Summer 2018

Virtual reality for the characterization of blood vessel to airway geometric relationships

Mostafa Abdelraouf
University of Iowa

VIRTUAL REALITY FOR THE CHARACTERIZATION OF BLOOD VESSEL TO AIRWAY GEOMETRIC RELATIONSHIPS

by

Mostafa Abdelraouf

A thesis submitted in partial fulfillment of the
requirements for the Master of Science
degree in Informatics (Health Informatics)
in the Graduate College of
The University of Iowa

August 2018

Thesis supervisor: Professor Eric A. Hoffman

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

MASTER'S THESIS

This is to certify that the Master's thesis of

Mostafa Abdelraouf

has been approved by the Examining Committee for the thesis requirement for the Master of Science degree in Informatics (Health Informatics) at the August 2018 graduation.

Thesis committee: _____
Eric A. Hoffman, Thesis Supervisor

_____
Joseph M. Reinhardt

_____
Alejandro Comellas

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Professor Eric Hoffman for the continuous support and guidance throughout my time at the University of Iowa. I thank my colleagues in Advanced Pulmonary Physiomic Imaging Laboratory (APPIL): Junfeng Guo, Amin Motahari, Krishna Iyer, Akar Jani and Mark Escher for their help and insights. Last but not the least, My wife Mariam for her patience, support and continuous encouragement throughout the preparation of this work.

ABSTRACT

An increase in the cross-sectional area (CSA) of the pulmonary arteries has been implicated in the progression of emphysema in COPD patients. Standardization of vessel size requires matching segments of the airway with their corresponding blood vessels. Automated matching is still error-prone, and manual matching by sifting through 2D slices is tedious and time-consuming. We propose a virtual reality (VR) system for the visualization of the airway and the vascular tree as a means of streamlining the verification of appropriate airway/vascular segment pairs selected for quantitation of arterial CSAs. In this work, we outline the technical specifications and design considerations and challenges for such system; we also compare user's performance on the proposed system with the conventional 2D method.

PUBLIC ABSTRACT

An increase in the cross-sectional area (CSA) of the arteries of the lung has been implicated in the progression of emphysema in chronic obstructive pulmonary disease (COPD) patients. Further investigation of this effect requires accurate measurement of the blood vessel sizes and matching these vessels with segments of the airway. Automated methods for matching exist, but they are not very accurate. Manual matching is more reliable but is time-consuming and hard. In this study, we investigate the use of virtual reality for matching airway to arteries and whether or not it makes the process easier and more accurate by comparing it against a conventional matching method. We also discuss the technical challenges faced when developing such system.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Problem Statement and Motivation

Peripheral pulmonary vascular abnormalities have been implicated in the progression of emphysema in COPD patients, particularly, an increase in the cross-sectional area (CSA) of the pulmonary arteries [58]. Expanding this work entails the investigation of this effect by matching segments of the airway with their corresponding blood vessels to normalize the arterial cross-sectional area across subjects in large multi-center studies [20] [121]. Fully-automated blood vessel to airway matching is still error-prone [59], and manual matching involves the tedious and time-consuming process of sifting through 2D slices and visually tracing individual segments of the airway and its surrounding blood vessels. This work showcases a virtual reality (VR) system built for the visualization of the airway and the vascular tree as a means of streamlining the verification of appropriate airway/vascular segment pairs selected for quantitation of arterial CSAs associated with RB10 and LB10. Technical specifications and design considerations for the system as well as the challenges encountered while building the system are outlined in this work.

1.2 Background and Related Work

1.2.1 Implications of Arterial Cross-sectional Area on Emphysema

Pulmonary vascular abnormalities act as a precursor to the damage that occurs in emphysema [3] [52] [48] [58] [5]. An increase in the ratio between the pulmonary

artery cross-sectional area (CSA) to aorta CSA (PA:A) corresponds to an increase in the occurrence of acute exacerbation of COPD [138]. Matsuoka et al. showed that this correlation exists with smaller blood vessels as well stating that the "total CSA of small pulmonary vessels at sub-subsegmental levels strongly correlates with the extent of emphysema" [82]. Iyer et al. [58] demonstrated that the airway normalized pulmonary arterial segments were larger in emphysema susceptible smokers and this was reversed with sildenafil.

1.2.2 Airway-to-artery Matching

Before discussing the airway-to-artery matching problem, it is helpful to briefly review the normal relationship between the airway and the vascular tree. The lung receives blood supply from two sources: bronchial vessels and pulmonary vessels. Bronchial vessels are small high-pressure arteries ($\approx$ 2mm in diameter at their origin) that originate from the systemic circulation; they don't follow the airway branching pattern [134]. Pulmonary vessels are pulmonary arteries and veins. Pulmonary arteries are large low-pressure arteries that originate from the pulmonary trunk (coming off of the right ventricle). There are two types of branches associated with the pulmonary arterial tree: conventional and supernumerary branches. Conventional branches follow the branching pattern of bronchi, and they are the primary object of interest for airway-to-artery matching. Supernumerary branches are smaller and more numerous than conventional arteries; they typically arise from their parent artery at about a 90-degree angle. The supernumerary arteries do not accompany the bronchi; the ratio

between supernumerary arteries to conventional arteries is around 2.8:1, but it can be higher in pulmonary acini [98] [51] [137]. Pulmonary veins carry oxygenated blood; they loosely follow the airway [137]. Towards the hilum, they take an independent route [85]. Supernumerary veins are also present, and their ratio to the conventional veins is around 3.5:1. This topological and geometric complexity coupled with wall fusion in reconstructed CT images due to partial volume effects make matching arteries to airway segments a non-trivial problem.

The airway-to-artery correspondence problem can be approached as a tree matching problem. In tree matching, the computer algorithm tries to draw a correspondence between nodes in two or more trees. Tree-matching is used to perform intra- and inter-subject airway-to-airway matching, intra- and inter-subject artery-to-artery matching or intra-subject airway-to-artery matching. Below are a few examples in the literature that tackled these problems.

For airway-to-airway matching, Kaftan et al. developed a path-based tree matching algorithm that compares paths within the airway and vascular trees using three features, namely, distance, angle and distance variance. They reported around 87% airway-to-airway matching accuracy [62]. Feragen et al. developed an algorithm for airway-to-airway matching based on the premise that the distance between two "tree shapes" can be represented in "tree space" as the shortest deformation connecting the two "tree shapes" [32].

Few works tackled the airway-to-artery matching problem. Bulow et al. developed an automatic airway-to-artery matching algorithm for their work to perform

automatic extraction of the pulmonary artery tree, they introduced a metric they named "arterialness" which is based on the fact that the arterial tree follows the airway tree. For example, if a vessel has a nearby bronchus that is almost parallel to the vessel, the "arterialness" metric for that vessel will be high. They reported that their algorithm was more effective for intermediately-sized vessels [10]. More recently, Jin et al. used fuzzy distance analysis, connectivity analysis, and orientation to establish the correspondence between an airway segment and a matching artery [59].

1.2.3 Virtual Reality

To aid the technologists as they seek to verify linkage of artery to airway segments and to facilitate interactive identification of the appropriate correlation of anatomic locations, we have turned to the use of newly emerging virtual reality hardware and software. Perhaps the earliest conceptualization of virtual reality came in 1965 with Sutherland paper titled "Ultimate Display." In that paper, he described an environment where the "computer controls the existence of matter" [124]. Such "display" would, indeed, be the "Ultimate display" but the state of technology is much less advanced.

In essence, a virtual reality system will have to establish a virtual environment (VE) for the users and receive feedback to achieve what is known in virtual environments literature as "Presence." Presence is the sense of "being there" in the virtual environment [53] or in other terms "The subjective experience of being in one place or environment, even when one is physically situated in another" [140]. Establishing

a virtual environment is not only limited to the visual aspect (i.e. 3D navigable environment) but should also include sound and haptic feedback. This work does not use sound or haptics.

1.2.3.1 Visual Aspect

Typically, a virtual reality system tracks the position of the user and the direction of their gaze. This information is then used to update the display. The system will then produce two images from the 3D scene, one from the perspective of each eye.

The visual aspect of VR has been approached in different ways. One approach is the Binocular Omni-Orientation Monitor (BOOM) developed by Fakespace Labs [70]. BOOM is a binocular monitor attached to a mechanical arm with six degrees of freedom. The position of the arm in 3D space is used to obtain head tracking information that is fed back into the system to update the 3D scene. A more popular approach is the head-mounted display (HMD) which is a binocular display worn on the head. HMDs have internal and external sensors that feed head tracking information into the controlling computer system. Cave Automatic Virtual Environment (CAVE) [21] is another take on the visual aspect of VR where the 3D scene to be displayed to the user is stereoscopically projected onto the walls of the room. The user wears a pair of 3D glasses that shows each eye the scene from a perspective appropriate to that eye. Head tracking information is obtained by tracking a set of tracking targets that are attached to the 3D glasses.

1.2.4 3D Imaging in Medicine

Projectional X-ray was perhaps the earliest imaging modality used in medicine, but due to its nature as a projectional modality, it suffered from many problems such as superposition. Computed Tomography (CT) offered an answer to this problem as it can produce multiple 2D images that represent slices in the 3D object being imaged. Several methods for visualizing these slices emerged; the most obvious method for visualization is viewing individual 2D slices as a group of thumbnails printed on a film or interactively through a computer software by scrolling through these slices.

Methods of visualization where information from multiple slices is represented in a single view have been devised. Minimum and maximum intensity projections are perhaps one of the simplest of these methods. In these techniques voxels along a particular axis are collapsed into one pixel whose value is either the minimum intensity or the maximum intensity encountered along that axis respectively. Despite suffering from the problem of superposition because of their projectional nature; these techniques have found use in many areas of medicine [89] [7] [99] [116] [128] [97] [96] [90] [64] [80].

More advanced 3D visualization techniques have been developed and can be classified into direct display and indirect display. Indirect display is often termed shaded surface display (SSD) or iso-surfacing, and it was introduced in the 1970s [93] [103] [40] [19] [42]. In this technique, a surface of iso-intensity is first extracted from the 3D volume (this step can be preceded by segmentation) then a 2D image is generated that represents the light reflected off that surface from arbitrarily positioned

light source(s) [131]. Additional depth cues such as shadows and motion parallax can be utilized to augment the 3D experience. Iso-surface extraction algorithms include marching cubes [76] [92], marching tetrahedrons [24], dual contouring [61] and others. Many modern-day implementations of SSD use marching cubes algorithm or one of its extensions to build a 3D mesh that is then rendered using one of the modern rendering pipelines (e.g., OpenGL) which handle hidden surface removal and motion parallax. Examples of SSD use in medicine include visualization of scintigraphic data [135], femur fractures [25], brain surface [130] [67], 3D ultrasound images [136], cerebral aneurysms [39] and surgical planning [28]. Kurenov et al. used surface rendering techniques to generate 3D meshes of pulmonary vessels to be 3D printed to facilitate anatomical study [68]. An important variant of SSD is "Virtual Endoscopy"; it is a technique where the internal surface of a hollow organ is rendered, and the user is allowed to move around the organ's lumen simulating an endoscopic examination [108] [44] [47] [133]. An overview of the early applications of 3D display to the study of the heart and lung is given by Hoffman [49].

Direct display is also known as volume rendering. In this type of visualization, the volume is considered a light emitting, semi-transparent medium. Transfer functions map intensity values in the input volume to medium properties such as opacity and color. There are several methods to perform volume rendering. A popular method is "Raycasting." In its simplest form, raycasting is a technique where multiple parallel rays (one for each pixel in the 2D output) are cast through the volume. Each voxel along a ray contributes to the final color of the pixel that corresponds to that ray.

The transfer function determines the material properties of any given voxel along a ray and its contribution to the final color. If all voxels contribute equally to the final pixel intensity value, the result is a projection image similar to an X-ray image [12] [127]. Raycasting is computationally expensive but can be parallelized and thus can benefit from modern GPU optimizations [65]. Other methods for volume rendering include texture-based volume rendering [46], splatting [139] and shear warp [71].

Initially, volume rendering was used for the visualization of skeletal anatomy [45], its use has grown greatly since. Clinically, it was used for many purposes including craniofacial surgery planning [81] [132] [75], orthopedic surgery planning [33] [35] [34] [94] and angiography [69] [120] [66]. Many clinical and research software packages support volume rendering [63] [117] [43] [109] [105] [74] [104].

1.2.5 Virtual Reality in Healthcare and Medical Research

Richard Satava [114] classified the use of VR in healthcare into four categories which are (1) Pre-, intra- and post-operative assistance in surgeries, (2) Medical education and training, (3) Medical database visualization and (4) Rehabilitation medicine. Judi Moline [87] added medical therapy and preventive medicine and patient education, among others. This work falls under the medical database visualization category. Before reviewing the work done in the domain of medical database visualization, we will briefly review notable VR applications in some of the other categories.

In the area of medical therapy, most VR applications address mental health.

VR has been tried as a treatment for Acrophobia [107] [29], Arachnophobia [13] [38], Clustrophobia [9], PTSD [106] [23] [6], eating disorders [100] [101], phantom limb [88], regional pain [115], burn pain [50], ADHD, autism [123] and Amblyopia (Lazy eye) [26] [8].

In 2005, Rizzo et al. published an excellent article that presented a "SWOT Analysis of the Field of Virtual Reality Rehabilitation and Therapy." VR technology has improved a lot since its publication, and some of the weaknesses cited are less pronounced with today's technologies. However, most of the points made in the paper are still valid today [102]. Many works examined the rehabilitative use of virtual reality. Grealy et al. reported improved cognitive functions in patients with Traumatic Brain Injury (TBI) who underwent virtual reality-based rehabilitation [41]. Christiansen et al. [15] and Zhang et al. [141] used virtual reality to evaluate daily living skills in patients with TBI. Laver et al. [73] have recently reviewed and analyzed 19 earlier clinical trials that examined the utility of VR in stroke rehabilitation. They found that most of the studies focused on upper limb training; they reported a limited but statistically significant effect on arm function compared to alternative approaches; they also noted that these trials had low recruitment rates.

Medical training is one of the areas where virtual reality lends itself. An example use case that was studied numerous times in the literature is surgeon training. Seymour and colleagues have reported improved operating room (OR) performance for surgical residents trained with VR on a specific task when compared to controls who weren't pre-trained [119]. Ahlberg et al. [4] and Larsen et al. [72] reported similar

improvement in trainee performance when trained using VR. Rahm et al. reported a significant improvement for novice trainees on knee arthroscopy when trained using VR [95]. Gallagher et al. published an article where they critically reviewed VR applications in surgery training and concluded that VR is an effective tool for training but only in the context of a well-thought-out surgical training program [37]. For a detailed account on the use of VR in medical training, the reader is directed to Ruthenbeck and Reynolds's paper entitled "Virtual reality for medical training: the state-of-the-art" [110].

There has been less focus on the area of medical data visualization using virtual reality. McDonald et al. created a system that allowed anesthesiology residents being trained on epidural technique to view and interact with 3D reconstructed datasets obtained from patients [83]. Satava reports on the work of J. Henderson where he built a 3D visualization tool for Vietnam war injuries database [113].

More recently, Farhani et al. examined the use of virtual reality for the visualization and diagnosis of pathology slides; they found that pathologists were equally accurate with VR and the conventional method [31]. Usher et al. built a virtual reality tool for neuron tracing that was shown to be superior to conventional utilities [129]. Eeger et al. integrated OpenVR [1] (A common application programming interface "API" for the most popular VR HMDs) into MeVisLab medical visualization platform [27].

There is a plethora of works that investigated the utility of virtual reality in a wide range of areas of biomedical research; many of these works pointed out

that a carefully designed virtual reality system can be a valuable asset, even though, at the time, virtual reality systems' performance was quite limited. In recent days, computing power has caught up to the demands of virtual reality systems to the extent that we have affordable, low-latency and high-performance virtual reality systems in the consumer market. This poses an opportunity for further exploration of this technology for biomedical research applications.

CHAPTER 2
TECHNOLOGY

2.1 Virtual Reality Hardware

2.1.1 Head-mounted Displays (HMDs)

As of this writing, there are three major competitors in the virtual reality headset market: "HTC Vive" [16], "Facebook Oculus Rift" [55] and "Sony Playstation VR" [18]. Only the former two are designed to work with Personal Computers (PCs). In terms of screen resolution, both the HTC Vive and the Oculus Rift have 2160 x 1200 displays and a 90 Hertz refresh rate. The major difference between the two systems is that the HTC Vive is designed to be used in room-scale virtual environments that allows the user to move around the environment. The Oculus Rift, on the other hand, operates in a smaller area. Also, the price of the Oculus Rift is lower than that of the HTC Vive. More recently, HTC announced the "VivePro" which offers tetherless HMD, higher resolution, and more sensors [17].

2.1.2 Controllers

Both the Oculus Rift and the HTC Vive have support for hand controllers. The HTC Vive's controllers are larger than those of the Oculus Rift's (see figure 2.2). The Oculus Rift's controllers (Oculus touch) are two wireless controllers that are roughly the size of an Apple, each with five buttons and two analog sticks. The controllers feed information about hand orientation, hand position in space and the state of each of the buttons to the application. The controllers also have programmable motors

that provide haptic feedback in the form of rumbling (see figure 2.1).

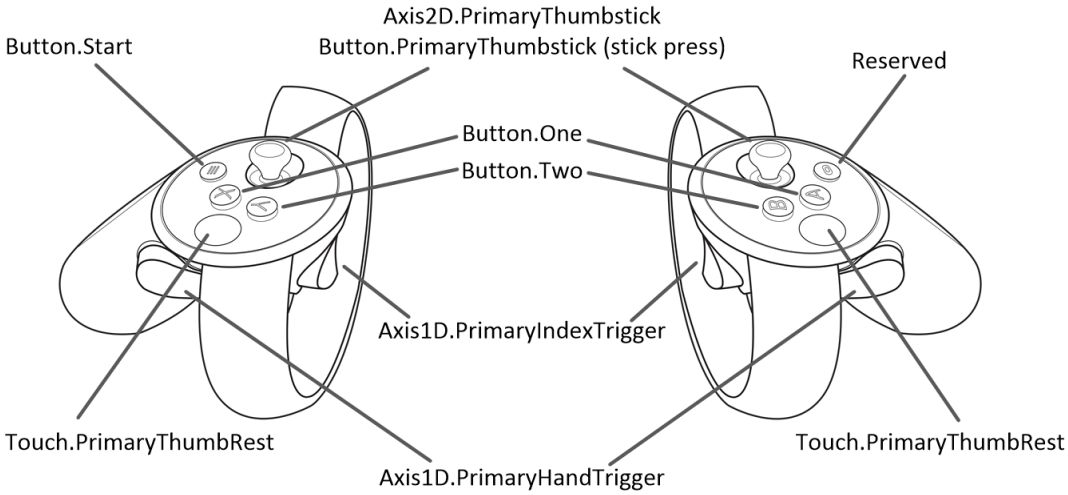


Figure 2.1: Oculus Rift's hand controllers
(Source: Oculus Rift Developer Documentation)

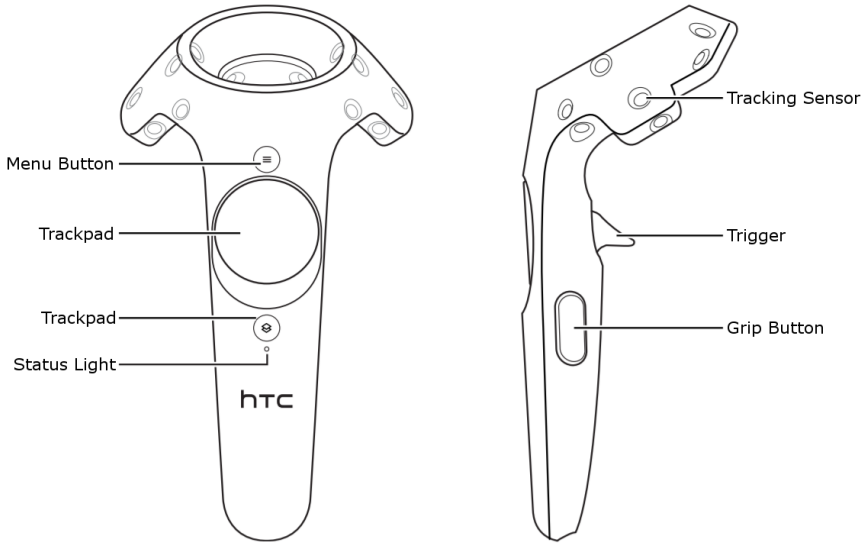


Figure 2.2: HTC Vive's hand controllers
(Source: HTC Vive PRE User Guide)

2.2 Rendering for Virtual Reality HMDs

End-to-end input latency is defined as the time interval between a user's physical motion and the corresponding on-screen visual feedback [142] [14]. Modern-day interactive desktop applications usually have a mean end-to-end input latency hovering around 50-60ms [14]; this latency can reach 100ms for games that run at 30 frames per second [122]. However, for virtual reality applications, the recommended latency is around 20-25ms or less [77]; greater latency can break immersion or cause motion sickness. End-to-end latency in virtual reality context is often termed "Motion-to-photon" latency. Sources of latency in virtual reality are illustrated in figure 2.3. Achieving a latency of 20ms requires optimizations on both hardware and software levels. Hardware optimizations are numerous, and they include using low latency sensors and low persistence screens. Some of the software optimizations are performed by the VR runtime (such as Time Warping which we will explain in a later section), but to reach the 20-25ms latency, the application developer has to maintain a consistent frame rate of 90 frames per second which translates into a render time budget of roughly 10ms per frame.

Another aspect where VR and desktop rendering differ is that a VR application is required to render any given scene twice, once from the perspective of the left eye and once from the perspective of the right eye; this effectively doubles the rendering workload. Additionally, VR applications render scenes with a wide field of view (FoV) [77]. VR Applications don't typically render directly to the screen but rather to an in-memory buffer; they submit the buffered frames to the VR runtime

which performs post-processing such as correcting for the distortion and chromatic aberrations introduced by the lens. The corrected frames are then displayed on HMD [57].

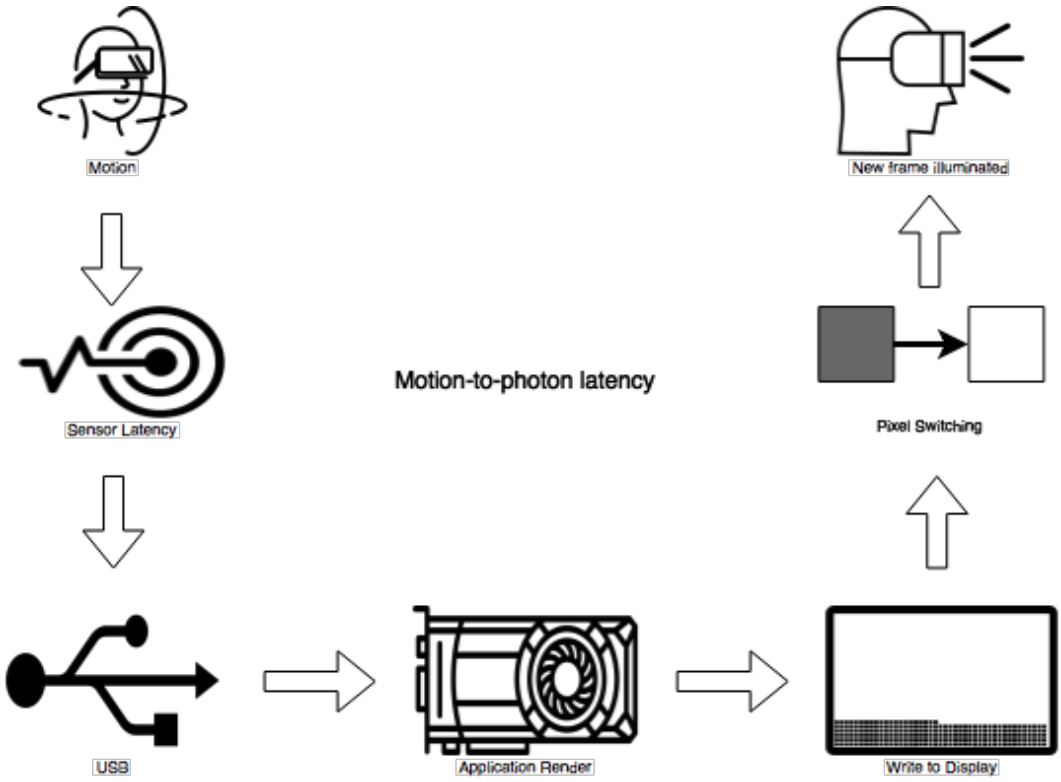


Figure 2.3: Motion-to-photon latency

Icons sourced from thenounproject.com

2.2.1 Rendering Timeline

A frame starts with a call to the VR runtime library to obtain sensor information, namely head, and hands' poses. The application uses sensor information to render two frames, one for each eye. This introduces some latency in the system

because, by the time the frames are submitted to the VR runtime, the head pose is already several milliseconds old. This latency may not be noticeable on a computer screen, but in VR settings it can be noticed. Afterward, the application submits these frames to the VR runtime which processes these frames and sends the final images to the HMD.

2.2.2 Reducing Latency

Long motion-to-photon latency can degrade the user experience significantly or even cause motion sickness. For ideal experience, latency should be maintained around 20-25ms. In this section, we overview some of the software techniques that can be used to reduce latency.

2.2.2.1 Time Warping

Time warping is a technique employed by the Oculus Rift's runtime to reduce latency; It doesn't require developer intervention. In this technique, the VR runtime reads the sensors just before sending the final frames to the HMD for rendering and applies a transformation to the final frames that reflects the new sensor information. This transformation can reduce the perceived latency (see figure 2.4). Time warping can correct for head rotation, but not translation as translation will introduce disocclusion artifacts.

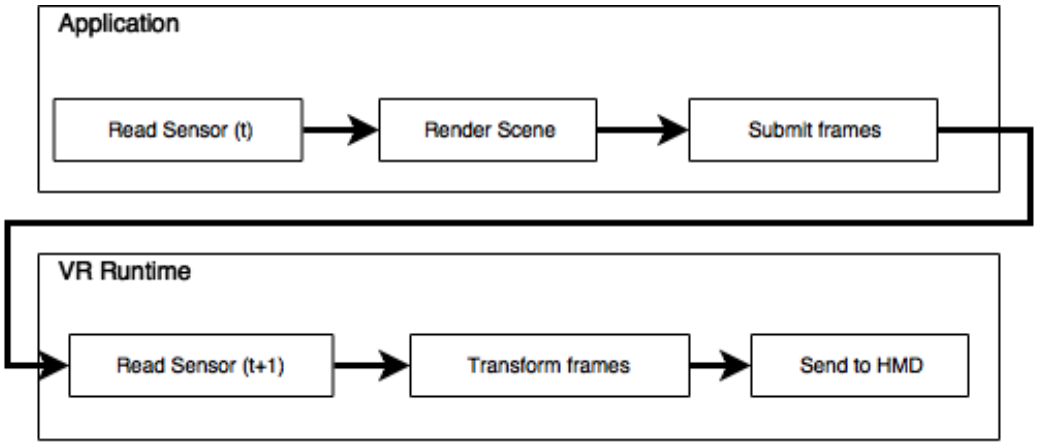


Figure 2.4: Asynchronus Time Wrap

2.2.2.2 Space Warping

Space warping is a feature that was recently added to Oculus Rift; it works in tandem with "Time Warping." In this technique, the VR runtime synthesizes missing frames by extrapolating from the previous two frames and both the user's head orientation and position instead of just warping the latest frame. This technique enables VR applications to run at a frame rate of 45 fps with minimal degradation in the experience thus allowing less powerful GPUs to run VR applications [54].

2.2.2.3 Single Pass Stereo Rendering

Rendering the scene two times once for each eye is mandatory for VR applications. This essentially doubles the work on both CPU and GPU sides. On the CPU side, the scene graph will be traversed twice, and the CPU will issue twice as many render calls and state changes such as attaching textures, enabling/disabling blending, depth testing, etc. On the GPU side, the number of commands to execute

is doubled, and the geometry will be rendered twice. This can be rather expensive, especially when the rendering budget is as low as 10ms. Luckily, there are means to reduce this overhead. On the CPU side, OpenGL instancing can be used. OpenGL instancing allows the application to ask the GPU to render a particular piece of geometry any given number of times with one call. For VR purposes, instancing allows the geometry to be rendered twice by the GPU with one set of render calls and state changes; this reduces the workload on the CPU side. However, the GPU is still doing twice the amount of work. Reducing the workload on the GPU side cannot be done using software alone. One approach to reducing the GPU workload is the *NV_stereo_view_rendering* OpenGL extension [125] that was proposed by Nvidia; it reduces the amount of repeated work done on the GPU side. The GPU will still render the geometry twice but more efficiently.

2.2.2.4 Multi-resolution Rendering

When rendering for virtual reality HMDs, rendering different parts of the scene at different resolutions can improve performance. The human visual system has lower resolution toward the periphery [36]. Coupling this with the fact that peripheral parts of the scene are represented with fewer pixels than the central parts in the final image due to the barrel distortion, it becomes clear that it is possible to render the peripheral parts of the scene at a lower resolution to improve performance with little to no degradation in the experience. The *NV_clip_space_w_scaling* OpenGL extension has been proposed to achieve that at OpenGL level [126].

The Oculus Rift's runtime also allows rendering different layers of the scene at different resolutions. For example, the developer can render non-textual elements at a lower resolution while rendering textual elements at a higher resolution to achieve higher frame rate without sacrificing legibility [56].

2.2.2.5 Efficient Transparency

In scientific visualization, transparency can be a handy device for visualizing complex objects. In 3D graphics, transparency is often simulated by using "Alpha blending" where each new fragment submitted to the rendering pipeline has a color and an alpha value. Alpha values reflect that fragment's "opaqueness" with a value of 1.0 meaning a fully opaque fragment. When a new fragment is sent to the pipeline, the color of the fragment is combined with the existent color in the frame buffer at that location using the following equation.

$$C_{out} = A_{src}C_{src} + (1 - A_{src})C_{dst}$$

Where C_{out} is the color to be stored in the frame buffer at a given location, A_{src} is the alpha value of the new fragment, C_{src} is the color of the new fragment and C_{dst} is the color value already stored in the frame buffer. This approach works well for many purposes but requires that fragments be sent to the pipeline in-order, with fragments closer to the camera coming last; this can be achieved for simple transparent objects such as windows or sheets of glass. However, it can be very hard to achieve if a single transparent object overlaps itself (as it will require reordering the triangles that make up the object itself) or even impossible if transparent objects

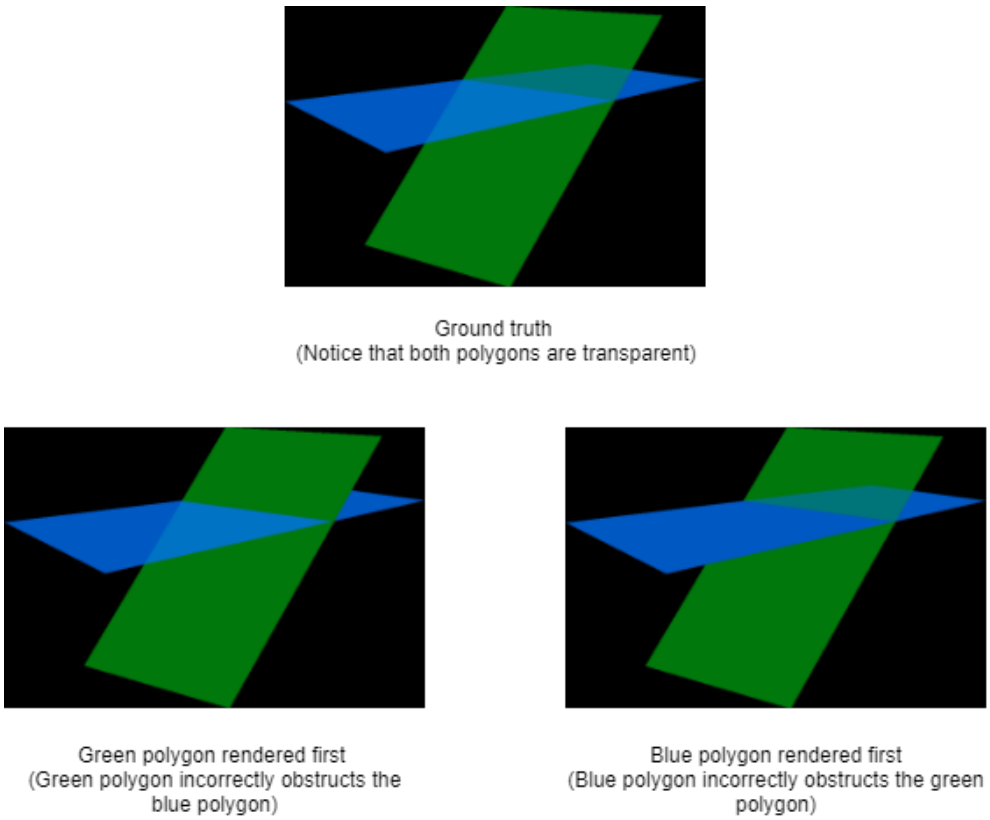


Figure 2.5: Order-dependent transparency impossible case

images derived from a question on StackOverflow.com [91]

intersect (see figure 2.5) which is commonplace in scientific visualization. Several order-independent transparency (OIT) algorithms have been introduced to address this problem.

Depth peeling is an OIT technique that was introduced by Cass Everitt [30] in 2001. In depth peeling, multiple rendering passes are done. For the first pass, only the pixels that are closest to the camera are rendered, in the next pass only pixels that are the next closest to the camera are rendered and so on, with each pass "peeling" away a depth layer; these layers are then combined and rendered. Depth peeling can

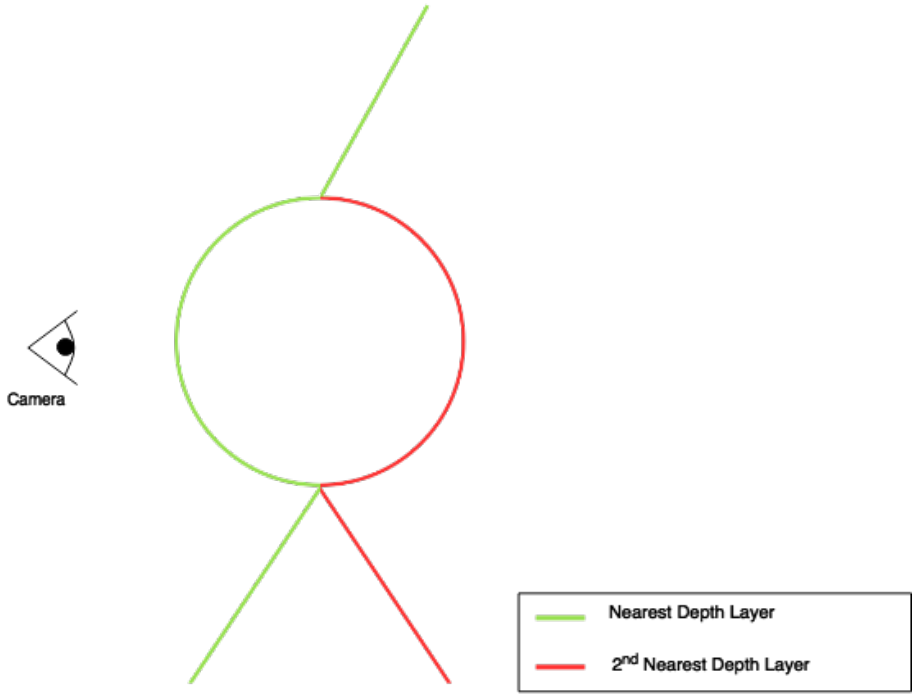


Figure 2.6: A simple 2D example on depth peeling.

Pixels on the "green" surface are not the same distance away from the camera but they are all the foremost pixels from the point view of the camera and thus constitute a "depth layer"

achieve pixel-perfect transparency but requires multiple passes especially for complex scenes that have multiple depth layers; this can be impractical for real-time rendering.

More recently, Mcguire introduced "Weighted, Blended Order-Independent Transparency" [84]. In this technique, the contribution of a transparent surface to the final color of a pixel is weighted by that surface's alpha value, and its distance from the camera; farther away surfaces will contribute less than near surfaces. In other words, the final color of a pixel that overlies one or more transparent surfaces will be a weighted sum of all underlying pixels on these transparent surfaces. The advantage of this approach is that it can be performed in one pass on modern hardware using

multiple render targets; this makes it ideal for real-time rendering, but it requires tweaking the weighting function to fit the range of depth values in the scene.

CHAPTER 3
APPLICATION DESIGN

3.1 Application Design

3.1.1 External Libraries

The application was built using C++, and several libraries. Table 3.1 shows these libraries, their versions, and their function. Figure 3.1 outlines the interaction between the application and these libraries.

Library	Version	Use
VTK [117]	8.1	Generating surface mesh from 3D mask volumes, smoothing and decimation, VTK wasn't used for rendering.
ITK [60]	4.13	Reading image data from CT scans, the data was used to generate 2D and 3D OpenGL textures[1]
FreeType [22]	2.8	Generating bitmaps from font files for typography
GLEW [86]	2.1.0	Dynamically loading modern versions of OpenGL
GLFW [78]	3.2.1	Cross-platform window creation and window life cycle management
Oculus SDK [56]	1.14	Interfacing with the Oculus Rift hardware

Table 3.1: Libraries used in the application
(1) ITK function "TransformPhysicalPointToIndex()" that transforms between physical coordinate system and the image voxel index coordinate system was too slow for real-time rendering and was replaced by less flexible but much faster implementation.

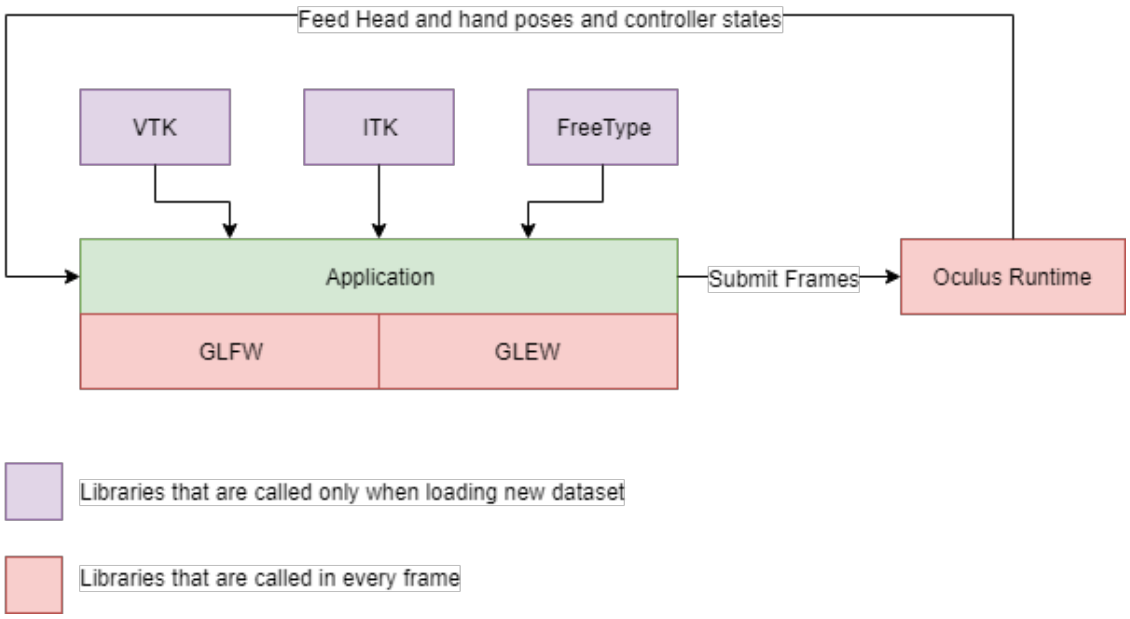


Figure 3.1: Interactions between libraries and application

3.1.2 Object Organization

3D objects in the application are organized in a scene graph. A scene graph is a tree that represents logical relationships (often spatial relationships) between objects in a 3D scene. For an example of spatial relationships in a scene graph, think of a person driving a car. In a scene graph, the person node will be a "child" of the car node which means that transformations applied to the car like translation or rotation will also be applied to the "person" node causing the driver to move with the car. In a scene graph, the developer doesn't have to explicitly transform each child individually; the rendering engine will "combine" parents' transforms with child node transform.

In this application, a simple scene graph was used, where each node can be

"queried" to return a set of "render commands" that define how it should be rendered. Each command defines a shader program and a render state (e.g., whether or not blending is enabled, whether or not depth testing is enabled, etc.). A "scene renderer" will then traverse the scene graph, querying all visible nodes, sorting the render commands by shader program and render state and caching these commands for quick access. Changes to the scene graph (e.g., hiding a node) would invalidate the cache, and it will be rebuilt in the next frame. A portion of the scene graph's implementation is based on the C++ library "Cinder" [2].

3.1.3 Anatomy of a Frame

A frame starts with a query to the VR runtime to get the latest head and hand poses as well as controller states. The application then uses this information to update the application state and the scene graph. For example, when the user wants to rotate the airway, the airway's position in the scene graph is changed to become a child of the "right hand" node. All "right hand" transformations will now be applied to the airway giving the users the illusion that they are holding the airway tree in their hand. After that, the application goes through the cached list of render commands (updating the cache if needed), rendering opaque samples first, followed by transparent samples. Afterward comes the step of post-processing, in this work, post-processing is only needed for weighted blended order-independent transparency. Figure 3.2 outlines the flowchart of a single frame in the application.

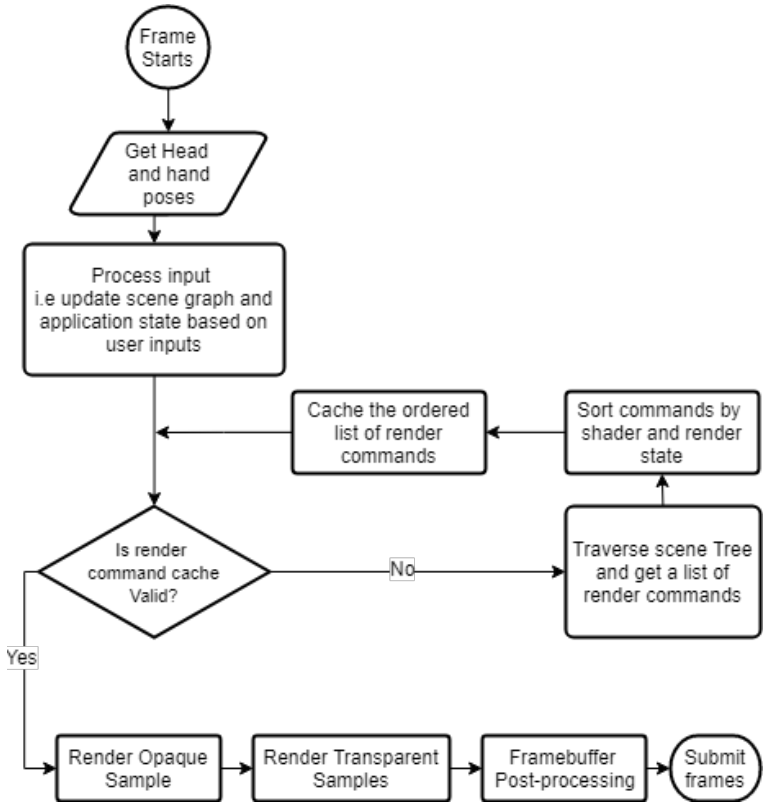


Figure 3.2: Single frame flowchart

3.1.4 Shaders

Shaders are developer-defined programs that run on the GPU at different points in the rendering pipeline. OpenGL shaders are written in a C-style language called "OpenGL Shading Language (GLSL)." GLSL shaders are an integral part of modern OpenGL-based applications. GLSL shaders are compiled and linked at runtime using OpenGL function. In this application, support for C-style "#include" macro was added to allow for the concatenation of multiple shader files which helps in code organization and reuse. Change detection is also implemented; this allows the application to detect changes to shader files in real-time which triggers compilation

and linkage of all the shaders that depend on the modified files. Change detection allows for real-time tweaking/debugging of shaders without having to recompile and restart the entire application.

3.1.5 The 3D Mesh

For each subject, two surface meshes are generated: a mesh for the airway tree produced from the airway mask and a mesh for the vascular tree produced from the blood vessel mask. The meshes are generated using VTK's implementation of marching cubes [76]. The raw mesh for the airway tree is composed of roughly 750k faces. The raw mesh for the vascular tree is composed of about 10 million faces. VTK decimation filter was used to reduce the number of faces. Decimation is a process where a mesh is transformed into another with fewer faces. The mesh decimation algorithm used by VTK is a modification to the algorithm described by Schroeder et al. [118]. The number of faces for both meshes was reduced to around 85% of the original number with no noticeable loss of details. Meshes were also smoothed by iteratively moving vertices according to the average position of the surrounding vertices using vtkSmoothPolyData Filter [79] (See figure 3.3).

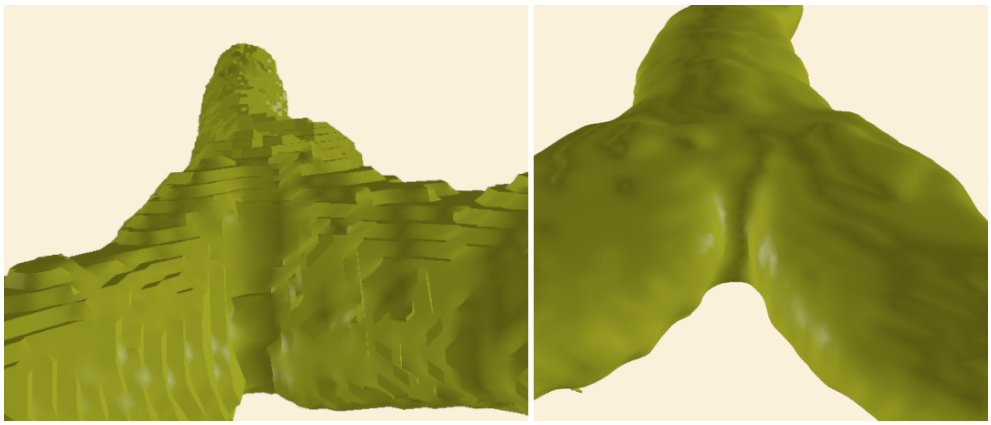


Figure 3.3: Mesh smoothing
Airway mesh without smoothing (left) and with smoothing (right).

3.1.6 Shading and Rendering

For shading the vascular and airway meshes, Phong shading model [93] was used. It was applied on per-vertex basis; this results in lower quality lighting but is faster than the higher quality per-fragment shading. The application reduces the CPU workload by using OpenGL instancing for stereo rendering. For transparency, weighted blended order independent transparency was used [84] (See figure 3.4).

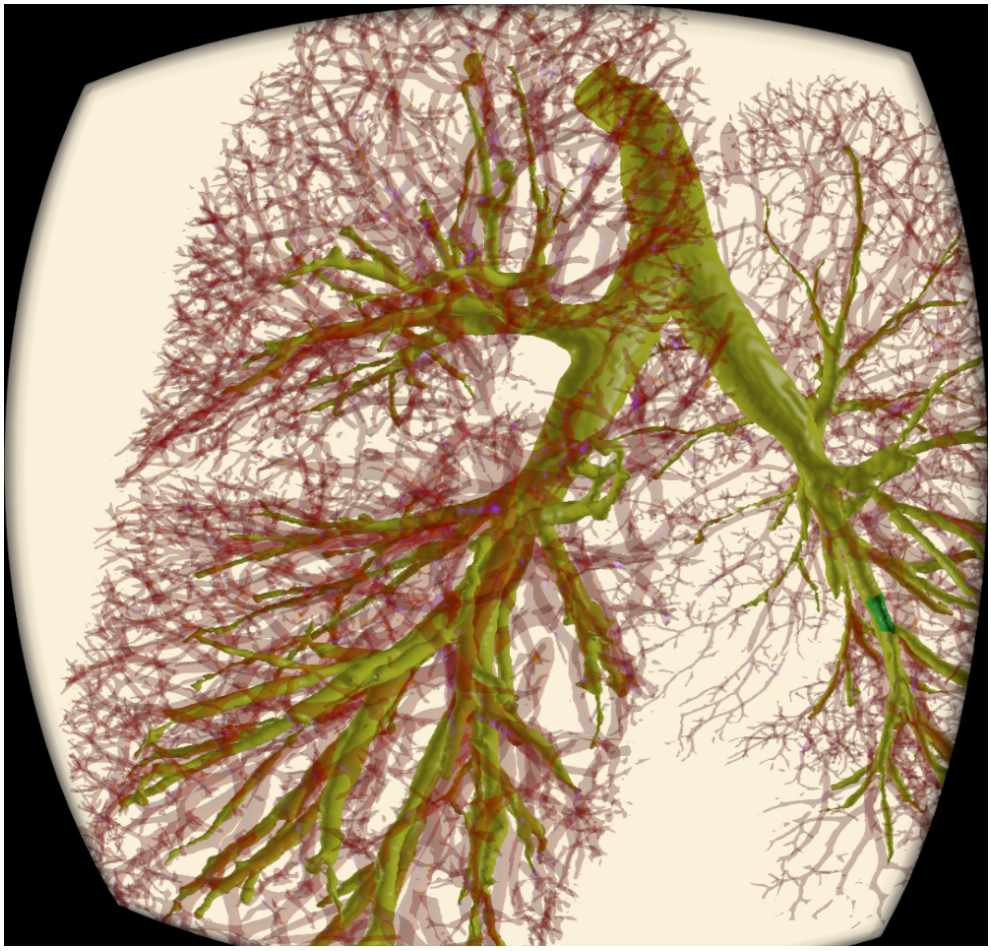


Figure 3.4: Weighted blended order-independent transparency
Triangles that make up the vascular tree weren't ordered before rendering

3.1.7 Text Rendering

To the untrained eye, text rendering and layout can seem simple, but it is a fairly complex system of problems. Luckily, this application has simple needs that can be addressed with simple solutions. To render latin text to the screen in a very simple use scenario, glyph information needs to be loaded from a font file. A bitmap is then generated for each character. Then a texture is generated from these bitmaps,

and finally, the texture for each character is rendered to the screen in the order that character appears in the text.

To reduce the number of draw calls, a bitmap is generated for the whole text instead of creating a bitmap for each character; this translates to one texture per text field which reduces the number of draw calls needed. The bitmap is generated on a different thread and the texture is only updated when the bitmap is loaded. This optimization causes text updates to lag behind a few milliseconds but improves the performance of the rendering thread because it doesn't spend time generating text bitmaps (See figure 3.5).

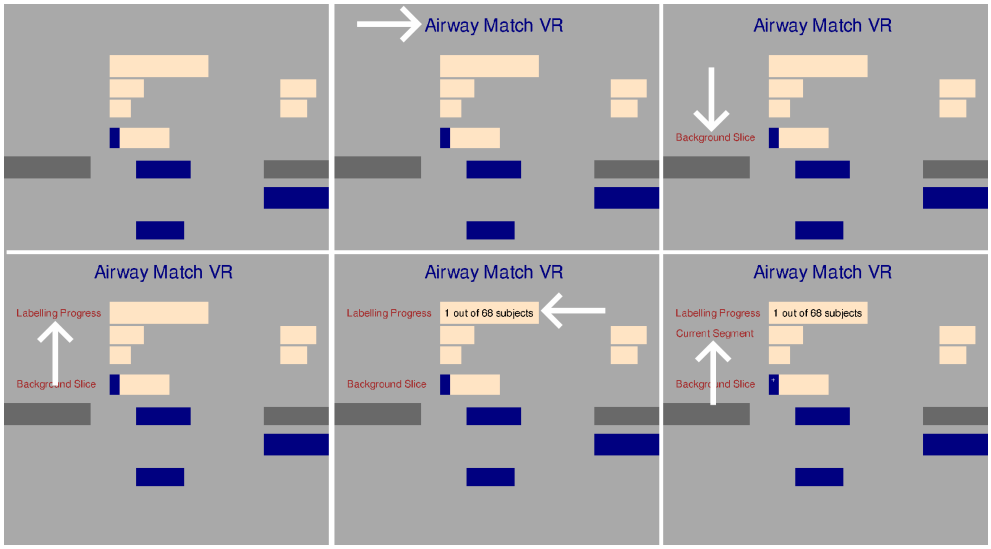


Figure 3.5: Text rendering
Arrows show text fields being added, each text field requires a single render call as characters are not added one by one.

3.1.8 Slice Rendering

The application allows the user to pick a square slice with an arbitrary orientation in 3D, move it and change its orientation as needed. The application shows the user the slice from the underlying CT volume that corresponds to the slice the user controls. The orientation and position of the slice are determined by user interaction through the hand controller. The hand controllers' state is updated each frame which means a slice from the volume will have to be extracted for every frame (i.e. 90 slices per second). If this work is performed on the CPU, the frame rate will suffer because of the overhead of slice extraction from the volume and the latency incurred by moving the slice data from the system memory to the GPU's video memory (VRAM). This problem is solved by loading the entire volume into the VRAM as a 3D texture and sampling from that texture based on the orientation and position of the user-controlled slice. This approach has the added value of OpenGL performing smoothing on the extracted slice efficiently. Sending the entire 3D volume to the GPU is slow and has to be done on the rendering thread, so to avoid slowing down the rendering thread, the texture transfer is amortized over several frames to keep each frame's render time within the allotted 10ms budget.

CHAPTER 4
METHODOLOGY

4.1 Overview

For this work, an Oculus Rift consumer version 1 (CV1) virtual reality HMD was used because it is more usable in a seated position in a confined workspace as well as being more cost-effective than other alternatives. The HMD was connected to a Microsoft Windows machine with Nvidia GTX 970 GPU. A total of 100 baseline scans from SPIROMICS subjects (from Iowa Radiology and Columbia sites) were used. A virtual reality application based on OpenGL was developed. The virtual reality software presented the user with a 3D view of the airway and vascular trees to aid the airway-to-artery matching. Using two hand-held controllers (see figure 2.1), the user is able to navigate, rotate, pan and crop the 3D rendering as well as view arbitrarily-oriented slices from the underlying volume. The user's goal for each subject is to pick two 3D planes that demarcate the beginning and end of the artery that matches a particular airway segment.

For evaluation, the user's performance in matching airway segments (specifically LB10 and RB10) with their corresponding arteries on the proposed virtual reality system and a 2D system (Pulmonary Analysis Software Suite (PASS))[43] was compared. The user was shown segments from 100 subjects and was required to mark the beginning and the end of the corresponding artery. Each airway segment was shown to the user twice in a double-blind, random order yielding a total of 200

observation per system. The performance of a single user was reported in this work (N=1). The user's performance on both systems was examined.

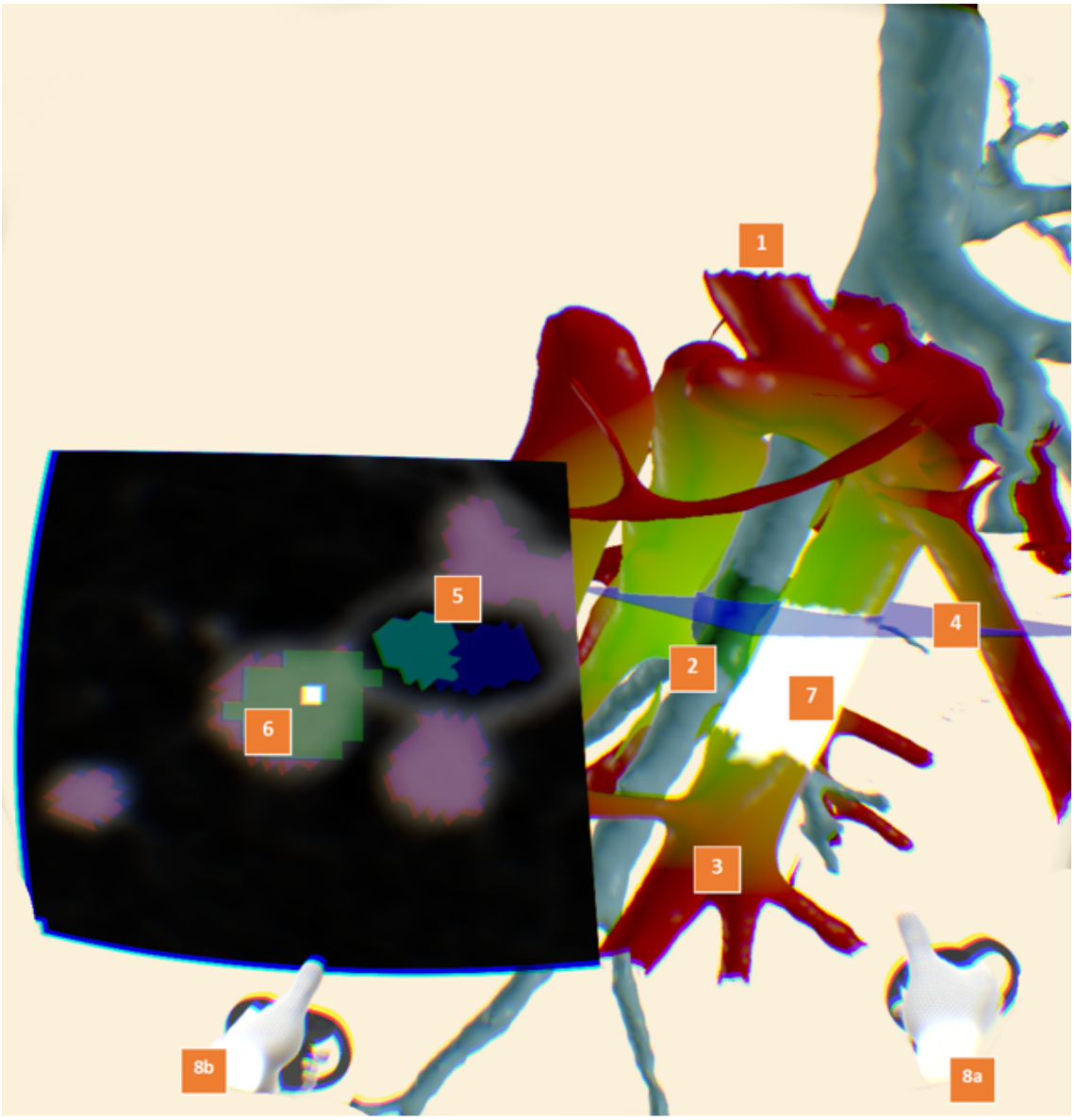
4.2 Dataset

A total of 100 baseline full inspiratory scans from SPIROMICS [121] subjects (from Iowa Radiology and Columbia sites) were used. The subjects represented GOLD COPD stages 0, 1 and 2. Males and females were equally represented in the dataset. Only LB10 and RB10 airway segments were considered. Each scan had an airway mask and a blood vessel mask, both were produced by VIDA diagnostics. These masks were used to create a 3D surface rendering of the airway and the blood vessels using VTK's marching cubes [76] implementation.

4.3 User Interaction

User interactions were expected to be performed while seated. Aside from looking around, all user interactions are performed using the hand controllers. The interactions include: (1) Moving around the 3D mesh, (2) Moving the 3D mesh, (3) Rotating the 3D mesh around an arbitrary point, (4) Viewing intensity image of an arbitrary slice as well as rotating, moving and scaling that slice, (5) Viewing Minimum, Maximum and average intensity projections overlayed on the 3D mesh, (6) Cropping the blood vessels from the scene to a sphere of arbitrary diameter around the targeted airway segment (see 4.2) and (7) Picking and orientating a 3D plane to mark the beginning/end of a blood vessel segment (the application provides the user with a preview of the slice represented by that plane in the volume). Figure 4.1 shows

the user interface.



(1) The vascular tree is cropped so that only the blood vessels within a certain radius are shown to reduce clutter. **(2)** Airway segment of interest is highlighted in green. **(3)** Vascular tree is given colors between green and red based on their proximity to the segment of interest. **(4)** A plane that the user indicated as the vessel segment starting point. **(5)** The CT slice corresponding to the plane in 4, the segment of interest is given a teal color, airway is shown in blue and the blood vessel mask is shown as faint magenta. **(6)** The detected vessel and its centroid. **(7)** The final selected segment that is demarcated by the two user-selected planes (only one shown in 4). **(8a, 8b)** Virtual representation of viewer's hands.

Figure 4.1: User interface

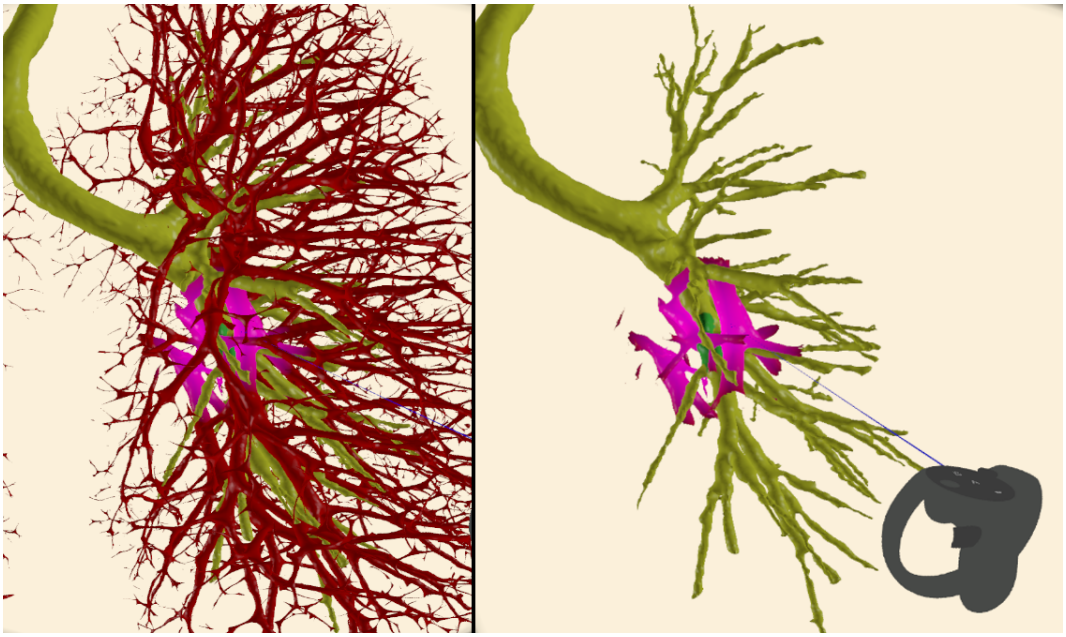


Figure 4.2: Cropping view to remove clutter

4.3.1 Workflow for VR

The user is presented with a 3D rendering of the airway; she/he then navigates the view, rotates, crops and traces blood vessels visually. The application overlays a heatmap on the blood vessel mesh that indicates its proximity to the targeted airway segment to aid in the process of locating the matching blood vessel. Ideally, the user visually traces blood vessel and airway branching patterns to find a match (see figure 4.3 for an example of a match). When the user is ready, she/he picks a point that marks the beginning of an artery, orientates a 3D plane centered around that point, and then confirms the selection of that plane. The user then proceeds to pick the plane that represents the end of the selected blood vessel. The application then highlights the blood vessel segment that the user picked. If the user is satisfied with

the result, the user moves on to the next subject. Figure 4.4 shows the segment selection workflow.

Figure 4.3: Manually cropped view of the airway and the corresponding artery

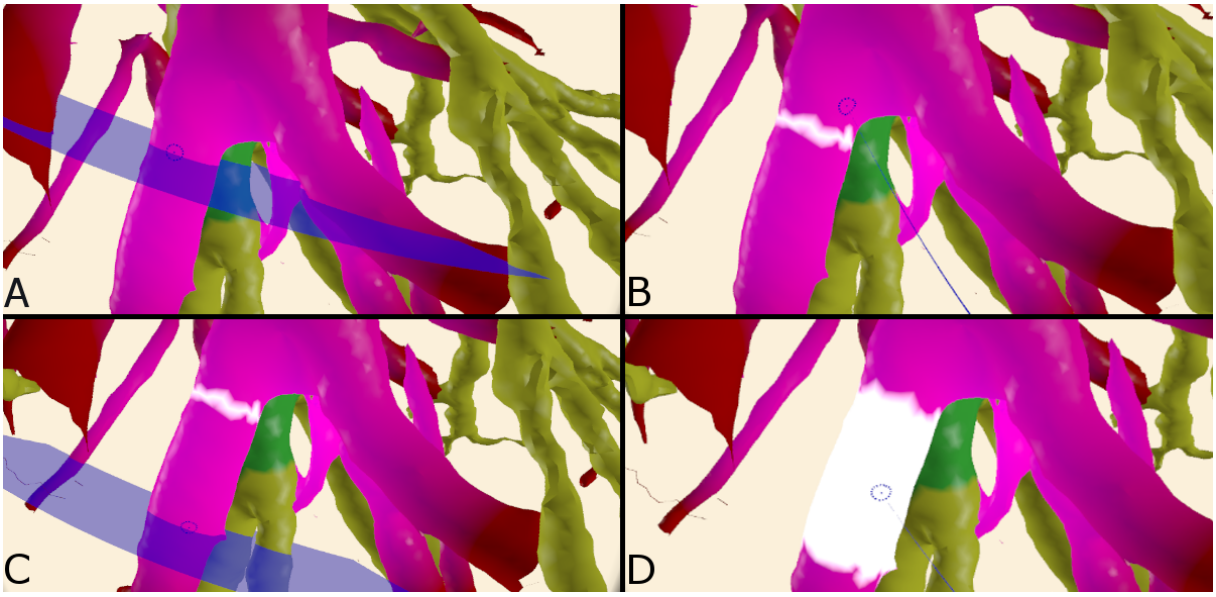


Figure 4.4: Artery segment selection workflow
(A) orientate the first plane, (B) confirm selection, (C) orientate the second plane, (D) confirm selection. The selected segment is marked in white.

4.4 Evaluation Metrics

Two metrics were used to evaluate and compare the user's performance on both systems, namely, the matching time and the matching error.

4.4.1 Matching Time

In both systems, The "matching time" was defined as the number of seconds elapsed between the time when the user was first presented with the 3D mesh or the 2D image to the time she/he picked the second point on the selected blood vessel.

4.4.2 Matching Error

In this work, there was a lack of a ground truth dataset. To evaluate user's performance, the number of "agreements" between two observations of the same airway segment in the same subject on each system was used as a metric.

Two observations are said to be in "agreement" if the two-point pairs fall within the same blood vessel with no branching points between them. In "One generation below" error the two pairs of points fall on the same blood vessel with one intervening branching point. In "two generation below" error, the two pairs of points fall on the same blood vessel with two intervening branching points between them. Supernumerary branching points were not considered as branching points.

If the two pairs of points are on different vessels or on vessels that have common origin above the beginning of the targeted airway segment, these pairs are said to have "Different vessel" error (see figure 4.5). Matching errors were counted manually.

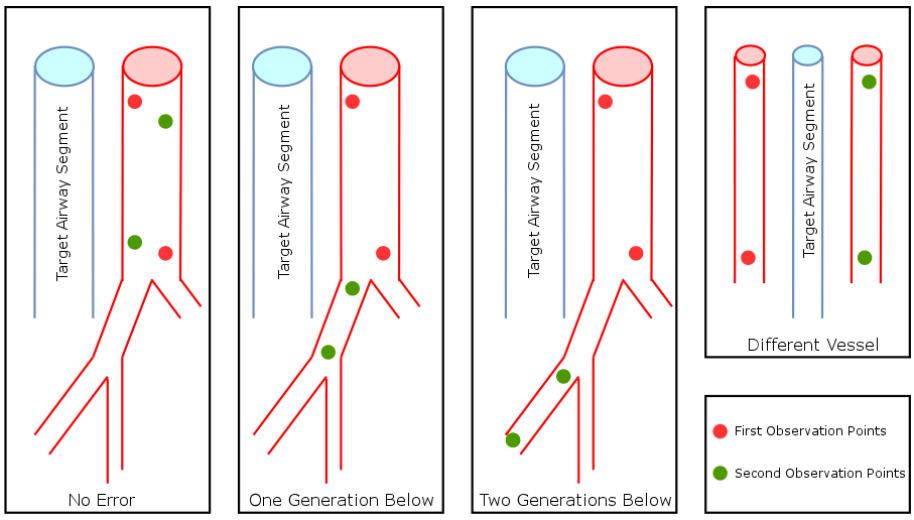


Figure 4.5: Types of errors

CHAPTER 5
RESULTS AND DISCUSSION

5.1 Results

5.1.1 Matching Time

The median matching time for VR was 46 seconds while the median matching time for 2D software was 30 seconds. Examination of the median matching time progression over five sessions in the case of VR revealed a downward trend indicating a "learning-curve" (see figure 5.1). The median matching time moved from 105 seconds (N=30) in the first session to 36.5 seconds (N=100) in the fifth session (p-value $\approx 10^{-20}$). For both systems, there was no statistically significant difference in matching time for different GOLD stages or subject's gender.

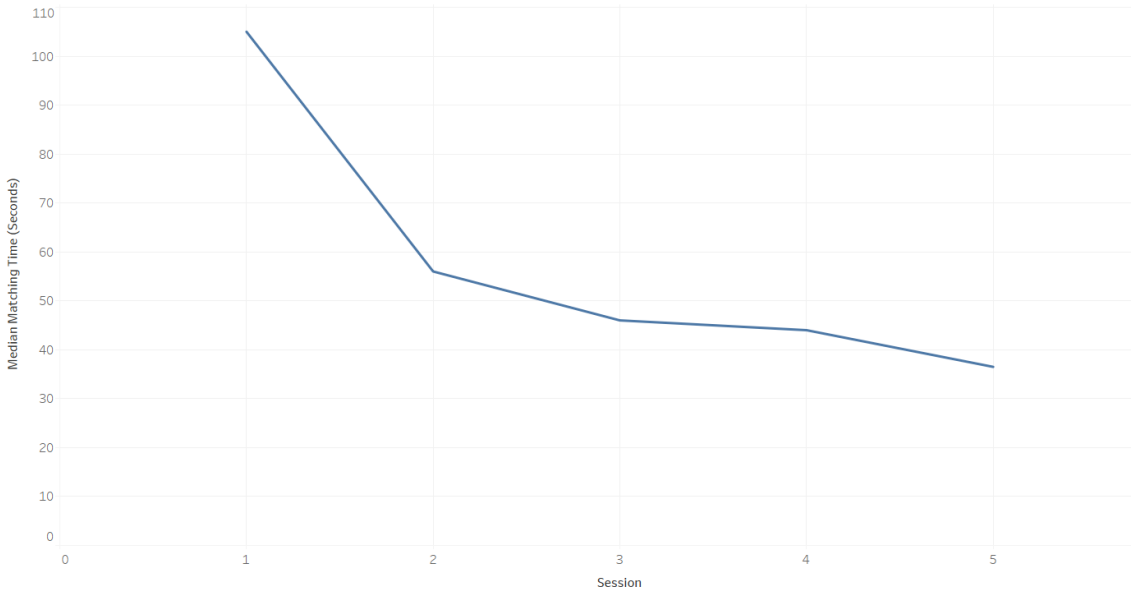


Figure 5.1: Progression of median matching time on VR

5.1.2 Matching Error

Out of the 198 pairs of observations that were done using the 2D system, 17 pairs had disagreements (13 "one generation below" errors, 1 "two generation below" error and 3 "different vessel" errors). Factors such as the side of the airway segment (right and left), GOLD stage and gender had no influence on errors. There was no statistically significant difference in matching times between pairs with agreement and those with disagreement. Out of the planned 200 pairs, two pairs of observations were not recorded due to a software error.

In the 198 pairs of observations that were done using the VR system, 33 pairs had disagreements (28 "one generation below" errors, 1 "two generations below" error and 4 "different vessel" errors). Again, there was no statistically significant difference in the number of errors between segment sides (right and left), GOLD stages and genders. The median matching time for pairs with disagreement was about 10 seconds more than that for pairs with agreement (p-value ≈ 0.004). Pairs with "different vessel" errors had a median matching time of 88 seconds. The selected vessel segment length was examined which is defined as the distance between the user-selected starting point and the ending point of the blood vessel (see figure 5.2). It was found that the mean selected vessel segment length was smaller for vessels with disagreement (9.96 voxels $\pm$4.15 voxels) than the mean for vessels with no error (12.91 voxels $\pm$5.48 voxels) (p-value ≈ 0.003). The mean distance between the selected segments was found to be 17 voxels ($\pm$33 voxels); this distance is how far apart are the selected segments in the first and the second observations (see figure 5.2

for illustration). Out of the planned 200 pairs, two pairs of observations were not recorded due to a software error.

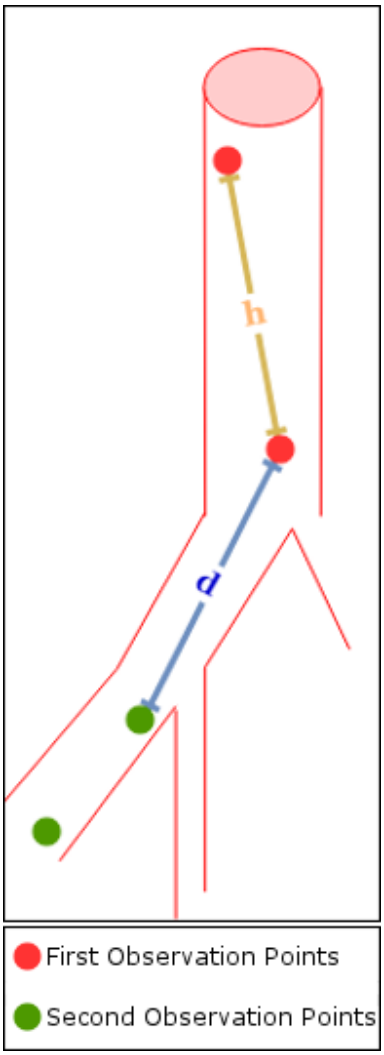


Figure 5.2: Segment length and distance between segments

Vessel segment length (h) is the distance between starting point and ending point for one observation. Distance between selected segments (d) is the distance that separates two selected vessel segments.

There were 181 pairs that had no matching errors when labeled using the 2D

system and 165 pairs that had no matching errors when labeled using the VR system. The intersection between these two groups has 154 pairs that are common to both groups with no matching errors. These pairs were further examined to see if there was an agreement between the two systems as well. There were 128 instances where the four observations (two on the VR system and two on the 2D system) agreed on the selected blood vessel and 26 instances where a "one generation below" error was made. "One generation below" error in this group refers to the case where there was an intra-system agreement (between the two observations on each system), but there was an inter-system disagreement (See table 5.1).

	2D system	VR	Common agreed upon
No Error	181 (91.4%)	165 (83.3%)	128 (83.1%)
One Generation Below	13 (6.6%)	28 (14.2%)	26 (16.9%)
Two Generations Below	1 (0.5%)	1 (0.5 %)	0 (0.0%)
Different Vessel	3 (1.5%)	4 (2%)	0 (0.0%)
Total	198	198	154

Table 5.1: Matching error frequency for the two systems
Note: "Common agreed upon" pairs are the pairs in the intersection between the "No Error" groups of both systems.

5.1.3 Rendering Performance

The application was tested on three different machines. System specifications and performance on these systems are reported in table 5.2.

	System 1	System 2	System 3
Processor	Intel 7700k	AMD Ryzen 5 1600	Intel Xeon E5-2620
Memory	32 Gigabytes	16 Gigabytes	32 Gigabytes
GPU	Nvidia Titan X (Pascal)	Nvidia GTX 1070	Nvidia GTX 970
	Performance		
Average Latency (Idle)	18ms	25ms	23ms
Latency (Viewing a slice[1])	19ms	30ms	31ms
PHR[2] (Idle)	65%	57%	57%
PHR[2] (Viewing a slice[1])	55%	35%	30%
Frame rate	90	90	90

Table 5.2: Specifications and performance of three machines that the VR system was tested on
(1) Performance on viewing a slice (see figure 3.1) is reported because it is the most computationally demanding function in the application. (2) Performance Headroom (PHR) is a metric reported by the Oculus runtime that reflects the unused CPU and GPU rendering capacity.

There were noticeable dips in frame rate that lasted around 2-3 seconds when

the user picks the first point; this is the moment when the 3D volumes begin loading.

5.2 Discussion

5.2.1 Matching Time

With a median matching time of 46 seconds, the user was slower on the VR system. However, there are some points to take in consideration. The user had previously been trained on and used the 2D system for months, but only used the VR system for this work, so there was a learning process for using the VR system but not for the 2D system. This is evidenced by the downward trend in user matching time on VR. Also, when the user picks the starting point of an artery, the VR application starts sending the intensity and mask volumes to the GPU. This process takes, on average, 12 seconds on the machine that was used for evaluation (See figure 5.3). Sending the data to the GPU only when it is needed makes sense in some use cases, but in the context of this application, it contributed a significant increase in the matching time because the user has to wait. Eliminating this bottleneck can potentially reduce the median matching time on the fifth session to about 24 seconds. The 2D system didn't suffer from this problem as the user doesn't have to wait to confirm the points she/he selected.

The user's matching time was improving with time (See figure 5.1) but was the user making more mistakes? Looking at the error rate in different sessions revealed no statistically significant difference in the number of errors made across sessions ($p \approx 0.74$), so the user was not picking up speed at the cost of making more mistakes.

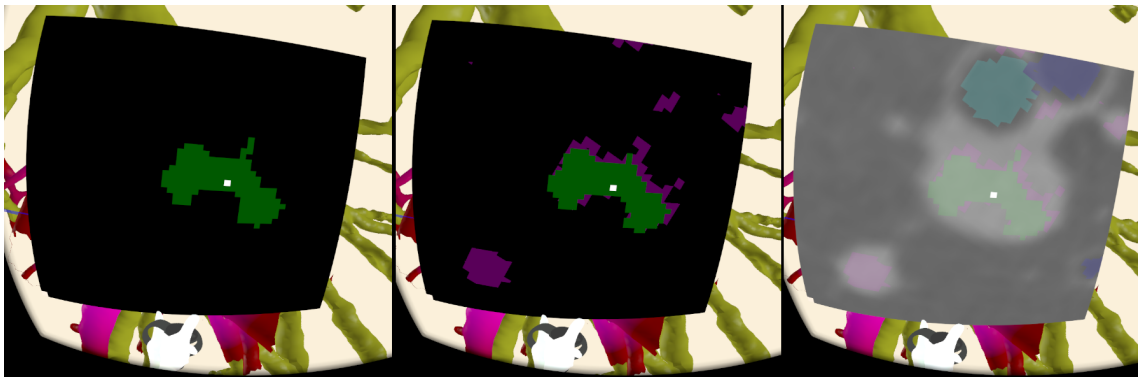


Figure 5.3: The progression steps of 3D volume loading

In the first panel, none of the volumes are loaded yet, in the second panel only the blood vessel mask is loaded and in the third panel both the airway mask and the intensity volume have been loaded (average loading time ≈ 12 seconds).

5.2.2 Matching Errors

Between the two system, only six pairs of observations consistently showed disagreement on both systems, that is, the user made a matching error on the 2D system and the VR system when working on those six pairs. All of these errors were of the type "one generation below." There was no immediately apparent vascular branching pattern specific to this group. One subject stood out where the Trachea and the airway appeared distorted. The Apical part of the left lung appeared to have very few blood vessels (see figure 5.6). In another subject, the segment labeled "LB10" was rather short (see figure 5.7). Even though a shorter target segment could be a possible cause of confusion, this wasn't unique to pairs of observations with errors, but it was also present in pairs of observations where no errors were made. There was no correlation between the length or the volume of the target airway segment and the number of errors made on the VR (p-value ≈ 0.5) or the 2D system (p-value

≈ 0.07).

In the VR application, four errors of the type "different vessel" were made. The median matching time for these observations was significantly longer than observations with other kinds of errors, indicating possible confusion. Examination of these four subjects revealed an interesting pattern. In all four subjects, there were two different vessels with branching pattern that seems to match the branching pattern of the airway thus causing the confusion. Figures 5.4 and 5.5 show two examples of this case.

Figure 5.4: Example 1 of "different vessel error"

Structures that have the same label (A or B) may appear to be companions which can be a source of confusion.

Figure 5.5: Example 2 of "different vessel error"

Structures that have the same label (A or B) may appear to be companions which can be a source of confusion.

The majority of matching errors made in the VR application are of the type "One generation below"; in fact, the number of errors made on both systems is more or less the same except for the "one generation below error." Examination of these subjects didn't reveal any specific branching patterns associated with that type of error. In the following section, we examine possible causes for confusion that may have contributed to these results.

Figure 5.6: A subject with errors on both systems

Deformed Trachea (left) Apical part of the lung having very few vessels (middle) A CT slice from the lung (right)

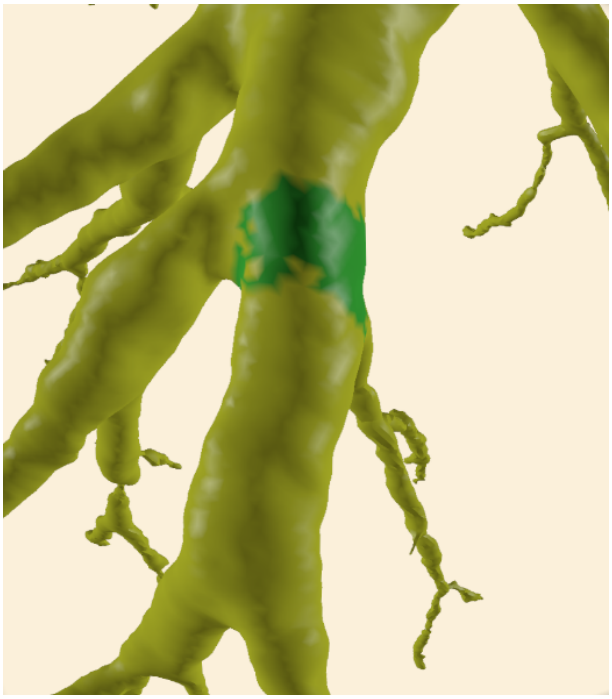


Figure 5.7: Short LB10 segment

5.2.2.1 Possible Causes of Confusion

In this application, the blood vessel mask is used to generate the surface mesh. Discontinuity in the blood vessel mask can cause confusion because it interferes with the ability of the user to track blood vessel branching pattern upstream. A discontinuity in the mask will cause the blood vessel to appear interrupted at random points, and the user will have to fill these spaces mentally or by viewing the underlying CT volume (see figure 5.8). Discontinuity of the blood vessel mask was present in many of the cases with errors.

Another possible cause of confusion is that the marching cubes algorithm fuses walls of blood vessels that come close to each other (partial volume effect); this gives the illusion that there are more branches for a blood vessel than what is actually there (see figure 5.9). A solution to this problem was proposed by Saha et al. [112] but unfortunately, it was not implemented in this work.

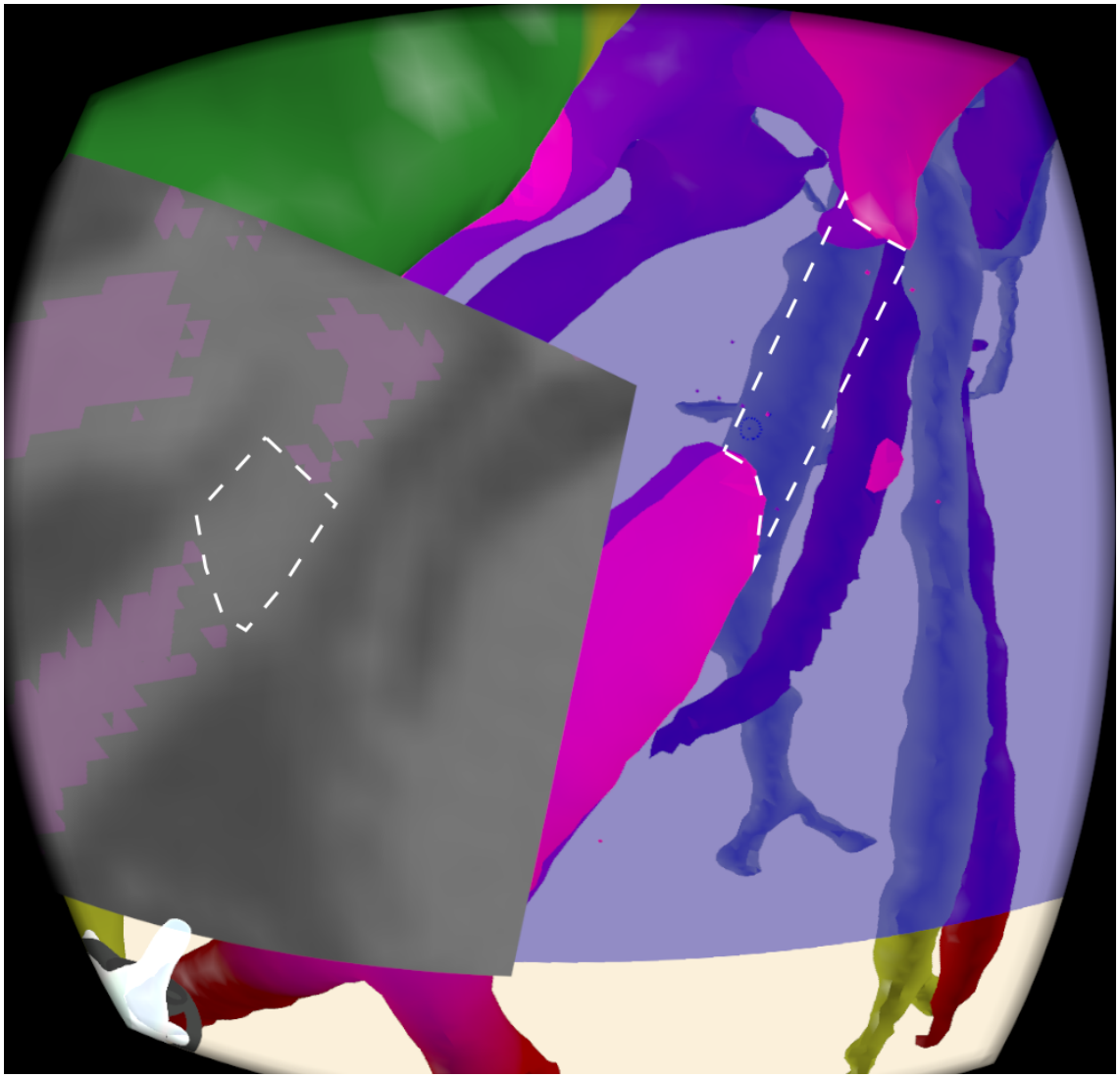


Figure 5.8: Blood vessel mask gap

The white outline shows a gap in the blood vessel mask and the corresponding gap in the blood vessel mesh

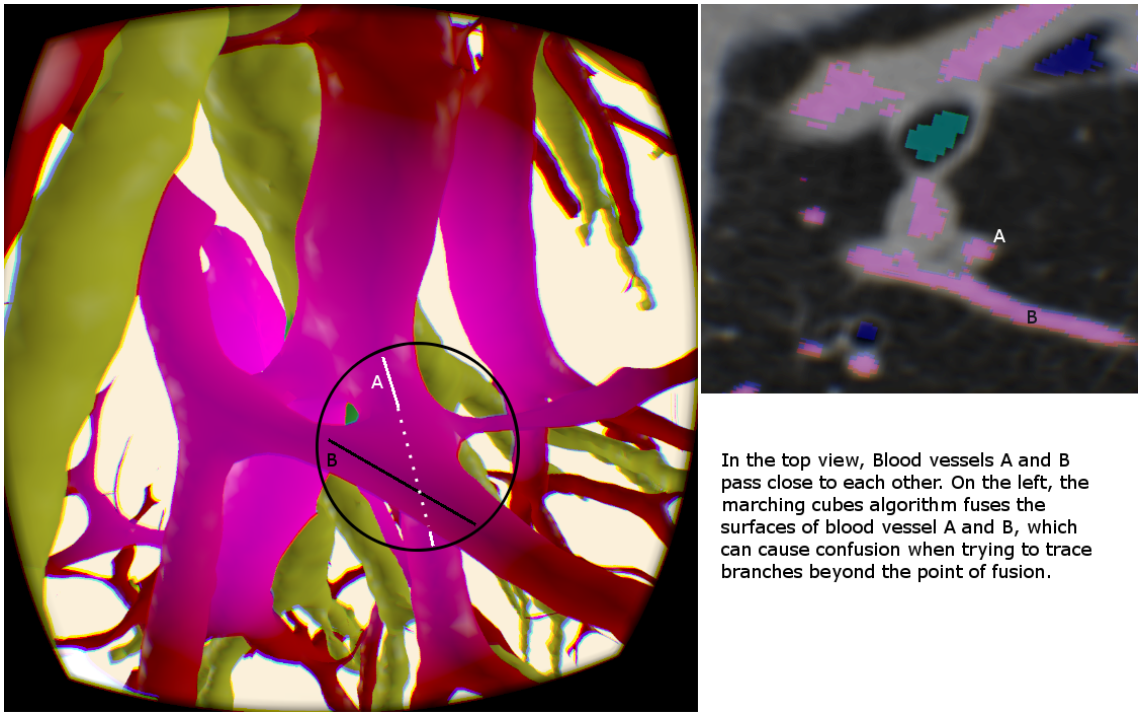


Figure 5.9: Surface fusion in vessels passing close to each other

5.2.3 Notes on the User Interface

The VR application employs a fairly simple cropping mechanism; it allows the user to change the radius of a sphere centered around the selected airway segment and crop anything outside it. This mechanism was quite useful for removing clutter and allowed the user to look at the vessel of interest more readily but there were cases where this cropping technique fell short. Ideally, the user tweaks the sphere radius until a point is reached where enough of the targeted blood vessel is visible so that a decision can be made on whether or not it is, in fact, the artery. However, it is often the case that smaller radii remove clutter but don't show enough of the targeted blood vessel while larger radii show enough of the targeted blood vessel but

add clutter to the scene (See figure 5.10). A possible solution to this problem is to implement a mechanism to highlight only a user-defined blood vessel and hide all other blood vessels. More specifically, the user will mark a seed point on a blood vessel's surface, and the application will follow the surface of that blood vessel in all directions to a user-defined distance. This approach will be affected by the surface fusion problem mentioned in the previous section. As an alternative to following the surface of the blood vessel, a skeletonization algorithm [111] can be used to extract central axes of the vascular tree and follow the central axis of blood vessels instead of their surfaces.

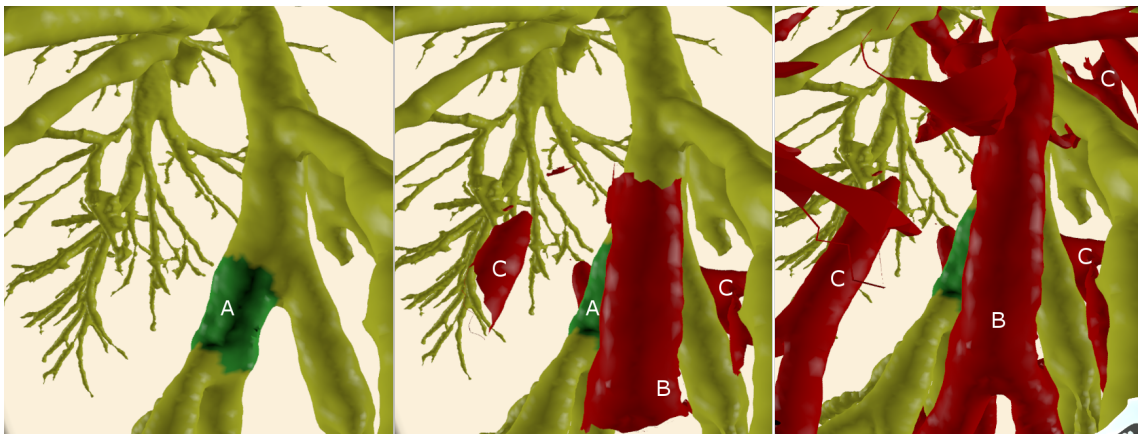


Figure 5.10: Shortcomings of the cropping mechanism

(A) Target airway (B) Targeted vessel (C) Other vessels. From left to right, the radius of the visibility sphere increases. Notice in the second panel that there aren't many irrelevant blood vessels cluttering the view, but the targeted blood vessel is cut short. Increasing the sphere radius does show more of the targeted blood vessel but also adds clutter to the view by showing irrelevant blood vessels.

CHAPTER 6
FUTURE WORK AND CONCLUSION

6.1 Future Work

This section lists suggested modifications and enhancements to the application; they are prioritized based on the impact they could have on the airway-to-vessel matching task.

6.1.1 High Priority Modifications

High priority modifications address issues that had a detrimental impact on the user's performance on the airway-to-vessel matching task. These modifications include: pre-processing vessel masks to fill gaps (see section 5.2.2.1 and figure 5.8), fixing the dips in the frame rate (see section 5.1.3), and pre-loading the 3D volume (or parts of it) to eliminate the 12-second bottleneck (see section 5.2.1 and figure 5.3).

6.1.2 Intermediate Priority Modifications

Intermediate priority modifications are modifications that are very likely to improve the user experience and can have positive effect on the users' performance on the task. These modifications include: implementing a better cropping mechanism (see section 5.2.3 and figure 5.10) and handling the surface fusion problem (see section 5.2.2.1 and figure 5.9).

6.1.3 Low Priority Modifications

Low priority modifications are modifications that enhance the look and feel of the application and could improve the rendering performance but are not likely to improve the users' performance on the matching task. These modifications include: the addition of support for ambient occlusion [11], anti-aliasing and shadow mapping, the addition of level of detail (LOD) rendering and the addition of support for volume rendering.

6.2 Conclusion

Significant advances in virtual reality technology have been made in recent years. Technology has started to catch up to the demands of virtual reality systems. We now have the hardware and software capabilities to build interactive, low-latency and high fidelity virtual reality applications. However, we are still far away from Sutherland's vision of the "Ultimate monitor"[124].

Virtual reality allows more degrees of freedom in interaction design than conventional desktop applications; this is a double-edged sword, on the one hand, it opens the door for more intuitive and creative interaction designs, but on the other hand, it can lead to confusing interaction experience. Quick iterations over the design and getting feedback is instrumental in navigating the vast search space of possible designs.

The ideal solution to the airway-to-artery problem is probably a fully-automated one, but until the technology is there, we will continue to rely on manual or semi-

automated matching.

In this work, we showcased a virtual reality visualization system; we outlined some technical challenges that should be considered when building virtual reality applications. We identified some of the places where our approach was lacking. We showed that virtual reality has the potential to be used as a tool for matching airways to arteries, interacting with and visualizing medical imaging datasets and achieve performance comparable to conventional tools.

In conclusion, we think that carefully designed virtual reality systems can be a valuable asset for biomedical applications.

REFERENCES

[1] Openvr page on github, 2017. [Online; accessed 30-November-2017].

[2] Cinder page on github, 2018. [Online; accessed 15-March-2018].

[3] Carrie P Aaron, Eric A Hoffman, Joao AC Lima, Steven M Kawut, Alain G Bertoni, Jens Vogel-Claussen, Mohammadali Habibi, Katja Hueper, David R Jacobs Jr, Ravi Kalhan, et al. Pulmonary vascular volume, impaired left ventricular filling and dyspnea: The mesa lung study. *PloS one*, 12(4):e0176180, 2017.

[4] Gunnar Ahlberg, Lars Enochsson, Anthony G Gallagher, Leif Hedman, Christian Hogman, David A McClusky, Stig Ramel, C Daniel Smith, and Dag Arvidsson. Proficiency-based virtual reality training significantly reduces the error rate for residents during their first 10 laparoscopic cholecystectomies. *The American journal of surgery*, 193(6):797–804, 2007.

[5] Sara K. Alford, Edwin J. R. van Beek, Geoffrey McLennan, and Eric A. Hoffman. Heterogeneity of pulmonary perfusion as a mechanistic image-based phenotype in emphysema susceptible smokers. *Proceedings of the National Academy of Sciences*, 107(16):7485–7490, 2010.

[6] J Gayle Beck, Sarah A Palyo, Eliot H Winer, Brad E Schwagler, and Eu Jin Ang. Virtual reality exposure therapy for ptsd symptoms after a road accident: An uncontrolled case series. *Behavior therapy*, 38(1):39–48, 2007.

[7] Meenakshi Bhalla, David P Naidich, Georgeann McGuinness, James F Gruden, Barry S Leitman, and Dorothy I McCauley. Diffuse lung disease: assessment with helical ct–preliminary observations of the role of maximum and minimum intensity projection images. *Radiology*, 200(2):341–347, 1996.

[8] James Blaha and Manish Gupta. Diplopia: A virtual reality game designed to help amblyopics. In *Virtual Reality (VR), 2014 iEEE*, pages 163–164. IEEE, 2014.

[9] C_ Botella, RM Baños, C Perpiñá, H Villa, Mu Alcaniz, and A Rey. Virtual reality treatment of claustrophobia: a case report. *Behaviour research and therapy*, 36(2):239–246, 1998.

[10] Thomas Bülow, Rafael Wiemker, Thomas Blaffert, Cristian Lorenz, and Steffen Renisch. Automatic extraction of the pulmonary artery tree from multi-slice ct data. In *Proc. of SPIE Vol*, volume 5746, page 731, 2005.

[11] Michael Bunnell. Dynamic ambient occlusion and indirect lighting. *Gpu gems*, 2(2):223–233, 2005.

[12] Paul S. Calhoun, Brian S. Kuszyk, David G. Heath, Jennifer C. Carley, and Elliot K. Fishman. Three-dimensional volume rendering of spiral ct data: Theory and method. *RadioGraphics*, 19(3):745–764, 1999. PMID: 10336201.

[13] Albert S Carlin, Hunter G Hoffman, and Suzanne Weghorst. Virtual reality and tactile augmentation in the treatment of spider phobia: a case report. *Behaviour research and therapy*, 35(2):153–158, 1997.

[14] Géry Casiez, Stéphane Conversy, Matthieu Falce, Stéphane Huot, and Nicolas Roussel. Looking through the eye of the mouse: A simple method for measuring end-to-end latency using an optical mouse. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 629–636. ACM, 2015.

[15] Charles Christiansen, Beatriz Abreu, Kenneth Ottenbacher, Kenneth Huffman, Brent Masel, and Robert Culpepper. Task performance in virtual environments used for cognitive rehabilitation after traumatic brain injury. *Archives of physical medicine and rehabilitation*, 79(8):888–892, 1998.

[16] HTC Corporation. Vive virtual reality system specs, 2018. [Online; accessed 10-January-2018].

[17] HTC Corporation. Vivepro virtual reality system specs, 2018. [Online; accessed 10-January-2018].

[18] Sony Corporation. Sony playstation vr page, 2017. [Online; accessed 10-January-2018].

[19] Craig M Coulam, James F Greenleaf, Anastasios G Tsakiris, and Earl H Wood. Three-dimensional computerized display of physiologic models and data. *Computers and Biomedical Research*, 5(2):166–179, 1972.

[20] David Couper, Lisa M LaVange, MeiLan Han, R Graham Barr, Eugene Bleecker, Eric A Hoffman, Richard Kanner, Eric Kleerup, Fernando J Martinez, Prescott G Woodruff, et al. Design of the subpopulations and intermediate outcomes in copd study (spiromics). *Thorax*, pages thoraxjnl–2013, 2013.

[21] Carolina Cruz-Neira, Daniel J Sandin, Thomas A DeFanti, Robert V Kenyon, and John C Hart. The cave: audio visual experience automatic virtual environment. *Communications of the ACM*, 35(6):64–73, 1992.

[22] Robert Wilhelm David Turner and Werner Lemberg. Freetype official website, 2017.

[23] Joann Difede and Hunter G Hoffman. Virtual reality exposure therapy for world trade center post-traumatic stress disorder: A case report. *Cyberpsychology & behavior*, 5(6):529–535, 2002.

[24] Akio Doi and Akio Koide. An efficient method of triangulating equi-valued surfaces by using tetrahedral cells. *IEICE TRANSACTIONS on Information and Systems*, 74(1):214–224, 1991.

[25] Robert A Drebin, Donna Magid, Douglas D Robertson, and Elliot K Fishman. Fidelity of three-dimensional ct imaging for detecting fracture gaps. *Journal of computer assisted tomography*, 13(3):487–489, 1989.

[26] RM Eastgate, GD Griffiths, PE Waddingham, AD Moody, TKH Butler, SV Cobb, IF Comaish, SM Haworth, RM Gregson, IM Ash, et al. Modified virtual reality technology for treatment of amblyopia. *Eye*, 20(3):370, 2006.

[27] Jan Egger, Markus Gall, Jürgen Wallner, Pedro Boechat, Alexander Hann, Xing Li, Xiaojun Chen, and Dieter Schmalstieg. Htc vive mevislab integration via openvr for medical applications. *PloS one*, 12(3):e0173972, 2017.

[28] R. E. Ellis, C. Y. Tso, J. F. Rudan, and M. M. Harrison. A surgical planning and guidance system for high tibial osteotomy. *Computer Aided Surgery*, 4(5):264–274, 1999. PMID: 10581524.

[29] PMG Emmelkamp, M Krijn, AM Hulsbosch, S De Vries, MJ Schuemie, and CAPG Van der Mast. Virtual reality treatment versus exposure in vivo: a comparative evaluation in acrophobia. *Behaviour research and therapy*, 40(5):509–516, 2002.

[30] Cass Everitt. Interactive order-independent transparency. *White paper, nVIDIA*, 2(6):7, 2001.

[31] Navid Farahani, Robert Post, Jon Duboy, Ishtiaque Ahmed, Brian J Kolowitz, Teppituk Krinchai, Sara E Monaco, Jeffrey L Fine, Douglas J Hartman, and Liron Pantanowitz. Exploring virtual reality technology and the oculus rift for the examination of digital pathology slides. *Journal of pathology informatics*, 7, 2016.

[32] Aasa Feragen, Lo Pechin, Vladlena Gorbunova, Mads Nielsen, Asger Dirksen, Joseph Reinhardt, François Lauze, and Marleen De Bruijne. An airway tree-shape model for geodesic airway branch labeling. In *Proceedings of the Third International Workshop on Mathematical Foundations of Computational Anatomy-Geometrical and Statistical Methods for Modelling Biological Shape Variability*, pages 123–134, 2011.

[33] Elliot K Fishman, Bob Drebin, Donna Magid, William W Scott Jr, DR Ney, AF Brooker Jr, LH Riley Jr, JA St Ville, EA Zerhouni, and SS Siegelman. Volumetric rendering techniques: applications for three-dimensional imaging of the hip. *Radiology*, 163(3):737–738, 1987.

[34] Elliot K Fishman, Donna Magid, Robert A Drebin, Jr AF Brooker, Jr WJ Scott, and Jr LH Riley. Advanced three-dimensional evaluation of acetabular trauma: volumetric image processing. *The Journal of trauma*, 29(2):214–218, 1989.

[35] Elliot K Fishman, Donna Magid, Derek R Ney, Robert A Drebin, and Janet E Kuhlman. Three-dimensional imaging and display of musculoskeletal anatomy. *Journal of computer assisted tomography*, 12(3):465–467, 1988.

[36] LOW FN. Peripheral visual acuity. *A.M.A. Archives of Ophthalmology*, 45(1):80–99, 1951.

[37] Anthony G Gallagher, E Matt Ritter, Howard Champion, Gerald Higgins, Marvin P Fried, Gerald Moses, C Daniel Smith, and Richard M Satava. Virtual reality simulation for the operating room: proficiency-based training as a paradigm shift in surgical skills training. *Annals of surgery*, 241(2):364, 2005.

[38] Azucena Garcia-Palacios, H Hoffman, Albert Carlin, TA Furness III, and Cristina Botella. Virtual reality in the treatment of spider phobia: a controlled study. *Behaviour research and therapy*, 40(9):983–993, 2002.

[39] Rocco Gasteiger, Mathias Neugebauer, Christoph Kubisch, and Bernhard Preim. Adapted surface visualization of cerebral aneurysms with embedded blood flow information. Citeseer.

[40] Henri Gouraud. Continuous shading of curved surfaces. *IEEE transactions on computers*, 100(6):623–629, 1971.

[41] Madeleine A. Grealy, David A. Johnson, and Simon K. Rushton. Improving cognitive function after brain injury: The use of exercise and virtual reality. *Archives of Physical Medicine and Rehabilitation*, 80(6):661 – 667, 1999.

[42] James F Greenleaf, Erik L Ritman, Craig M Coulam, Ralph E Sturm, and Earl H Wood. Computer graphic techniques for study of temporal and spatial relationships of multidimensional data derived from biplane roentgen videograms with particular reference to cardioangiography. *Computers and Biomedical Research*, 5(4):368–387, 1972.

[43] Junfeng Guo, Matthew K Fuld, Sara K Alford, Joseph M Reinhardt, and Eric A Hoffman. *The First International Workshop on Pulmonary Image Analysis*. Sep 2009.

[44] Amy K Hara, C Daniel Johnson, Judd E Reed, Richard L Ehman, and Duane M Ilstrup. Colorectal polyp detection with ct colography: two-versus three-dimensional techniques. work in progress. *Radiology*, 200(1):49–54, 1996.

[45] Gabor T Herman and Hsun Kao Liu. Three-dimensional display of human organs from computed tomograms. *Computer graphics and image processing*, 9(1):1–21, 1979.

[46] William Hibbard and David Santek. Interactivity is the key. In *Proceedings of the 1989 Chapel Hill workshop on Volume visualization*, pages 39–43. ACM, 1989.

[47] William E Higgins, Krisnan Ramaswamy, Roderick D Swift, Geoffrey McLennan, and Eric A Hoffman. Virtual bronchoscopy for three–dimensional pulmonary image assessment: state of the art and future needs. *Radiographics*, 18(3):761–778, 1998.

[48] Janne Mykland Hilde, Ingunn Skjørten, Ole Jørgen Grøtta, Viggo Hansteen, Morten Nissen Melsom, Jonny Hisdal, Sjur Humerfelt, and Kjetil Steine. Right ventricular dysfunction and remodeling in chronic obstructive pulmonary disease without pulmonary hypertension. *Journal of the American College of Cardiology*, 62(12):1103–1111, 2013.

[49] Eric A Hoffman. *"A historical perspective of heart and lung 3D Imaging"*. CRC Press, Inc., Oxford, 1991.

[50] Hunter G. Hoffman, Jason N. Doctor, David R. Patterson, Gretchen J. Carrougher, and Thomas A. Furness III. Virtual reality as an adjunctive pain control during burn wound care in adolescent patients. *Pain*, 85(1):305 – 309, 2000.

[51] Keith Horsfield. Morphometry of the small pulmonary arteries in man. *Circulation research*, 42(5):593–597, 1978.

[52] Katja Hueper, Jens Vogel-Claussen, Megha A Parikh, John HM Austin, David A Bluemke, James Carr, Jiwoong Choi, Thomas A Goldstein, Antoinette S Gomes, Eric A Hoffman, et al. Pulmonary microvascular blood flow in mild chronic obstructive pulmonary disease and emphysema. the mesa copd study. *American journal of respiratory and critical care medicine*, 192(5):570–580, 2015.

[53] Wijnand IJsselsteijn and Giuseppe Riva. Being there: The experience of presence in mediated environments. 2003.

[54] Facebook Inc. Asynchronous spacewarp documentation, 2018. [Online; accessed 04-March-2018].

[55] Facebook Inc. Oculus rift specs, 2018. [Online; accessed 04-January-2018].

[56] Facebook Inc. Oculus rift vr pc sdk, 2018. [Online; accessed 30-January-2018].

[57] Facebook Inc. Rendering to the oculus rift, 2018. [Online; accessed 30-January-2018].

[58] Krishna S Iyer, John D Newell Jr, Dakai Jin, Matthew K Fuld, Punam K Saha, Sif Hansdottir, and Eric A Hoffman. Quantitative dual-energy computed tomography supports a vascular etiology of smoking-induced inflammatory lung disease. *American journal of respiratory and critical care medicine*, 193(6):652–661, 2016.

[59] Dakai Jin, Krishna S Iyer, Eric A Hoffman, and Punam K Saha. Automated assessment of pulmonary arterial morphology in multi-row detector ct imaging using correspondence with anatomic airway branches. In *International Symposium on Visual Computing*, pages 521–530. Springer, 2014.

[60] Hans J. Johnson, M. McCormick, L. Ibáñez, and The Insight Software Consortium. *The ITK Software Guide*. Kitware, Inc., third edition, 2013.

[61] Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. Dual contouring of hermite data. In *ACM transactions on graphics (TOG)*, volume 21, pages 339–346. ACM, 2002.

[62] Jens N Kaftan, Atilla P Kiraly, David P Naidich, and Carol L Novak. A novel multi-purpose tree and path matching algorithm with application to airway trees. *SPIE Med. Im*, 6143:215–224, 2006.

[63] Ron Kikinis, Steve D Pieper, and Kirby G Vosburgh. 3d slicer: a platform for subject-specific image analysis, visualization, and clinical support. In *Intraoperative imaging and image-guided therapy*, pages 277–289. Springer, 2014.

[64] Hyun Cheol Kim, Seong Jin Park, Sung II Park, Sang Heum Park, Hyun Jun Kim, Hyeong Cheol Shin, Won Kyung Bae, II Young Kim, and Hae Kyung Lee. Multislice ct cholangiography using thin-slab minimum intensity projection and multiplanar reformation in the evaluation of patients with suspected biliary obstruction: preliminary experience. *Clinical imaging*, 29(1):46–54, 2005.

[65] Jens Kruger and Rüdiger Westermann. Acceleration techniques for gpu-based volume rendering. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 38. IEEE Computer Society, 2003.

[66] Christoph Kubisch, Sylvia Glaßer, Mathias Neugebauer, and Bernhard Preim. Vessel visualization with volume rendering. In *Visualization in Medicine and Life Sciences II*, pages 109–132. Springer, 2012.

[67] Jennifer J Kulynych, Katalin Vladar, Douglas W Jones, and Daniel R Weinberger. Gender differences in the normal lateralization of the supratemporal cortex: Mri surface-rendering morphometry of heschl's gyrus and the planum temporale. *Cerebral cortex*, 4(2):107–118, 1994.

[68] Sergei N Kurenov, Ciprian Ionita, Dan Sammons, and Todd L Demmy. Three-dimensional printing to facilitate anatomic study, device development, simulation, and planning in thoracic surgery. *The Journal of thoracic and cardiovascular surgery*, 149(4):973–979, 2015.

[69] BS Kuszyk, DG Heath, DR Ney, DA Bluemke, BA Urban, TP Chambers, and EK Fishman. Ct angiography with volume rendering: imaging findings. *AJR. American journal of roentgenology*, 165(2):445–448, 1995.

[70] Fakespace Labs. Fakespace labs research tools. [Online; accessed 02-April-2018].

[71] Philippe Lacroute and Marc Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 451–458. ACM, 1994.

[72] Christian R Larsen, Jette L Soerensen, Teodor P Grantcharov, Torur Dalsgaard, Lars Schouenborg, Christian Ottosen, Torben V Schroeder, and Bent S Ottesen. Effect of virtual reality training on laparoscopic surgery: randomised controlled trial. *Bmj*, 338:b1802, 2009.

[73] Kate Laver, Stacey George, Susie Thomas, Judith E Deutsch, and Maria Crotty. Virtual reality for stroke rehabilitation. *Stroke*, 43(2):e20–e21, 2012.

[74] Andreas Markus Loening and Sanjiv Sam Gambhir. Amide: a free software tool for multimodality medical image analysis. *Molecular imaging*, 2(3):15353500200303133, 2003.

[75] Patricia ML Lopes, Carla R Moreira, Andréia Perrella, José L Antunes, and Marcelo GP Cavalcanti. 3-d volume rendering maxillofacial analysis of angular measurements by multislice ct. *Oral Surgery, Oral Medicine, Oral Pathology, Oral Radiology and Endodontics*, 105(2):224–230, 2008.

[76] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, August 1987.

[77] Abrash M. What vr could, should, and almost certainly will be within two years.

[78] Camilla Berglund Marcus Geelnard. Graphics library framework official website, 2017.

[79] Ken Marin, Lorensen Bill Schroeder, Will, and other VTK contributors. Vtk: vtksmoothpolydatafilter class reference.

[80] Michael P Marks, Sandy Napel, John E Jordan, and Dieter R Enzmann. Diagnosis of carotid artery disease: preliminary experience with maximum-intensity-projection spiral ct angiography. *AJR. American journal of roentgenology*, 160(6):1267–1271, 1993.

[81] Jeffrey L Marsh and Michael W Vannier. Surface imaging from computerized tomographic scans. *Surgery*, 94(2):159–165, 1983.

[82] Shin Matsuoka, George R. Washko, Mark T. Dransfield, Tsuneo Yamashiro, Raul San Jose Estepar, Alejandro Diaz, Edwin K. Silverman, Samuel Patz, and Hiroto Hatabu. Quantitative ct measurement of cross-sectional area of small pulmonary vessel in copd: Correlations with emphysema and airflow limitation. *Academic Radiology*, 17(1):93 – 99, 2010.

[83] Louis B. Rosenberg McDonald, John S. and Don Stredned. Virtual reality technology applied to anesthesiology. In *Interactive Technology and the New Paradigm for Healthcare*, chapter 38, page 237. IOS Press, Washington, DC, 1995.

[84] Morgan McGuire and Louis Bavoil. Weighted blended order-independent transparency. *Journal of Computer Graphics Techniques*, 2013.

[85] Richard F. McLaughlin, Walter S. Tyler, and Robert O. Canada. A study of the subgross pulmonary anatomy in various mammals. *American Journal of Anatomy*, 108(2):149–165.

[86] Marcelo E. Magallon Milan Ikits and Lev Povalahev. Opengl extension wrangler library official website, 2017.

[87] Judi Moline. *Virtual environments for health care*, volume 5740. DIANE Publishing, 1995.

[88] Craig D. Murray, Stephen Pettifer, Toby Howard, Emma L. Patchick, Fabrice Caillette, Jai Kulkarni, and Candy Bamford. The treatment of phantom limb pain using immersive virtual reality: Three case studies. *Disability and Rehabilitation*, 29(18):1465–1469, 2007. PMID: 17729094.

[89] S Napel, GD Rubin, and RB Jeffrey. Sts-mip: a new reconstruction technique for ct of the chest. *Journal of computer assisted tomography*, 17(5):832838, 1993.

[90] Sandy Napel, MP Marks, GD Rubin, MD Dake, CH McDonnell, SM Song, DR Enzmann, and RB Jeffrey Jr. Ct angiography with spiral ct and maximum intensity projection. *Radiology*, 185(2):607–610, 1992.

[91] User Newbie. Opengl: Rendering two transparent planes intersecting each other: impossible or not?, 2010. [Online; accessed 05-February-2018].

[92] Timothy S Newman and Hong Yi. A survey of the marching cubes algorithm. *Computers & Graphics*, 30(5):854–879, 2006.

[93] Bui Tuong Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, June 1975.

[94] Mark L Purnell, Andrew I Larson, and William Clancy. Anterior cruciate ligament insertions on the tibia and femur and their relationships to critical bony landmarks using high-resolution volume-rendering computed tomography. *The American journal of sports medicine*, 36(11):2083–2090, 2008.

[95] Stefan Rahm, Karl Wieser, Ilhui Wicki, Livia Holenstein, Sandro F Fucentese, and Christian Gerber. Performance of medical students on a virtual reality simulator for knee arthroscopy: an analysis of learning curves and predictors of performance. *BMC surgery*, 16(1):14, 2016.

[96] Narayana Dlv Rao, Manpreet Singh Gulati, Shashi Bala Paul, Girish Kumar Pande, Peush Sahni, and Tushar Kanti Chattopadhyay. Three-dimensional helical computed tomography cholangiography with minimum intensity projection in gallbladder carcinoma patients with obstructive jaundice: Comparison with magnetic resonance cholangiography and percutaneous transhepatic cholangiography. *Journal of gastroenterology and hepatology*, 20(2):304–308, 2005.

[97] Vassilios Raptopoulos, Panos Prassopoulos, Ram Chuttani, MM McNicholas, John D McKee, and Herbert Y Kressel. Multiplanar ct pancreatography and distal cholangiography with minimum intensity projections. *Radiology*, 207(2):317–324, 1998.

[98] L. Reid. The angiogram and pulmonary artery structure and branching (in the normal and with reference to disease). *Proc R Soc Med*, 58(9):681–684, Sep 1965. 4220448[pmid].

[99] Martine Remy-Jardin, Jacques Remy, Bernard Gosselin, Marie Christine Copin, Alain Wurtz, and Alain Duhamel. Sliding thin slab, minimum intensity projection technique in the diagnosis of emphysema: histopathologic-ct correlation. *Radiology*, 200(3):665–671, 1996.

[100] Giuseppe Riva, Monica Bacchetta, Margherita Baruffi, Silvia Rinaldi, and Enrico Molinari. Virtual reality based experiential cognitive treatment of anorexia nervosa. *Journal of Behavior Therapy and Experimental Psychiatry*, 30(3):221 – 230, 1999.

[101] Giuseppe Riva, Monica Bacchetta, Margherita Baruffi, Silvia Rinaldi, Francesco Vincelli, and Enrico Molinari. Virtual reality-based experiential cognitive treatment of obesity and binge-eating disorders. *Clinical Psychology and Psychotherapy*, 7(3):209–219, 2000.

[102] Albert Skip Rizzo and Gerard Jounghyun Kim. A swot analysis of the field of virtual reality rehabilitation and therapy. *Presence: Teleoperators & Virtual Environments*, 14(2):119–146, 2005.

[103] Richard A. Robb, James F. Greenleaf, Erik L. Ritman, Steven A. Johnson, Jerome D. Sjostrand, Gabor T. Herman, and Earl H. Wood. Three-dimensional visualization of the intact thorax and contents: A technique for cross-sectional reconstruction from multiplanar x-ray views. *Computers and Biomedical Research*, 7(4):395 – 419, 1974.

[104] Richard Arlin Robb, Dennis P Hanson, RA Karwoski, AG Larson, EL Workman, and MC Stacy. Analyze: a comprehensive, operator-interactive software package for multidimensional medical image display and analysis. *Computerized Medical Imaging and Graphics*, 13(6):433–454, 1989.

[105] Antoine Rosset, Luca Spadola, and Osman Ratib. Osirix: an open-source software for navigating in multidimensional dicom images. *Journal of digital imaging*, 17(3):205–216, 2004.

[106] Barbara O Rothbaum, Larry F Hodges, David Ready, Ken Graap, and Renato D Alarcon. Virtual reality exposure therapy for vietnam veterans with posttraumatic stress disorder. *The Journal of clinical psychiatry*, 2001.

[107] Barbara Olasov Rothbaum, Larry F Hodges, Rob Kooper, Dan Opdyke, et al. Effectiveness of computer-generated (virtual reality) graded exposure in the treatment of acrophobia. *The American journal of psychiatry*, 152(4):626, 1995.

[108] Geoffrey D Rubin, Christopher F Beaulieu, Vincent Argiro, Helmut Ringl, Alexander M Norbash, John F Feller, Michael D Dake, R Brooke Jeffrey, and Sandy Napel. Perspective volume rendering of ct and mr images: applications for endoscopic imaging. *Radiology*, 199(2):321–330, 1996.

[109] Curtis T Rueden, Johannes Schindelin, Mark C Hiner, Barry E DeZonia, Alison E Walter, Ellen T Arena, and Kevin W Eliceiri. Imagej2: Imagej for the next generation of scientific image data. *BMC bioinformatics*, 18(1):529, 2017.

[110] Greg S Ruthenbeck and Karen J Reynolds. Virtual reality for medical training: the state-of-the-art. *Journal of Simulation*, 9(1):16–26, 2015.

[111] Punam K Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja. A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters*, 76:3–12, 2016.

[112] Punam K Saha, Zhiyun Gao, Sara K Alford, Milan Sonka, and Eric A Hoffman. Topomorphologic separation of fused isointensity objects via multiscale opening: Separating arteries and veins in 3-d pulmonary ct. *IEEE transactions on medical imaging*, 29(3):840–851, 2010.

[113] Karen Morgan Hans B. Sieburg Rudy Mattheus Satava, Richard M. and Jens P. Christensen. *Interactive Technology and the New Paradigm for Healthcare*. IOS Press, Wasington, DC, 1995.

[114] Richard M. Satava. Emerging medical applications of virtual reality: A surgeon's perspective. *Artificial Intelligence in Medicine*, 6(4):281 – 288, 1994. Virtual Reality in Medicine.

[115] Kenji Sato, Satoshi Fukumori, Takashi Matsusaki, Tomoko Maruo, Shinichi Ishikawa, Hiroyuki Nishie, Ken Takata, Hiroaki Mizuhara, Satoshi Mizobuchi, Hideki Nakatsuka, et al. Nonimmersive virtual reality mirror visual feedback therapy and its application for the treatment of complex regional pain syndrome: an open-label pilot study. *Pain medicine*, 11(4):622–629, 2010.

[116] Shiro Satoh, Shinichi Ohdama, and Hitoshi Shibuya. Sliding thin slab, minimum intensity projection imaging for objective analysis of emphysema. *Radiation medicine*, 24(6):415–421, 2006.

[117] William J Schroeder, Kenneth M Martin, and William E Lorensen. The design and implementation of an object-oriented toolkit for 3d graphics and visualization. In *Proceedings of the 7th conference on Visualization'96*, pages 93–ff. IEEE Computer Society Press, 1996.

[118] William J Schroeder, Jonathan A Zarge, and William E Lorensen. Decimation of triangle meshes. In *ACM Siggraph Computer Graphics*, volume 26, pages 65–70. ACM, 1992.

[119] Neal E Seymour, Anthony G Gallagher, Sanziana A Roman, Michael K Obrien, Vipin K Bansal, Dana K Andersen, and Richard M Satava. Virtual reality training improves operating room performance: results of a randomized, double-blinded study. *Annals of surgery*, 236(4):458, 2002.

[120] Wan-Yin Shi, Yong-Dong Li, Ming-Hua Li, Bin-Xian Gu, Shi-Wen Chen, Wu Wang, Bei-Lei Zhang, Min Li, et al. 3d rotational angiography with volume rendering: The utility in the detection of intracranial aneurysms. *Neurology India*, 58(6):908, 2010.

[121] Jered P Sieren, John D Newell Jr, R Graham Barr, Eugene R Bleecker, Nathan Burnette, Elizabeth E Carretta, David Couper, Jonathan Goldin, Junfeng Guo, MeiLan K Han, et al. Spiromics protocol for multicenter quantitative computed tomography to phenotype the lungs. *American journal of respiratory and critical care medicine*, 194(7):794–806, 2016.

[122] Adeel Soomro. Console latency: Exploring video game input lag, 2015. [Online; accessed 10-January-2018].

[123] Dorothy Strickland. Virtual reality for the treatment of autism. *Studies in Health Technology and Informatics*, 44, 1997.

[124] Ivan E Sutherland. The ultimate display. *Multimedia: From Wagner to virtual reality*, 1965.

[125] Kedarnath Thangudu, Mark Kilgard, Pat Brown, and Jason Schmidt. Nv_stereo_view_rendering opengl extension proposal, 2016.

[126] Kedarnath Thangudu, Brown Kilgard Werness, Esser, and Schmidt. Nv_clip_space_w_scaling opengl extension proposal, 2017.

[127] Jayaram K Udupa. *"Computer Aspects of 3D imaging in medicine: A Tutorial"*. CRC Press, Inc., Oxford, 1991.

[128] René WM Underberg, Frank J Lagerwaard, Ben J Slotman, Johan P Cuijpers, and Suresh Senan. Use of maximum intensity projections (mip) for target volume generation in 4dct scans for lung cancer. *International Journal of Radiation Oncology Biology Physics*, 63(1):253–260, 2005.

[129] Will Usher, Pavol Klacansky, Frederick Federer, Peer-Timo Bremer, Aaron Knoll, Jeff Yarch, Alessandra Angelucci, and Valerio Pascucci. A virtual reality visualization tool for neuron tracing. *IEEE transactions on visualization and computer graphics*, 24(1):994–1003, 2018.

[130] David C. Van Essen, Heather A. Drury, James Dickson, John Harwell, Donna Hanlon, and Charles H. Anderson. An integrated software suite for surface-based analyses of cerebral cortex. *Journal of the American Medical Informatics Association*, 8(5):443–459, 2001.

[131] Michael W Vannier, Charles F. Hildebolt, Donald E Gayou, and Jeffrey L Marsh. *"Introduction to 3D Imaging"*. CRC Press, Inc., Oxford, 1991.

[132] Michael W Vannier, Jeffrey L Marsh, and James O Warren. Three dimensional ct reconstruction images for craniofacial surgical planning and evaluation. *Radiology*, 150(1):179–184, 1984.

[133] David J Vining, Kun Liu, Robert H Choplin, and Edward F Haponik. Virtual bronchoscopy: relationships of virtual reality endobronchial simulations to actual bronchoscopic findings. *Chest*, 109(2):549–553, 1996.

[134] Christopher M. Walker, Melissa L. Rosado de Christenson, Santiago Martnez-Jimnez, Jeffrey R. Kunin, and Brandt C. Wible. Bronchial arteries: Anatomy, function, hypertrophy, and anomalies. *RadioGraphics*, 35(1):32–49, 2015. PMID: 25590386.

[135] Jerold W Wallis, Tom R Miller, Charles A Lerner, and Eric C Kleerup. Three-dimensional display in nuclear medicine. *IEEE Transactions on Medical Imaging*, 8(4):297–230, 1989.

[136] Lai Khin Wee, Hum Yan Chai, and Eko Supriyanto. Surface rendering of three dimensional ultrasound images using vtk. 2011.

[137] Ewald R Weibel. *Morphometry of the human lung*. Springer, 1965.

[138] J. Michael Wells, George R. Washko, MeiLan K. Han, Naseer Abbas, Hrudaya Nath, A. James Mamary, Elizabeth Regan, William C. Bailey, Fernando J. Martinez, Elizabeth Westfall, Terri H. Beaty, Douglas Curran-Everett, Jeffrey L. Curtis, John E. Hokanson, David A. Lynch, Barry J. Make, James D. Crapo, Edwin K. Silverman, Russell P. Bowler, and Mark T. Dransfield. Pulmonary arterial enlargement and acute exacerbations of copd. *New England Journal of Medicine*, 367(10):913–921, 2012. PMID: 22938715.

[139] Lee Alan Westover. *Splatting: a parallel, feed-forward volume rendering algorithm*. PhD thesis, University of North Carolina at Chapel Hill Chapel Hill, NC, 1991.

[140] Bob G Witmer and Michael J Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence*, 7(3):225–240, 1998.

[141] Ling Zhang, Beatriz C Abreu, Gary S Seale, Brent Masel, Charles H Christiansen, and Kenneth J Ottenbacher. A virtual reality environment for evaluation of a daily living skill in brain injury rehabilitation: reliability and validity1. *Archives of physical medicine and rehabilitation*, 84(8):1118–1124, 2003.

[142] Jingbo Zhao, Robert S Allison, Margarita Vinnikov, and Sion Jennings. Estimating the motion-to-photon latency in head mounted displays. In *Virtual Reality (VR), 2017 IEEE*, pages 313–314. IEEE, 2017.