera was zoomed out far enough such that it was located in another section of the colon, its visibility would be blocked by the interior of that colon, as in figure 37(a). As figure 37(b) shows, applying the clipping plane method will remove this issue entirely.

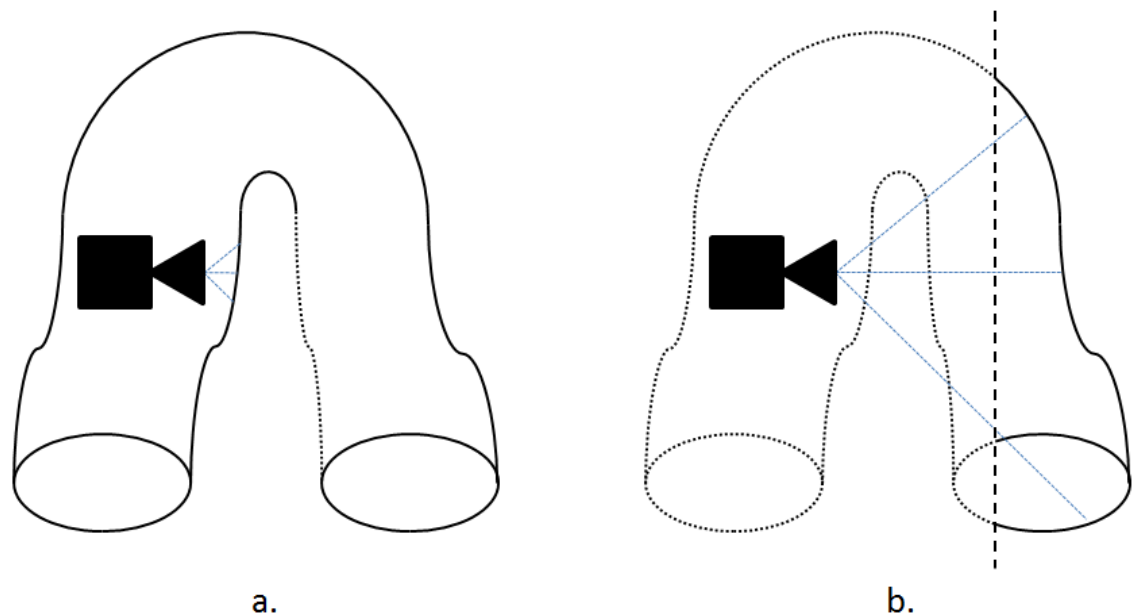


FIGURE 37 - a. OST fails when the camera is located inside of the colon. b. The clipping plane method removes this issue.

While the clipping plane method solves this occlusion issue, it only visualizes a maximum of 50% of the colon at one time, figure 38(b). The clear reconciliation is to partially use the clipping plane method to allow only the section of colon that is currently of interest to be visualized. Figure 38(c) shows the combination of methods, which has OST's greater visibility and the clipping plane's lack of occlusion.

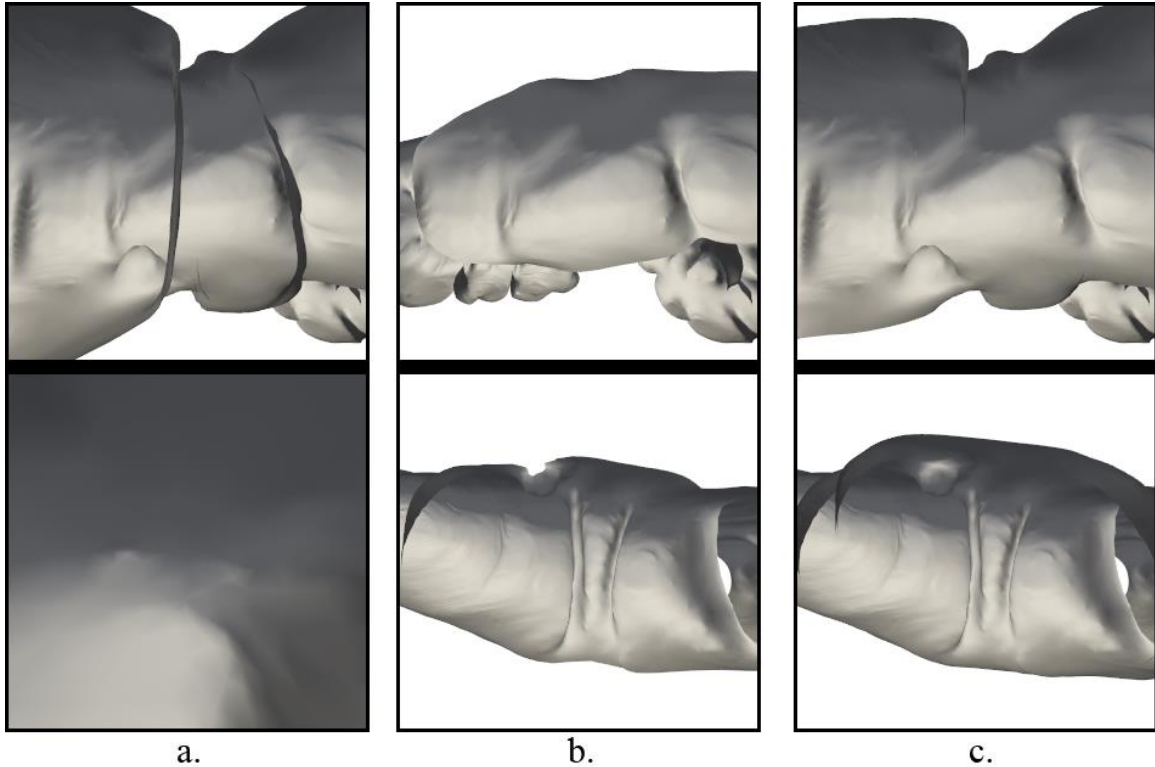
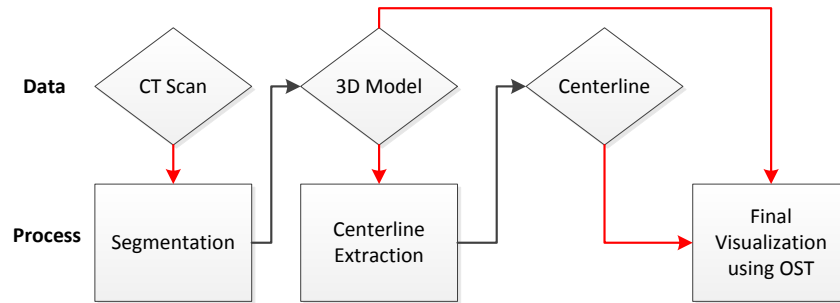


FIGURE 38 - View with camera placed in another section of the colon using a) OST only, with the bottom view occluded by another section of the colon b) Clipping plane only c) OST combined with clipping plane.

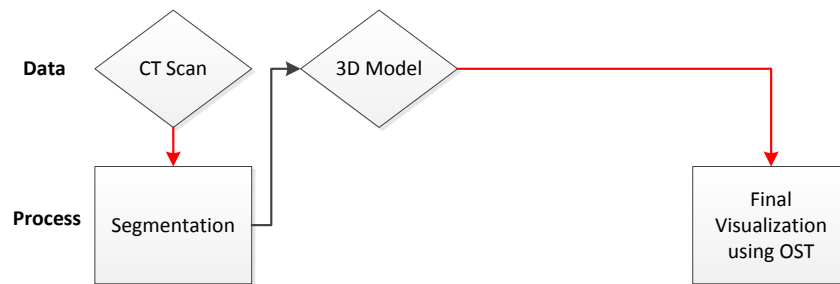
C. One-Sided Transparency without Navigation

More surprisingly, OST has merit in VC without any navigation. In other words, simply visualizing the model and applying OST, as outlined in figure 39. Figure 40 shows what a typical colon would look like when it was normally visualized, and figure 41 shows the same view visualized with OST. While this method is not comprehensive by any means, it nonetheless could have applications. It could serve as a pre-screening

tool, allowing for a quick overview of the colon as a whole. It could also serve as an alternate perspective available during other kinds of navigation. For example, a marker in this view could indicate the current position in the colon.



a.



b.

FIGURE 39 - Pipelines for visualization, beginning with the input CT scans and ending with visualization. Red lines denote input, black lines denote output. a) OST fly-over pipeline, as in figure 6(a). b) New pipeline resulting from direct application of OST.

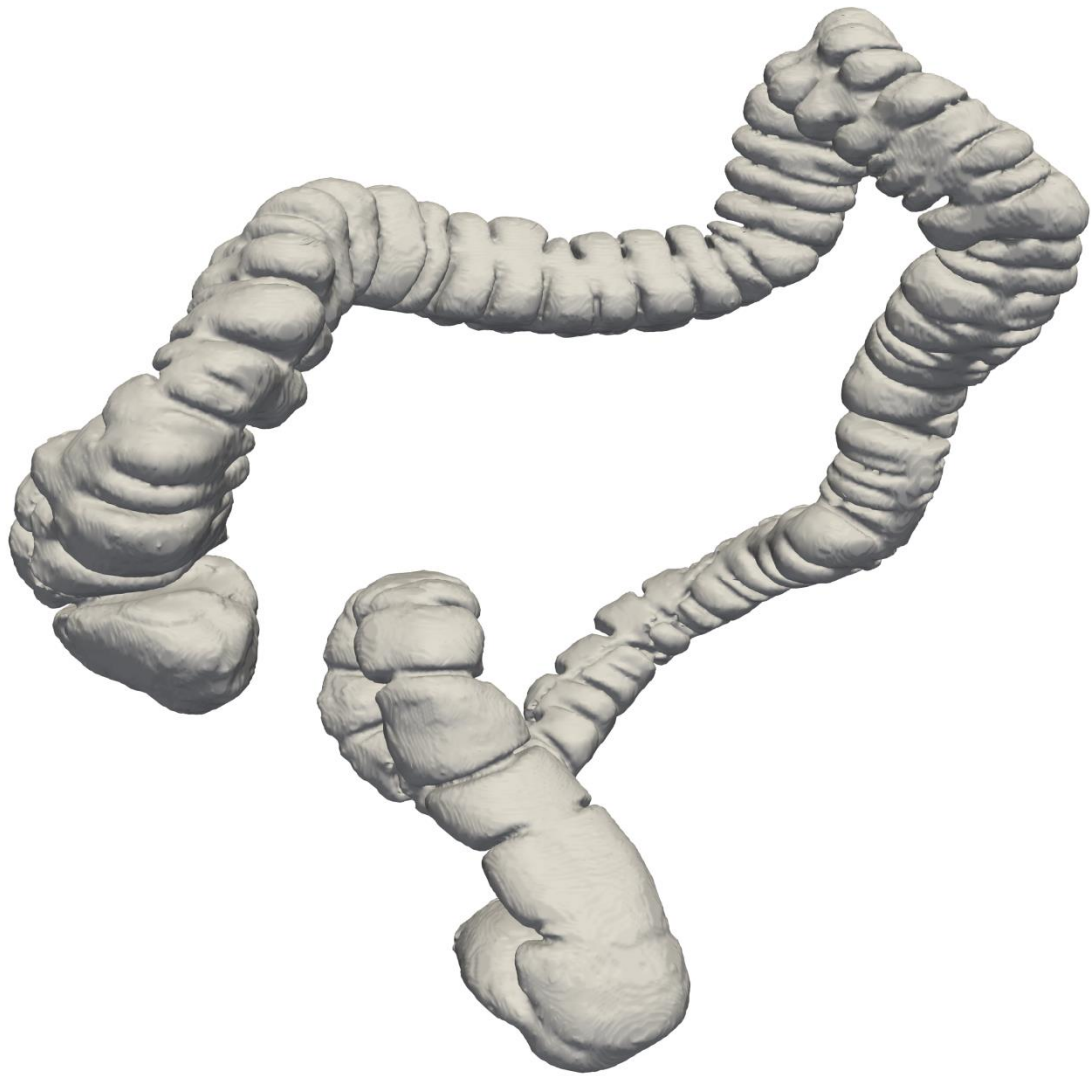


FIGURE 40 - Colon rendered normally



FIGURE 41 - Colon rendered using OST without navigation

D. GUI Interface for One-Sided Transparency in Fly-Over

A GUI interface was created in this work in order to present the work to physicians and for development purposes, shown in figure 42. It includes the ability to move forward and backward by adjusting the speed dial just above the Play button

between positive and negative values. Clicking the button marked “Right” will change the text to “Left” and change the camera direction so that forward is left instead of right. This is believed to be more intuitive than simply moving forward and backward by reversing the speed.

The GUI includes information such as the centerline point location and current angle (if it is not the default). The user can save camera locations and load previously saved camera locations. The user can save screenshots of the current views, which will simultaneously save the camera location. The user can also jump to a particular point, which will move the cameras to the closest viewing point near the target point and then focus on the target point.

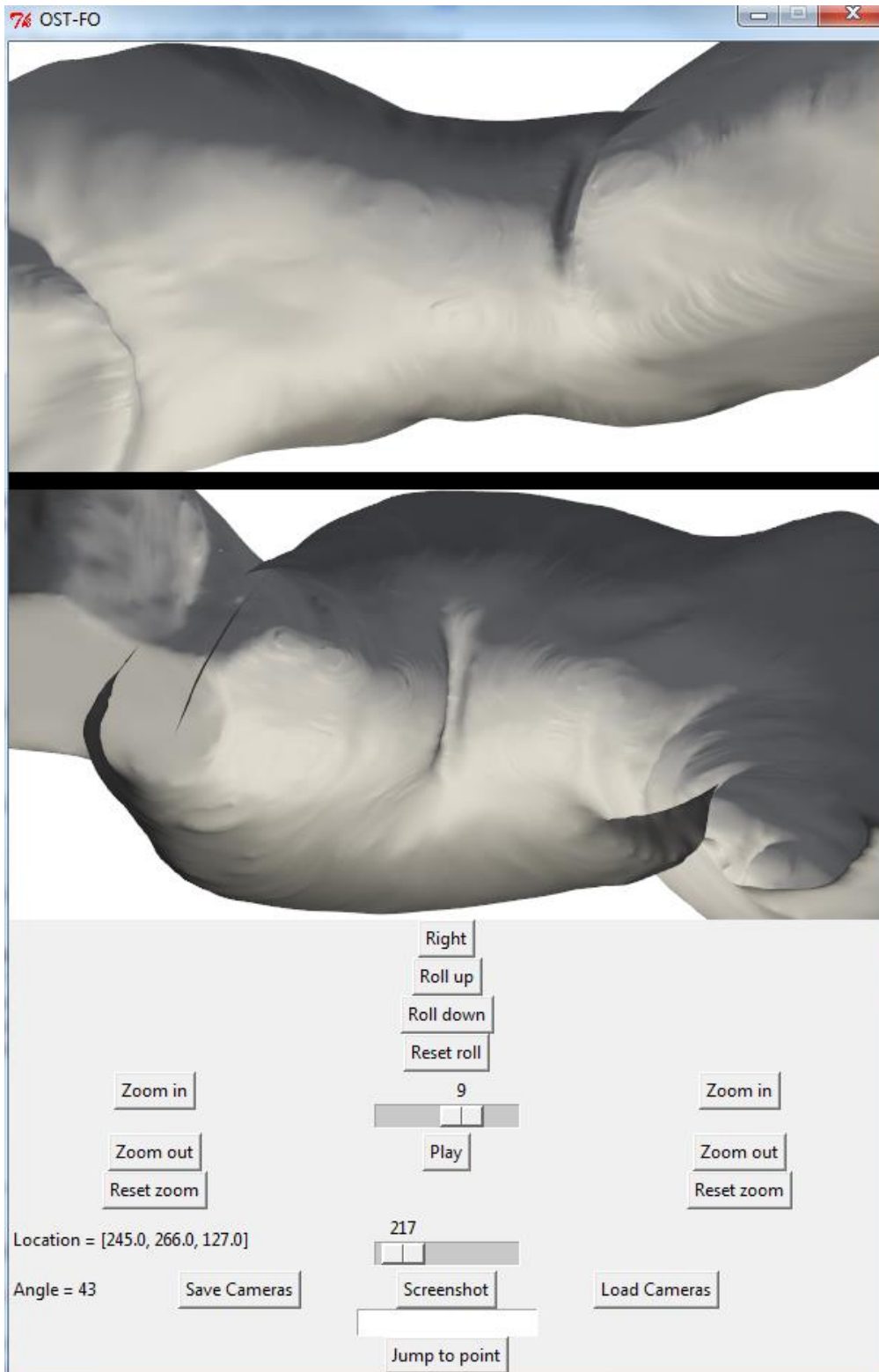


FIGURE 42 - GUI interface developed in Python and Tkinter. This GUI allows for navigation and taking screenshots, as well as reloading previously saved views.

The GUI allows the user to roll about the centerline of the colon, which allows better viewing of the colon, figure 43. If a user wishes to view a suspected polyp from a difficult angle, this allows them to do that easily.

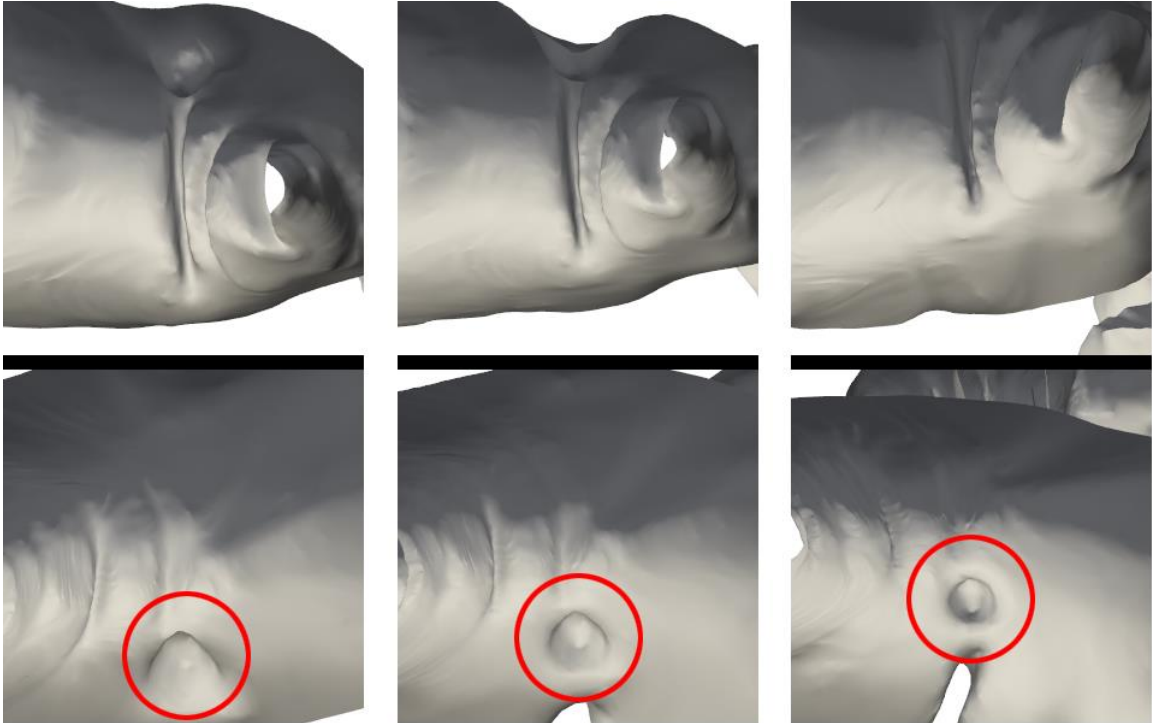


FIGURE 43 – View of a polyp as it rotates into view through use of the Roll commands.

E. Summary

OST reveals several possibilities within VC. As OST originally was developed to augment FO, it naturally works well in that area. It avoids several pitfalls that previous FO work fell into. These pitfalls include holes created during ring generation, slow computation of halves, and difficulty visualizing polyps located exactly on the boundary between halves. A new method for generating normals simplifies computation, allows rotation about the centerline, and provides smooth left-to-right traversal of the colon.

Additionally, the GUI presented in this thesis includes features such as rolling about the centerline, a more intuitive system for movement, and the ability to save screenshots and camera locations for future reference. Finally, applying OST directly to the colon model without navigation creates an interesting view that could serve as a pre-screening tool or as an assistant to navigation.

VI. VALIDATION AND RESULTS

A. Comparison to Fly-through

OST-FO inherits the advantages of FO with regard to FT. First of all, FO covers 20% more of the colon than FT [24]. A large part of that 20% is due to occlusion by folds in the colon wall. A polyp located between two folds could be missed by FT. Also, a polyp that can only be visualized in one direction of FT.

The next advantage that OST-FO inherits over FT is that FT requires four passes through the colon, where FO only requires two. As discussed in the literature review, a polyp on one side of a fold will only be visible in one direction of FT. This means that FT requires two passes through the colon in order to get full coverage of a set of scans. This amounts to four passes in total, as the screening must be performed on both supine and prone sets. This is taxing on the physician, as it requires time and energy to ensure correct screening. In contrast, FO always shows the same view regardless of direction. Thus, FO performed in reverse will show the exact same views as FO performed forward. There is no need for two passes over the same colon set, and in fact there is no particular benefit to it. Thus, FO requires only two passes in total, one supine and one prone.

Figure 44 shows a comparison between polyps when viewed with OST-FO versus when viewed with FT. Figure 44(a,b) show a surface polyp that is visible in both views, and is more prominently visible using OST-FO. Figure 44(c,d) show a polyp located between haustrum at a bend in the colon, which is a particularly difficult location to

detect using VC. Nonetheless, it was easily located using OST-FO. Due to the bend in the colon as well as the location between haustra, FT missed this polyp entirely in both directions. Finally, Figure 44(e,f) show a polyp located on the side of a haustra. This polyp was easily identified using OST-FO. Predictably, this polyp was missed in one direction of FT. While it was visible in the other direction, it did not look like a polyp due to the angle of visualization.

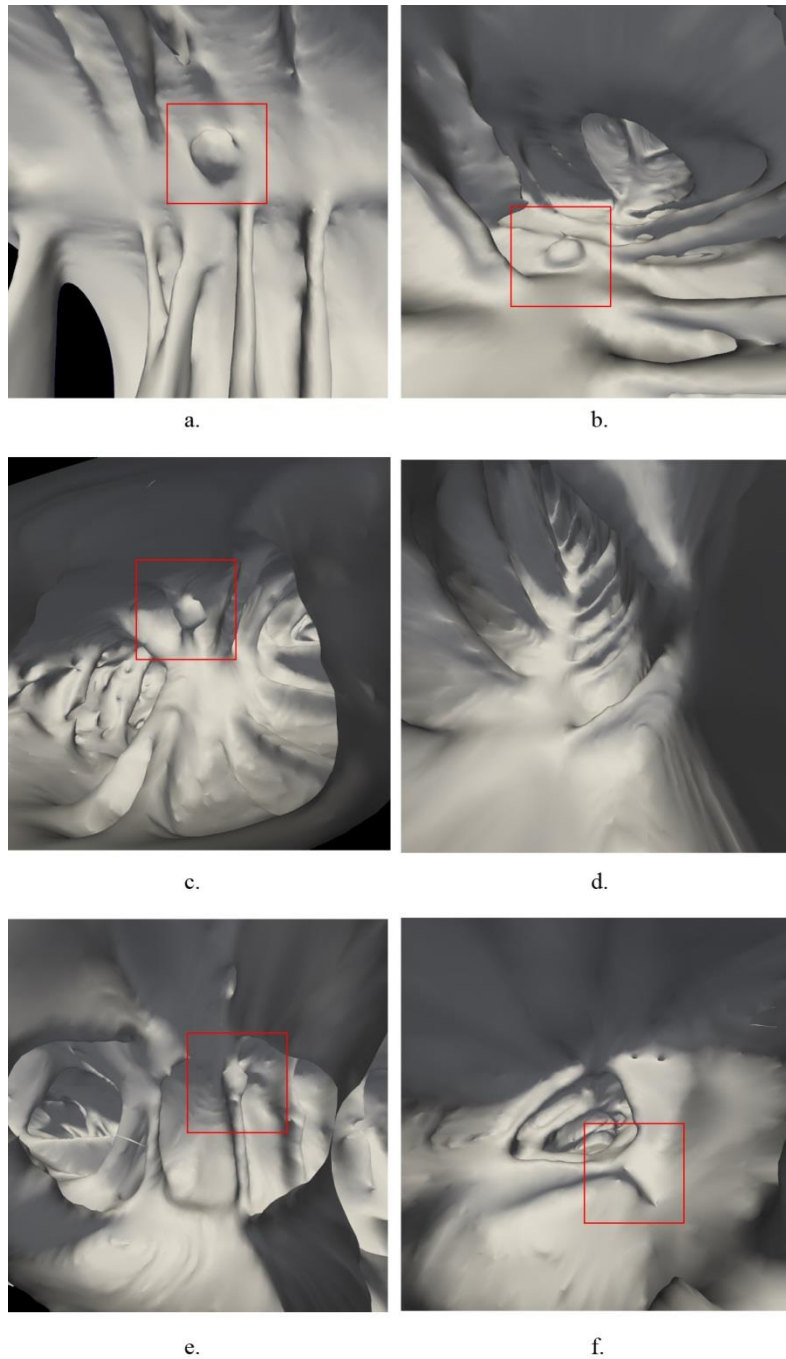


FIGURE 44 - a) OST-FO view of surface polyp. b) FT view of same surface polyp. c) OST-FO view of polyp between haustrum. d) FT view of area with polyp between haustrum, no visible polyp. e) OST-FO view of polyp on haustra. F) Flythrough view of polyp on haustra.

B. Comparison to Fly-over

FO's drawbacks included issues visualizing polyps on the cutting plane, errors in assignment during ring generation leading to holes in the colon, and excessive computation. OST-FO solves each of these issues while simultaneously reducing complexity in the pipeline and adding redundant information.

1. Polyps Located on the Cutting Plane

FO has had issues visualizing polyps on the cutting plane because it divides its information evenly between two views. In other words, there is no overlap between each side of the colon. This could lead to a polyp being split at the cutting line and becoming no longer recognizable as a polyp. Additionally, even if it were detected, it would need to be confirmed. To confirm in the 3D model, the entire cutting process would need to be performed again at a new angle, which would be time-intensive.

At first glance, it may seem that OST-FO can deal with this issue by rolling about the centerline in order to bring the polyp into view. However, OST-FO naturally deals with this issue already. Unlike FO, OST-FO has redundant information between its two views. Figure 45(a) shows a camera viewing the colon from a distance d above the colon surface. Regardless of camera position, it will always show exactly one half of the colon per view at best, with no redundancy between views. In contrast, OST-FO visualizes all of the surface past where the camera's view is tangent to the surface, figure 45(b). At a distance $d = r$, $\alpha = 240^\circ$. This, each camera covers two-thirds of the entire colon, which

is a significant overlap. This is an idealized example, however, and in practice OST-FO has several other behaviors that prove useful for polyp identification.

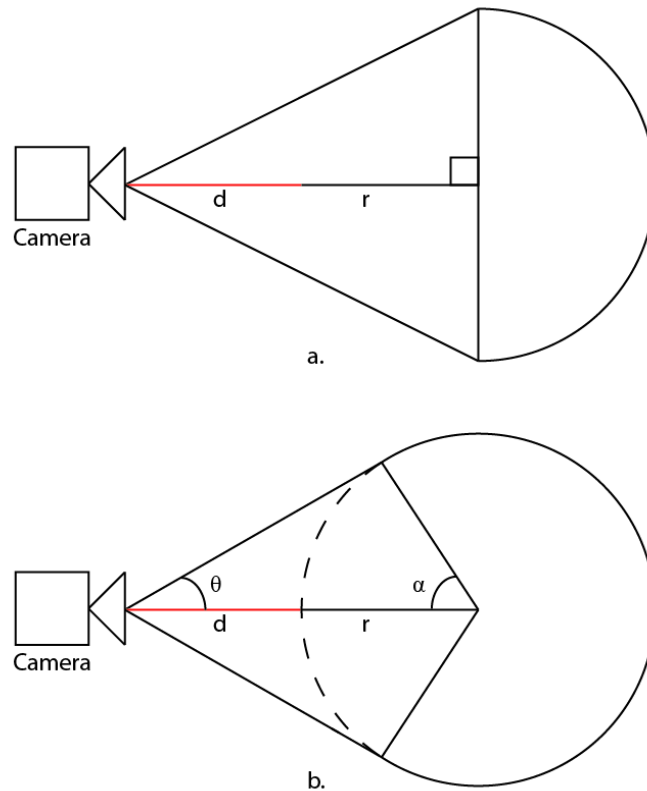


FIGURE 45 – Cross-section of a colon a) Visualized using FO, which shows exactly one half of the colon. b) Visualized using OST-FO, which shows greater than one half of the colon. Increasing the distance d from the colon surface will increase the tangent angle θ , resulting in less visibility. As d approaches infinity, the visibility angle α goes to 90° , the same as FO. Thus, OST-FO will always provide a greater visibility than FO.

Figure 46 shows three polyps visualized using both FO and OST-FO. The center polyp is located directly on the cutting plane for FO, and the OST-FO view has been rotated so that the same polyp is directly on the tangent boundary. Given the idealized example of figure 45, we would expect the polyp to not be visible. However, as the polyp protrudes from the surface, its exterior faces the camera even when the colon's exterior

no longer does. Thus, all three polyps are visible on both camera views. This also helps polyp detection, as they often are floating by themselves in one camera view.

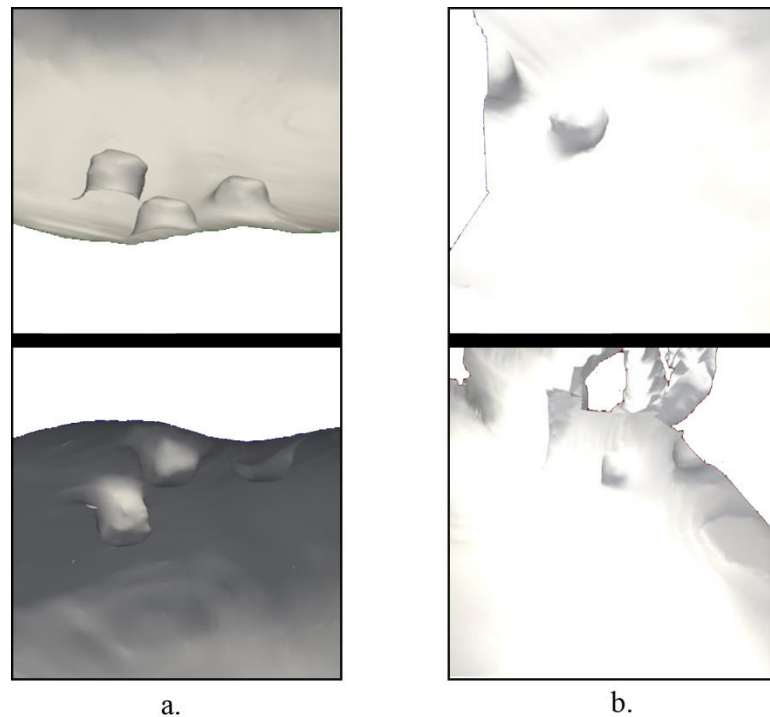


FIGURE 46 – Three polyps located on the cutting plane, visualized using a) FO and b) OST-FO. In a complex region, a polyp that would be split by FO would be clearly visible using OST-FO.

Figure 47 shows polyp visibility in OST-FO when rolling the camera. In figure 47(a), the polyp is visible in both views due to overlap in visibility. In figure 47(b), the polyp is beginning to disappear from one view, but is prominently floating in the air and is fully visible in the other view. Finally, in figure 47(c), the polyp has fully vanished from one view. However, this only occurs when the polyp is directly oriented towards the opposite camera, which ensures that it will always be visible in one of the two views.

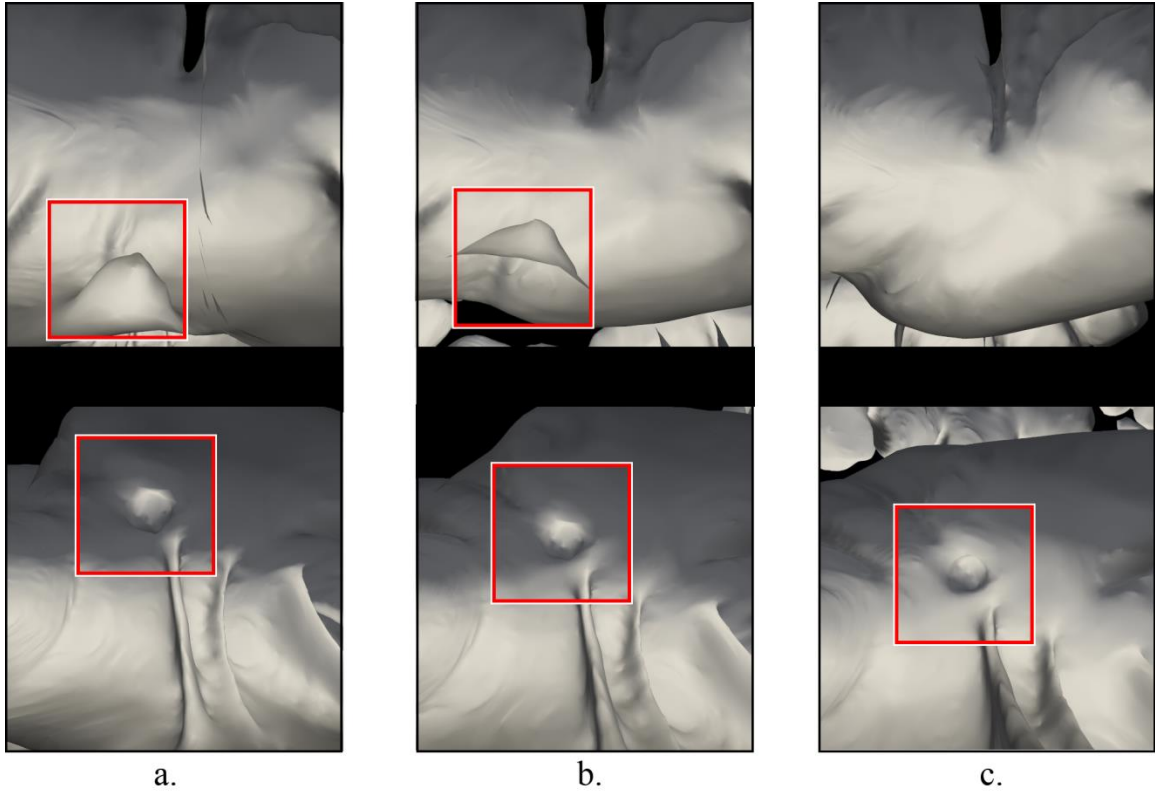


FIGURE 47 - OST-FO polyp visibility while rolling the cameras about the centerline. a) The polyp is visible in both views. b) The polyp is visible in one view and partially visible in the other. c) The polyp is no longer visible in one view, but is fully visible in the other view.

2. Holes in Ring Generation

One major issue with FO occurred during ring generation. Some points on the colon surface may actually be closer to a centerline segment that those points are not encircling, leading to their inclusion in the incorrect ring. This leads to holes in the colon surface. In addition to being visually distracting, these holes could contain important

information such as polyps. This loss of information is inexcusable in a medical application, as it could cost lives.

OST-FO naturally solves this problem due to bypassing the ring generation entirely. Figure 48 shows a side-by-side comparison of FO and OST-FO rendering the same view. While FO has holes, OST renders the entire surface.

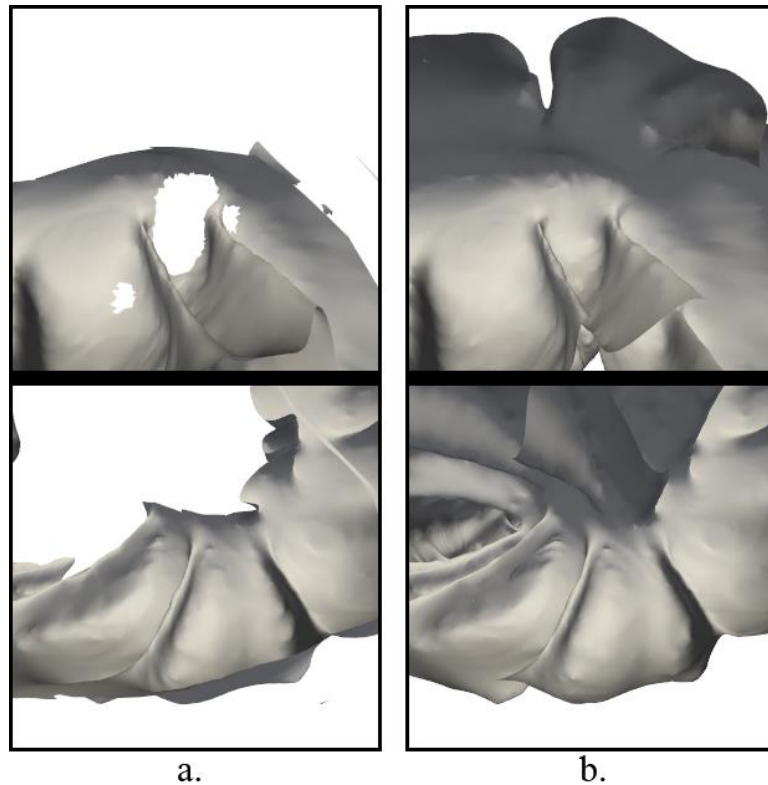


FIGURE 48 - a) Original FO with holes resulting from improper ring assignment. b) Same view rendered using OST-FO.

3. Computation time

One drawback with the previous FO method is the amount of time required to cut the colon into two halves. This was the majority of time associated with opening the

program for the first time. A physician’s time is valuable, and adding five minutes to every screening quickly adds up. As OST removes the cutting step entirely, it sees a 81% increase in speed, saving an average of 98 seconds (1.63 minutes), table 1.

Figure 49 shows the comparison between the existing FO startup time and OST-FO startup time. The OST-FO startup time is much lower across the board. Set 1 was a very small set, so the previous FO technique had a much smaller number of points to iterate over. This meant that the previous FO technique was much faster on this set than on the other sets. As OST-FO does not iterate over the points in the colon, it was only slightly affected by the difference in points. This means that OST-FO would see only slight startup time increases in larger sets, while the previous FO method would take much longer.

TABLE 1 – Startup Time for the Previous Fly-Over Method and One-Sided Transparency Fly-Over

| Set Name | Previous FO | OST-FO | Difference | Percent Time Saved |
|----------|-------------|--------|------------|--------------------|
| 0 | 54 | 15 | 39 | 72% |
| 1 | 129 | 22 | 107 | 83% |
| 2 | 139 | 21 | 118 | 85% |
| 3 | 150 | 22 | 128 | 85% |
| Average | 118 | 20 | 98 | 81% |

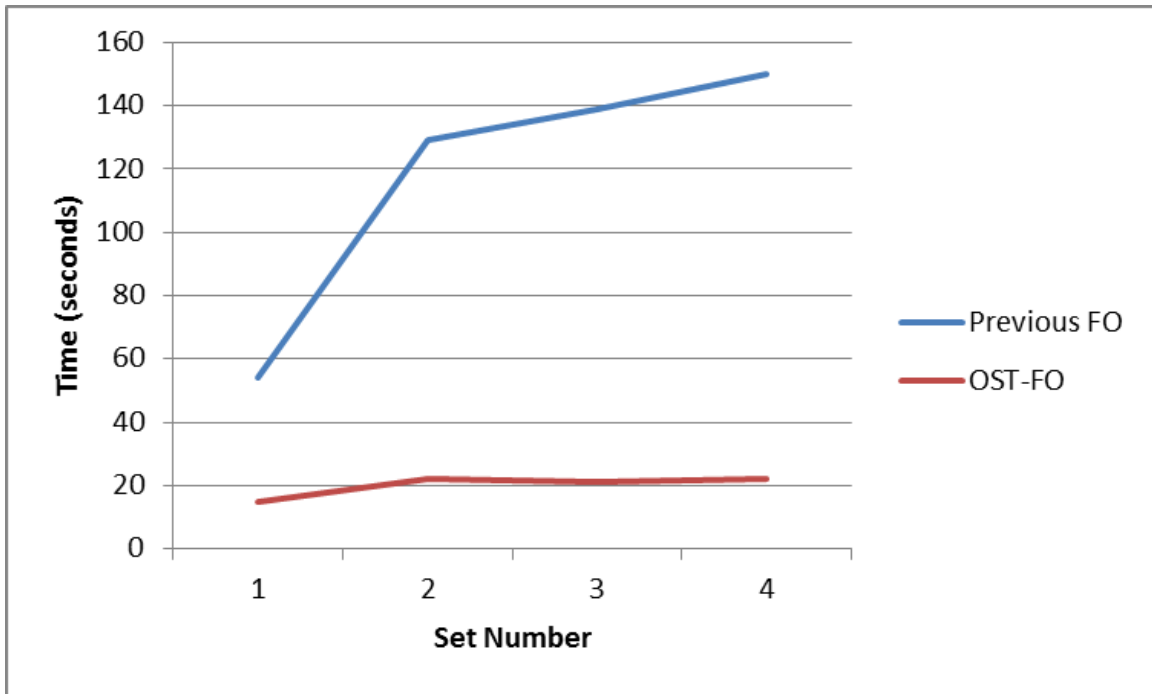


FIGURE 49 - First startup time of previous FO compared to OST-FO. OST-FO has a drastic time reduction

C. Summary

OST-FO has several advantages over existing techniques. It inherits from FO several advantages over FT. FO and thus OST-FO cover 20% more of the colon than FT. FT has a few blind spots that OST-FO does not have, such as a polyp located between folds. In order to minimize blind spots, FT requires two passes through the colon per set. This amounts to four passes total. OST-FO requires only two passes, which is less mentally taxing on the physician and requires less time to perform. Several individual cases have been presented of polyps in difficult locations that OST-FO visualized more clearly than FT.

However, FO does have its own disadvantages. It is computationally complex, and requires several minutes to separate the halves. In contrast, OST-FO requires only seconds from startup to visualization, saving an average of 1.63 minutes. As FO separates the two halves, a polyp straddling the boundary between halves will be partially visualized on each half and may not be identifiable as a polyp. There have been attempts to mitigate this by re-cutting along a different orientation when this issue occurs, but this again is computationally expensive and would take time. OST-FO removes this issue by including an overlap between halves, and allowing for rotation about the centerline. Finally, FO's ring generation led to holes in the colon surface. As OST-FO skips the ring generation step, this is no longer an issue.

VII. OTHER ONE-SIDED TRANSPARENCY APPLICATIONS

A. One-Sided Transparency Applied to Determining Structural Integrity

Inside bridges, ships, buildings, and support structures; structural integrity is key. Potential issues can exist undetected for years within a structure before revealing their issues, sometimes to catastrophic result. Normally, it is impossible to view cracks and other issues that are mainly inside a solid structure. With OST, however, this is trivial. Figure 50 shows a spiral structure with a crack in it. The normal view shows some indentation, but with OST the problem becomes clear. Figure 51 shows the same structure, but with a wholly interior crack. This time, there is no exterior indication of the problem.

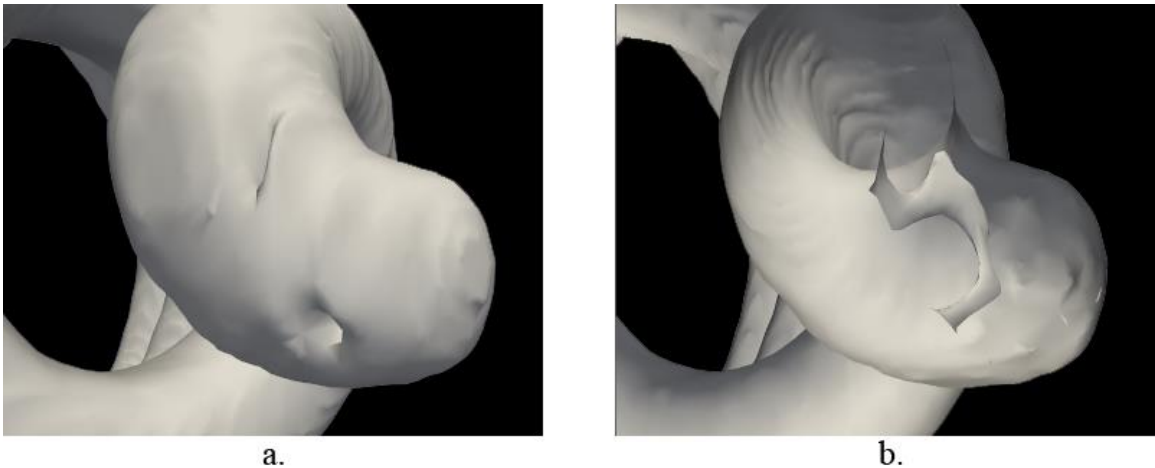


FIGURE 50 – Spiral object with crack shown a) Normally, with visible holes b) Using OST, with the crack fully visible

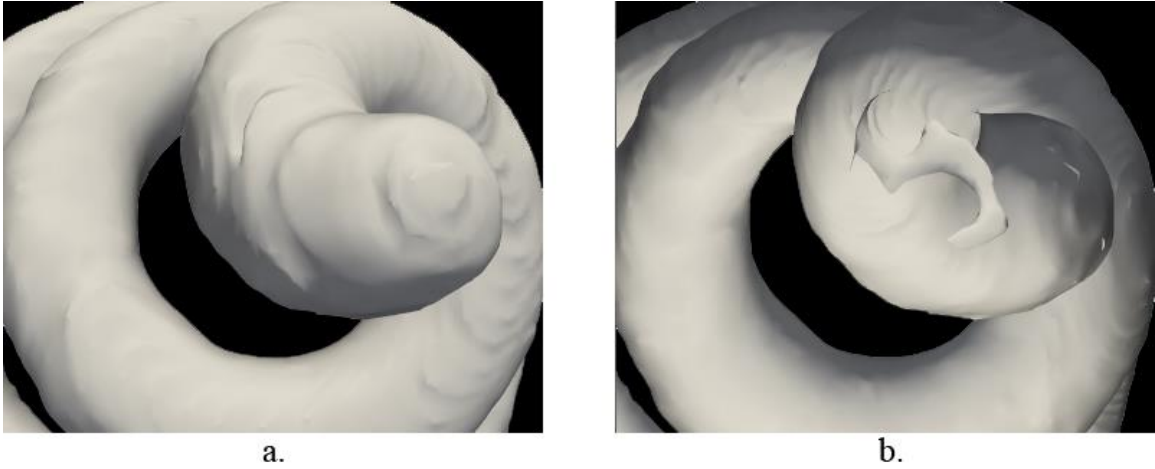


FIGURE 51 – Spiral object with interior crack shown a) Normally, with no visible issues
b) Using OST, with the crack fully visible

Air pockets in a solid structure can be failure points. If we can identify air pockets before they collapse, we can avoid catastrophic failure. OST allows us to see air pockets clearly, which helps us identify points of failure long before they become an issue. Figure 52 shows an air pocket along the length of a tube.

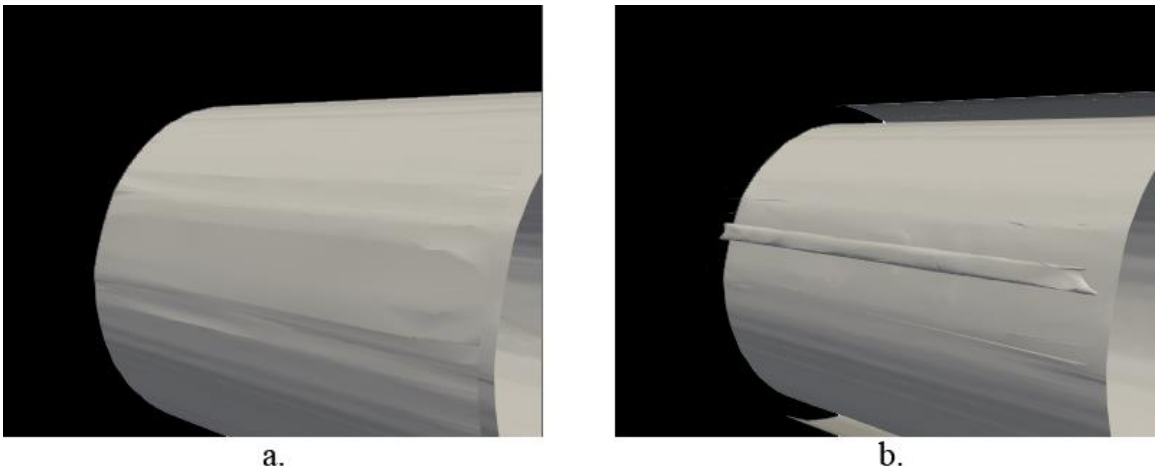


FIGURE 52 – Thick tube with an air pocket, shown a) Normally, with no visible issues b)
Using OST, with the air pocket clearly visible

B. Further Medical Applications for One-Sided Transparency

One possible medical application outside of VC is in dental work. In figure 53, you can see a tooth visualized normally and with OST. The tooth's roots are clearly visible. This visualization could be used in procedures such as a root canal as well as detecting cavities. This inside information could assist in understanding existing issues and recognizing potential future issues. As of now, there is no method for isolating a single tooth as in this example, so this is a distant opportunity. But it is important to recognize the potential for OST in case the opportunity to apply it presents itself in future.

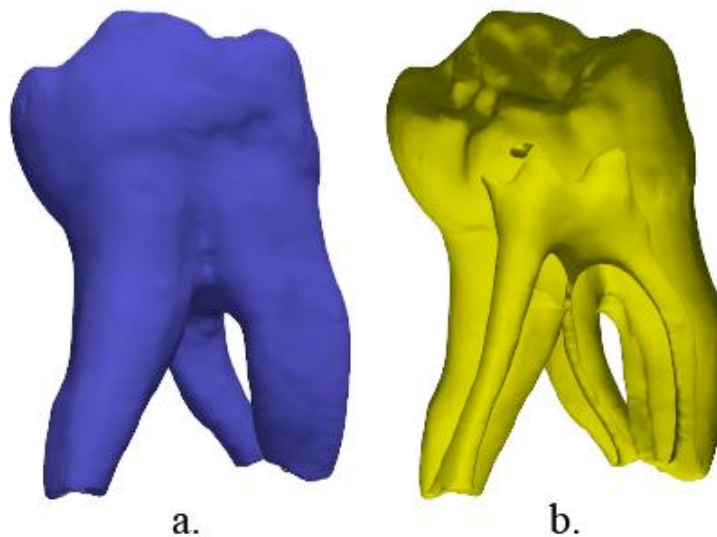


FIGURE 53 – Tooth shown a) Normally, with exterior surface visible b) Using OST, with roots and interior surface visible

There are a few more possible medical applications outside of dental work. Figure 54 depicts a spine viewed from one end using OST. Like the structural integrity application outlined above, this could help find potential points of failure. Finally, figure 55 shows a lung visualized using OST. This could be used to detect growths or other issues, similar to the VC applications. These examples highlight the versatility of OST, as well as demonstrate just a small amount of the potential that this technique has.

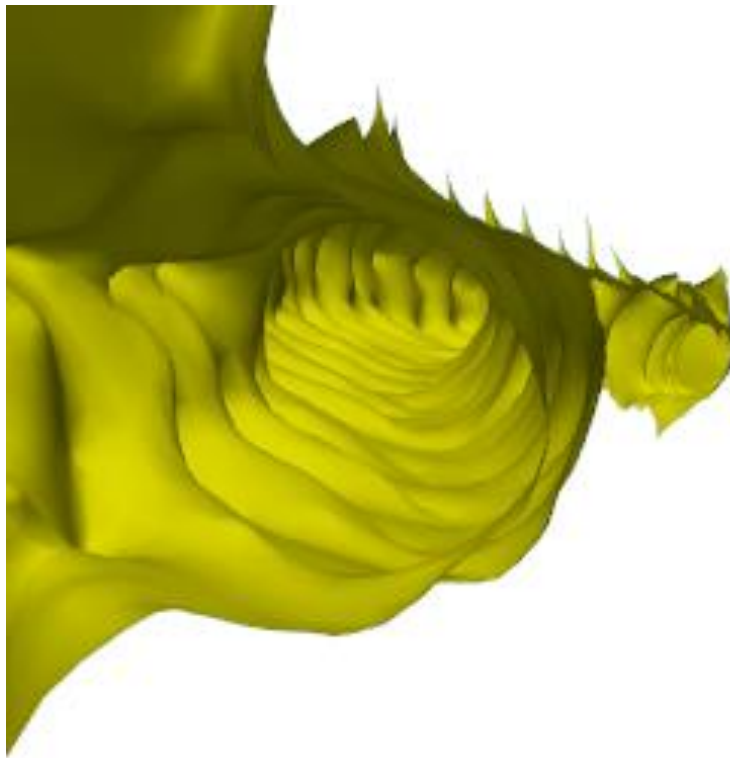


FIGURE 54 - OST applied to the spine and viewed from one end



FIGURE 55 - Lung visualized using OST

C. Summary

OST shows promise outside its intended area. While it was originally created to augment and improve flyover navigation for virtual colonoscopy, it is much more than that. Its opportunities in the medical field outside of VC include aiding in dental work, examining spinal integrity, and detecting other forms of cancer. In structural work, it

reveals interior failings such as cracks and air pockets that could become points of failure.

While these are exciting possibilities, these are only a few applications for a versatile method. OST subverts normal expectations of visualization, and allows a unique view into an object. There are surely countless potential applications for this kind of technique throughout not only medical imaging, but all of engineering.

VIII. CONCLUSIONS AND FUTURE WORK

The OST approach to visualization is a novel solution to the problems presented in this thesis, and also provides potential benefits in other fields. It opens up a new realm of possibilities for the future, while simultaneously solving the issues of the past. When combined with the clipping plane method and fly-over, it is a robust and powerful technique for virtual colonoscopy screening. On its own, it has potential to give greater insight into structural integrity as well as other medical applications.

Now that the fly-over system's major visualization failings have been remedied by the work of this thesis, it is ready for implementation. The next step for this system is to create a comprehensive package that will take in CT scans and output a fly-over navigation. This will also require synchronizing visualization of CT scans with the fly-over navigation. In the final package, the physician should be able to select a potential polyp from the fly-over view and see it in the CT scans, as well as select a potential polyp in the CT scans and visualize it in the fly-over view.

Outside of VC, OST's application to structural integrity can be more thoroughly studied. This thesis used synthetic examples to show its potential use, so the next step would be to acquire 3D models of real objects that include interior structural information. Once it is proven that this can work on real-life objects, the next step would be to apply it to specific situations in which it could be useful. These situations could be anything from bridge piers to ship hulls to the foundation of a building. This research will be limited mainly by the techniques for acquiring this type of data.

REFERENCES

- [1] Lefere, Virtual Colonoscopy: A Practical Guide, New York: Springer, 2010.
- [2] Wikipedia, "File:Stomach colon rectum diagram.svg - Wikimedia Commons," 19 December 2006. [Online]. Available: http://commons.wikimedia.org/wiki/File:Stomach_colon_rectum_diagram.svg. [Accessed 21 July 2014].
- [3] Wikipedia, "File:Colonoscope.jpg - Wikimedia Commons," 1 July 2009. [Online]. Available: <http://commons.wikimedia.org/wiki/File:Colonoscope.jpg>. [Accessed 25 July 2014].
- [4] Y. H. Dachman AH, "Virtual Colonoscopy: Past, Present, and Future," *Radiologic Clinics of North America*, vol. 41, no. 2, pp. 377-93, 2003.
- [5] Hong, Lichan et al., "3D Virtual Colonoscopy," in *Biomedical Visualization*, Atlanta, GA, 1995.
- [6] Jiang, Yuchong et al., "Improved Diagnosis and Navigation for CT Colonography," in *27th Annual International Conference of the Engineering in Medicine and Biology Society*, Shanghai, 2006.
- [7] Hassouna, MS et al., "Virtual fly-over: A new visualization technique for virtual colonoscopy," *Medical Image Computing and Computer-Assisted Intervention*, vol. 9, no. 1, pp. 381-8, 2006.
- [8] Wang, Ge et al., "GI tract unraveling in volumetric CT," in *Visualization in Biomedical Computing*, Berlin, Springer, 1996, pp. 1-12.
- [9] Bartroli, A.V. Vilanova et al., "Nonlinear Virtual Colon Unfolding," in *Visualization*, San Diego, 2001.
- [10] Ebert, David et al., "Virtual colon flattening," in *Data Visualization*, Ascona, 2001.
- [11] Ismail, Marwa et. al, "Model-Based Reconstruction and Visualization in Virtual Colonoscopy," *IET Computer Vision (Under Review)*, 2014.
- [12] Beaulieu C, Jeffery RJ et al., "Display modes for CT colonography part ii: Blinded comparison of axial CT and virtual endoscopic and panoramic endoscopic volume-rendered studies," *Radiology*, vol. 1, no. 212, pp. 203-212, 1999.
- [13] Pickhardt, Perry J. et al., "Computed Tomographic Virtual Colonoscopy to Screen for Colorectal Neoplasia in Asymptomatic Adults," *New England Journal of Medicine*, no. 349, pp. 2191-2200, 2003.
- [14] Lorensen, William E. et al., "Marching cubes: A high resolution 3D surface construction algorithm," in *SIGGRAPH*, New York, 1987.
- [15] Kitware, "VTK: vtkMarchingCubes Class Reference," Kitware, 27 December 2008. [Online]. Available: <http://www.vtk.org/doc/release/5.2/html/a00821.html>. [Accessed 25 July 2014].
- [16] Jmtrivial, "File:MarchingCubes.svg - Wikimedia Commons," Wikimedia, 28 August 2010. [Online]. Available: <http://commons.wikimedia.org/wiki/File:MarchingCubes.svg> . [Accessed 25 July

- 2014].
- [17] Kumar, Subodh et al., "Hierarchical Back-Face Culling," *7th Eurographics Workshop on Rendering*, pp. 231-240, 1996.
 - [18] Kitware, "VTK: vtkProperty Class Reference," Kitware, 24 July 2014. [Online]. Available: <http://www.vtk.org/doc/nightly/html/classvtkProperty.html>. [Accessed 24 July 2014].
 - [19] Y, Ge et al., "Computing the centerline of a colon: a robust and efficient method based on 3d skeletons," *Journal of Computer Assisted Tomography*, vol. 5, no. 23, pp. 786-94, 1999.
 - [20] M. Haussana and A. Farag, "On the Extraction of Curve Skeletons using Gradient Vector Flow," in *IEEE 11th International Conference on Computer Vision*, Rio de Janeiro, 2007.
 - [21] Dachman, AH et al., "CT Colonography: Visualization, Methods, Interpretation, and Pitfalls," *Radiologic Clinics of North America*, vol. 2, no. 45, pp. 347-59, 2007.
 - [22] Kitware, "VTK: vtkParametricKlein Class Reference," 24 August 2011. [Online]. Available: <http://www.vtk.org/doc/release/5.8/html/a01418.html>. [Accessed 25 July 2014].
 - [23] Kitware, "VTK: vtkCamera Class Reference," 22 January 2014. [Online]. Available: <http://www.vtk.org/doc/release/6.1/html/classvtkCamera.html>. [Accessed 25 July 2014].
 - [24] Chen, Dongqing et al., "On clinical validation of Fly-Over visualization technique for virtual colonoscopy," in *16th IEEE International Conference on Image Processing*, Cairo, 2009.
 - [25] Rchoetzlein, "Polygon mesh - Wikimedia Commons," 14 June 2009. [Online]. Available: http://en.wikipedia.org/wiki/Polygon_mesh#mediaviewer/File:Mesh_overview.svg. [Accessed 25 July 2014].