# COMPARING VASCULAR VISUALIZATION TECHNIQUES WITH GAMIFICATION

Andrey Titov

A thesis
in
The Department
of
Gina Cody School of Engineering and Computer Science

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montréal, Québec, Canada

December 2020

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Andrey Titov

Entitled: Comparing Vascular Visualization Techniques with Gamification

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Charalambos Poullis — Chair

Charalambos Poullis — Examiner

Thomas Fevens — Examiner

Marta Kersten-Oertel — Thesis Supervisor(s)

_____ Thesis Supervisor(s)

Approved by _____
Leila Kosseim — Chair of Department or Graduate Program Director

Dean Mourad Debbabi — 17-12-2020

# Abstract

Comparing Vascular Visualization Techniques with Gamification

Andrey Titov

One of the bottlenecks of visualization research is the lack of volunteers for studies that evaluate new methods and paradigms. However, with the technological advancement in mobile hardware and the availability of online marketplaces, it has become possible to perform user studies using the gamification paradigm. Furthermore, the possibility of implementing volume rendering, a computationally expensive method, on mobile devices opened the door to the use of gamification in the context of medical image visualization studies.

In this thesis, we describe a gamified study that we performed with the goal of comparing several cerebrovascular visualization techniques. The study was implemented in the form of a mobile game, "Connect Brain", that was developed and distributed on both Android and iOS platforms. Connect Brain features two minigames, one of which asks the player to make decisions about the depth of different vessels, and the second one that has them determine if two vessels are connected. The gameplay data was sent to a server and analyzed to determine the most effective visualization techniques.

The results of our study confirmed that fog, chroma-depth and pseudo chromdepth are among the most effective depth perception cues, similar to previous studies. However, our results differed for the edge enhancement cue, which we found to be one of the worst in terms of depth perception. The gamification paradigm, which allowed us to collect more data samples from more participants than similar studies, had more similar results to the larger studies. This suggests the importance of having a large number of subjects and trials when evaluating visualization paradigms.

Although gamification presented some challenges, we believe that this technique has great potential for future studies in not only medical image visualization but for the other clinical tasks.

# Acknowledgments

First, I would like to deeply thank my supervisor Professor Marta Kersten-Oertel for taking me as her research assistant and guiding me throughout my master's studies. I have greatly appreciated her mentorship skills that made my graduate studies a great experience, despite some occasional obstacles that I encountered in the process. Thanks to her passion in the subject, I got interested in the domains of medical imaging and academia in general, which I was hesitant about initially. Additionally, I would like to thank the students of the Applied Perception Lab and Professor Simon Drouin for their help and feedback regarding my project.

Second, I am very grateful to my family and friends who have always encouraged and supported me both during the ups and downs of my academic life. I am also thankful to everyone who participated in my user study and allowed me to gather all the necessary information for the data analysis. As it turned out, it is very hard to get an initial player base, simply because of how many other apps exist already. So, I am very thankful to those who participated!

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

With the advancement of display technologies, computer and graphics hardware, and interaction devices, the visualization of complex, multimodal, and high dimensional images has become more preeminent across a variety of devices. Today, smartphones and tablets allow for real-time rendering using modern graphic APIs such as OpenGL ES [1], Vulkan [2] and Metal [3]. While the performance of these mobile devices is still significantly lower than that of desktop machines, they nevertheless offer many modern GPU features such as 3D texturing and compute shaders.

One of the application areas of computer graphics that requires significant computational power is the field of medical imaging. This field concentrates on the visualization of data acquired from medical scans, such as computed tomography (CT) or magnetic resonance (MR) [1]. Volume rendering can be used to create 3D anatomical models of these scans for diagnostic purposes, surgical planning, and surgical guidance. Unlike indirect rendering, which only focuses on rendering the surfaces of an object, volume rendering is a direct technique where the volumetric data set, obtained through sampling, simulation or modelling techniques, is reconstructed into a three–dimensional model [2]. Over the last years, volume rendering has gained more widespread acceptance from the medical community and clinicians have begun to welcome 3D visualization [3].

One type of medical image used for diagnosing abnormalities in the vasculature of the brain or for planning neurosurgical procedures is cerebral angiography [4].

[1] https://www.khronos.org/opengles/ (Last visited December 1, 2020)
[2] https://www.khronos.org/vulkan/ (Last visited December 1, 2020)
[3] https://developer.apple.com/metal/ (Last visited December 1, 2020)

Here, the blood vessels inside of the brain are imaged using X-ray, MR or CT [5]. Due to the inherent complex nature of the cerebral blood vessels, visualizing them in a way so that allows for intuitive three-dimensional and depth understanding has been a focus of much research [4, 5, 6]. Multiple techniques have been developed to visualize this type of data and multiple studies have been performed with the goal of comparing different visualization techniques to determine their strengths and weaknesses [7, 4, 8, 5, 9, 10, 11, 12]. Traditionally, such studies were performed in a lab environment where subjects were completing specific experimental tasks (e.g. determining which of two vessels was closer or further) under the supervision of the researcher. However, this type of study has certain drawbacks including, a small pool of participants [13] and a high time and/or monetary cost per participant [14].

With the improvements of mobile GPUs, progress has been made at implementing volume rendering on mobile devices. Some optimization techniques such as the ones presented by Alcocer *et. al.* [15] and Noguera *et. al.* [16] focused on generally reducing the number of samples to achieve better framerates, while other techniques such as the one presented by Mobeen *et. al.* [17] focused more on reducing the number of rendering passes. Other algorithms, such as the one presented by Hachaj *et. al.* [18] explored a client-server approach, where rendering was done on a more powerful machine and the final image was then sent to the mobile device.

The focus of the research described in this dissertation was to explore the use of gamification [14] to collect data from a large and diverse population in order to compare the effectiveness of different volume visualization techniques in the context of cerebral vascular imaging. In order to do this, we took advantage of modern mobile device hardware and the accessibility of the current app marketplaces. Specifically, we developed "Connect Brain", a mobile App that renders the cerebral vasculature (captured using CTA) and distributed the game in Google Play [4] and the App Store [5]. The game play data was analyzed and the results compared to previous studies in this field.

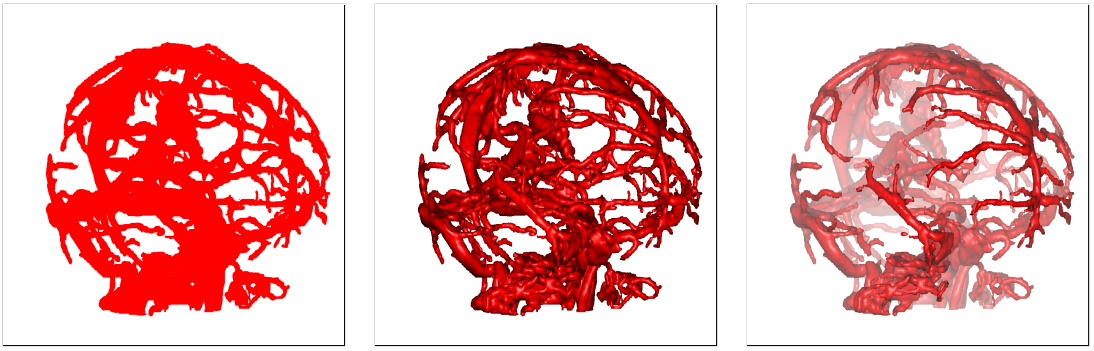
[4]Google Play link:
https://play.google.com/store/apps/details?id=ca.andreytitov.connectbrain
(Published May 2020)

[5]App Store link:
https://apps.apple.com/us/app/id1524359191
(Published Aug 2020)

## 1.1 CT Angiography Visualization

In this thesis, we focus on the visualization of vascular structures that come from computed tomography angiography (CTA) scans. Such scans are created by injecting a contrast substance into the blood stream of a patient [5] and then performing a series of 2D CT scans that can then be assembled into a single 3D volumetric dataset. The resulting 3D CTA data can be used for diagnostic purposes, surgical planning, and surgical guidance [5]. Yet, the visualization of the CTA data needs to be considered as multiple factors can hinder the depth and spatial understanding of the CTA dataset. These include: (1) the complex branching, (2) the anatomical difference between patients and (3) the set of visualization techniques and technologies that may be used to overcome these challenges might be limited due to specific hardware requirements or the limitations of the clinical environment[4, 19].

In order to have a good spatial understanding of a 3D CTA, volume rendering can be used. However, a simple volume rendered image of the 3D CTA lacks depth information (Figure 1), thus multiple visualization techniques have been developed or explored to overcome the aforementioned challenges and improve three-dimensional perception of these medical images. These methods include, using perceptual cues [4, 5, 11], illustrative techniques [12, 7, 9, 10], and general-purpose shading techniques such as Blinn-Phong [20], chromadepth [21] and edge enhancement [22].



(a) No Cue (b) Blinn-Phong (c) Fog + Blinn-Phong

Figure 1: Comparison between a visualization with no cues (a), Blinn-Phong shading (b) and fog cue and Blinn-Phong combined (b)

## 1.2 Gamification

Gamification is the process of adding game play elements to a non-game context in order to make a rather boring task more interesting [14]. The core idea of using gamification for user studies is to make the experimental task a game that is fun to play and may encourage more users to participate and thus allow for more data collection. The game can be distributed online, which results in a higher number of participants and higher diversity between the participants than in a lab setting [13]. Also, the time and/or financial cost per participant is lower. In this sense, gamification is very similar to crowdsourcing [23], which allows distribution of a study online, but without game elements. The advantage that gamification has over crowdsourcing is that in an environment which is not controlled by the researcher, it may allow for a higher quality of data as players are motivated to perform well, while in crowdsourcing, participants might only be interested in monetary incentives [13]. Gamification also has a number of disadvantages compared to both crowdsourcing and a traditional lab study, such as the fact that not every study can be transformed into a game and that adding game elements requires additional development time [14].

## 1.3 Contributions

In this research, we explored the use of mobile gamification to perform a study on the comparison of specific cerebrovascular visualization techniques. The game "Connect Brain" was distributed on Google Play and in the App Store. Players of the game are presented with vessel structures shaded with different volume rendered visualization techniques, and they have to perform simple tasks such as deciding upon the depth relationship or the connectivity between different vessels (See Figure 2). The volume rendering approach used in the game was described by Kruger *et. al.* [24]. To achieve real-time rendering on mobile devices, we employed the 3D Chamfer distance algorithm [25]. Gameplay data related to, if the player completed the task correctly, the decision time and the $(x, y, z)$ distance between the points, was collected. All this information was then sent to a server, and was later analyzed. In the course of conducting the research the following was determined:

1. Aerial perspective, chromadepth and pseudo chromadepth result in the best

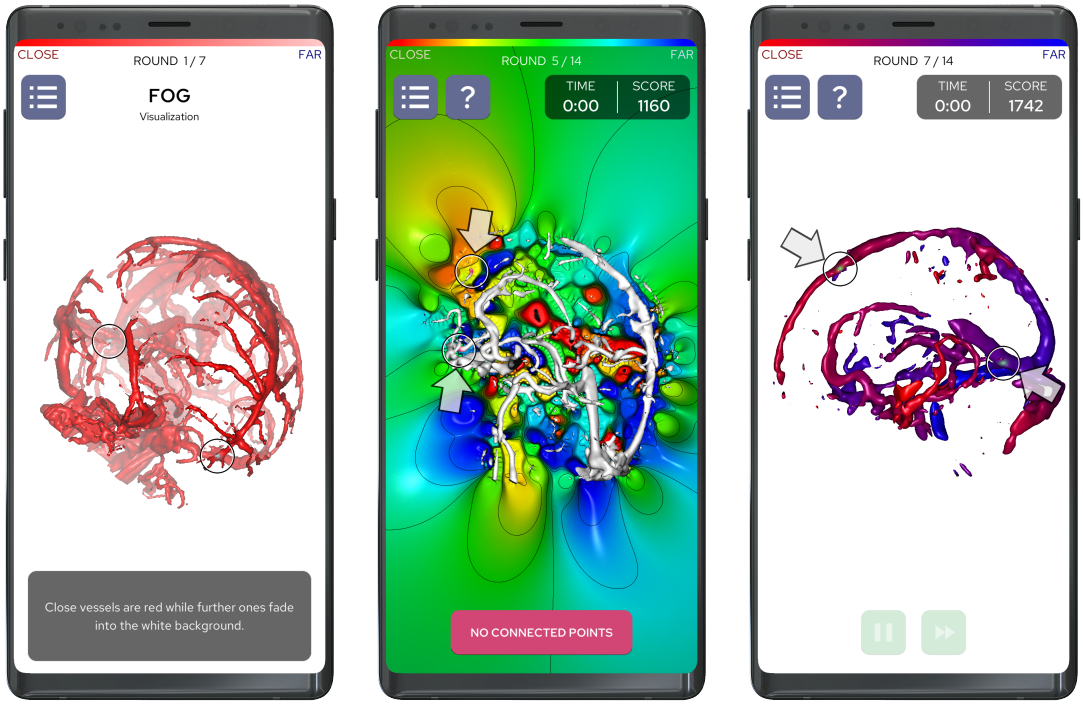



Figure 2: Screenshots from our published game, Connect Brain. Phone mockup source: http://www.freepik.com, Designed by zlatko_plamenov / Freepik

relative depth perception between vessels and the best decision time, while for connectivity, the vessel visualization techniques doesn't significantly alter the results (the correctness or the decision time).

2. The gamification paradigm allows for the collection of more data samples from more participants. Our results were more similar to the larger studies, suggesting the importance of having a large number of subjects and trials when evaluating visualization paradigms.

3. The biggest drawback of using mobile gamification in the domain of medical image visualization is the long development time. It is due both to the limitations of the mobile hardware, requiring additional optimizations, as well as the challenge of adding fun game mechanics to the experiment.

4. Implementation of volume rendering on mobile devices still requires significant optimization for real-time rendering, with the biggest bottleneck being the texture lookups. The optimization often refers to either reducing the number of

texture samples or by employing parallelization (both on the CPU and the GPU).

5. In terms of game design, having a competitive element (e.g. a leaderboard) and allowing players to understand their mistakes increases player's interest in the game and encourages the user to play for a longer time.

## 1.4 Organization of Thesis

The remainder of the thesis is organized as follows. In Chapter 2, we describe volume rendering, a subfield of computer graphics that concentrates on the visualization of discretely sampled data and is extensively used in medical visualization. In Chapter 3, we present the related work on vascular volume visualization, the development of the "Connect Brain" game, and the analysis of the results of the captured game play data. Finally, in Chapter 4, we conclude the thesis and describe avenues of future work that could be pursued, specifically in regards to improving and adding features to the developed game.

# Chapter 2

# Volume Rendering

In the following chapter, we describe the theory behind volume rendering, as well as describe the most popular volume rendering methods.

Volume rendering is a technique that is used to visualize three-dimensional data by projecting it into a two-dimensional image [26]. This technique is often used in medical image visualization since it allows rendering patient-specific data obtained from, for example, computed tomography (CT), magnetic resonance imaging (MRI) or X-ray scans [5]. These scans consist of a series of stacked 2D slices (with samples along each line of the slice) that combined create a 3D array of data [8], i.e. the volume or 3D texture (Figure 3). Each individual sample in this aggregated volume is called a voxel. The data represented in a voxel depends on the type of imaging, for example, in CT, the voxel at position $(x, y, z)$ stores the X–ray absorption coefficient which is measured in Hounsfield units [27]. Rather than visualizing each slice of the volume individually, volume rendering allows for 3D visualization of the anatomy that can be rotated and seen from any viewpoint, thus providing more contextual information and spatial understanding.

The volume rendering pipeline (Figure 4), which allows for reconstruction of a volume, was described in the context of medical image visualization by Preim *et al.* [28, p. 138]. The authors outline the steps used to render a medical dataset into a 2D image as follows:

1. **Filtering:** The data may be filtered to enhance the image quality, for example, by denoising or smoothing the volume.

2. **Segmentation:** The goal of this step is to select certain regions of interest of the

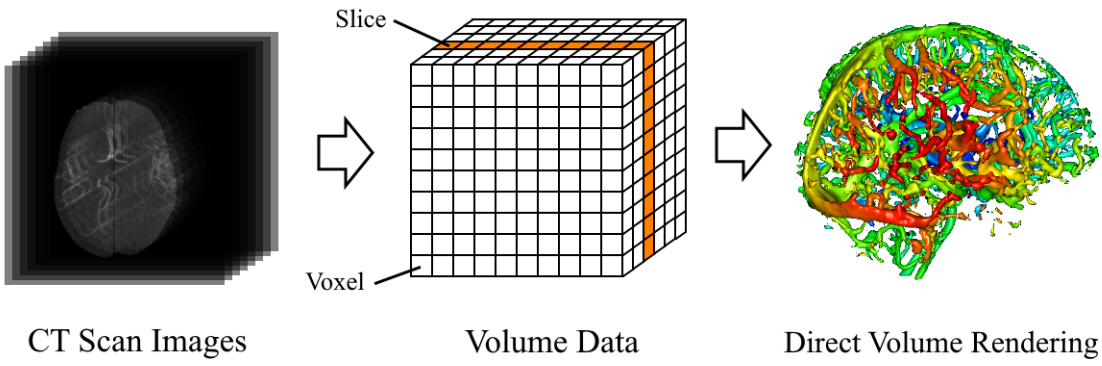



Figure 3: Volumetric data comprises of a three–dimensional array of voxels. A typical volume in medical imaging can come from computed tomography (CT) scans.

volume and delineate them from other regions. Segmentation can be done using such methods as manual segmentation, thresholding, region growing, watershed segmentation and live-wire segmentation [28, pp. 95-104].

3. **Selection:** This step consists of limiting the number of rendered voxels. This is usually done by clipping or cropping the dataset.

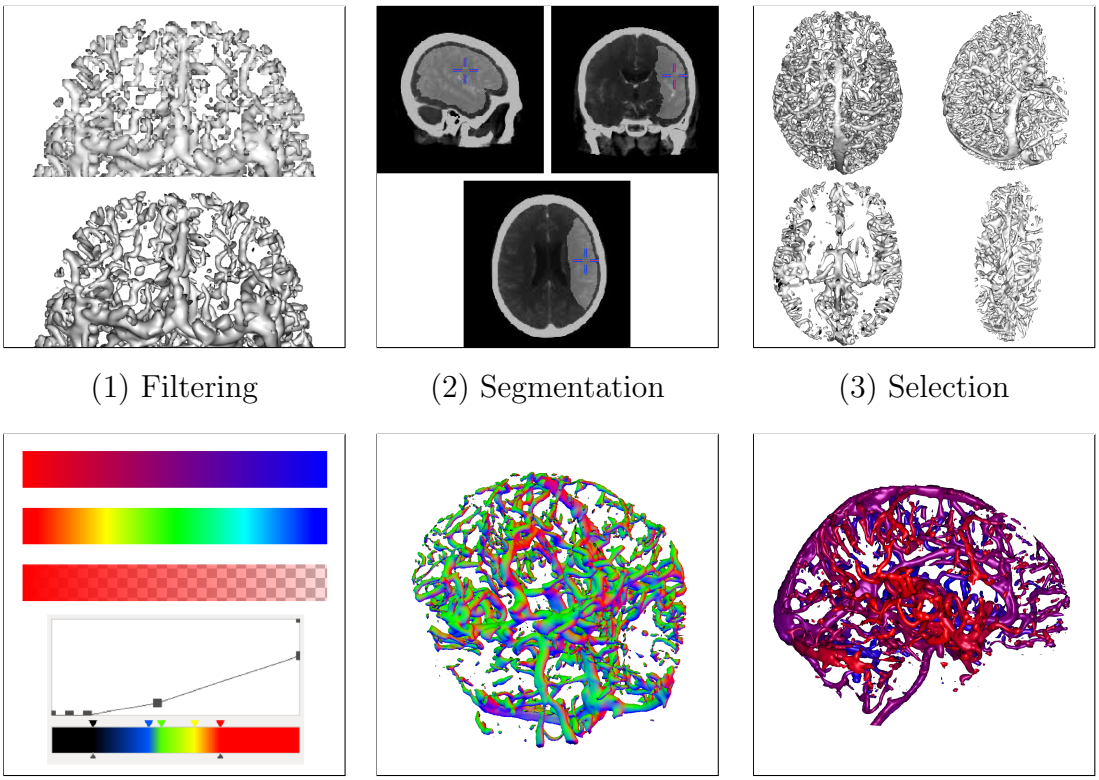
4. **Classification:** This step consists of defining a transfer function that will determine for any point in the dataset what the resulting color will be.

5. **Normal Computation:** The goal of this step is to calculate the normal vectors for the surfaces in the volume that are needed for lighting and shading (if used).

6. **Indirect or Direct Volume Rendering:** This step projects the 3D dataset into a 2D image and it shades the pixels on the final image depending on the transfer function and calculated normals.

## 2.1 Direct Volume Rendering Theory

With indirect volume rendering (IVR) the data is converted into a mesh using, for example, skeletonization [29] or a polygonal mesh extracting algorithm such as Marching Cubes [30]. In contrast, in direct volume rendering (DVR) rays sample the volume, and every sample of the volume is mapped to a color and an opacity using a transfer



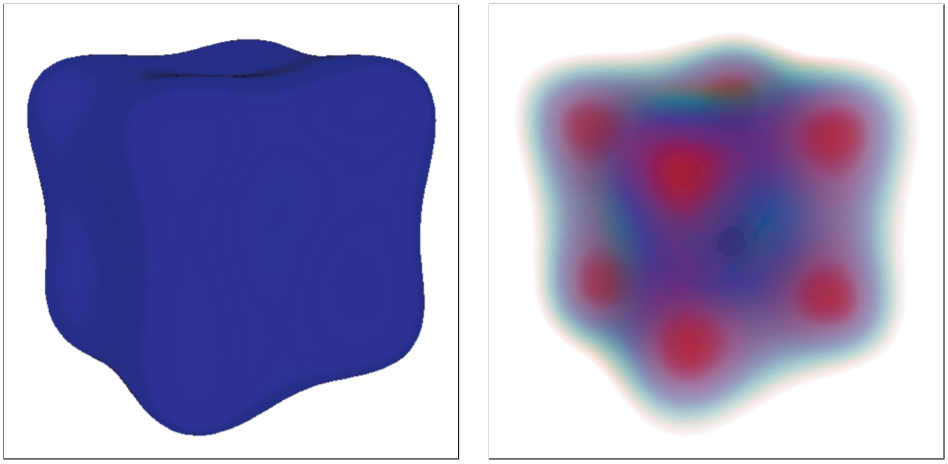
(1) Filtering (2) Segmentation (3) Selection

(4) Classification (5) Normal Computation (6) Indirect or Direct VR

Figure 4: Illustration of the steps of the volume rendering pipeline.

function. Figure 5 gives an example depicting the difference between surface rendering, which is an indirect volume rendering technique, and direct volume rendering.

In this chapter, we focus on direct volume rendering as it has many advantages over the indirect version [1] [2]. First, DVR doesn't require the data to be transformed into an intermediate form (such as a mesh), and thus during runtime the quality of the visualization is not limited by the quality of the intermediate data format. Second, DVR is much better at dealing with semi-transparency and coloring on a per-voxel basis, allowing significantly more control over how each pixel on the screen is shaded. Third, DVR allows more flexibility, making it possible to incrementally switch between different interesting features by modifying the transfer function at runtime. One disadvantage of DVR is that it is computationally expensive, however, this issue has been for the most part resolved due to faster GPUs (i.e. graphical processing units)[1].



(a) IVR (b) DVR

Figure 5: Comparison between (a) surface rendering and (b) direct volume rendering (b). Note that the image created in (a) could easily be implemented using DVR, while (b) would be much harder to implement using IVR. Figure adapted from [1] © [2008] IEEE

### 2.1.1 Optical Model

We perceive the world when light rays, interacting with the objects around us, enter the eye and are transformed into a 2D image on our retinas. Direct volume rendering algorithms are based on this same principle, they take into account the physical way in which light rays interact with a "participating medium" [31], i.e. an object that affects the transport of light through its volume. Using this principle, in 1995, Max [31], defined a volume as a particle-filled slab of width $\Delta s$, through which a light ray, $r$, is cast in the direction of the viewer (Figure 6). The light interacting with the participating medium can be absorbed, emitted and scattered [31] (see Figure 7).

#### Absorption

Particles in a participating medium absorb light rays $r$ travelling through it, resulting in a reduction of the light intensity. Where $s$ denotes different locations along the length of the ray passing through the participating medium, $\tau(s)$ is the probability per unit distance of the light being absorbed at location $s$ or simply the probability density. Thus, $\tau(s)$ is proportional to the particle density along the ray $r$ at $r(s)$.

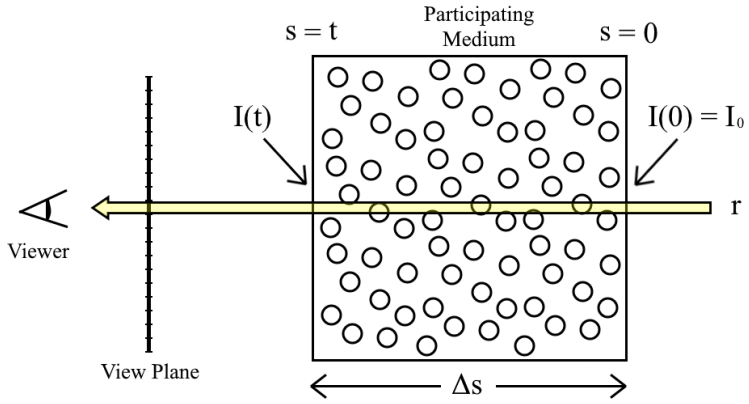
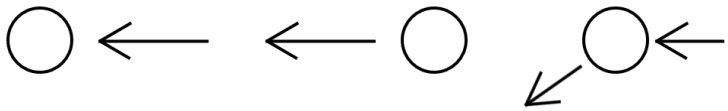



Figure 6: Ray $r$ traverses a participating medium of width $\Delta s$. $I(0)$ represents the intensity of the ray before it enters the participating medium, while $I(t)$ represents the intensity of the ray after it exits the participating medium.



(a) Absorption (b) Emission (c) Scattering

Figure 7: There are three types of interaction of a ray with the particles of a participating medium: (a) absorption, (b) emission, and (c) scattering.

Given the intensity of light, $I(s)$, when traversing a distance of $\Delta s$, the change of the intensity of the light is equal to $\Delta I$ and a fraction $\tau(s)\Delta s$ of $I(s)$ is absorbed over a distance $\Delta s$ such that:

$$\Delta I = -\tau(s)I(s)\Delta s \tag{1}$$

An expression for the light intensity leaving the volume at position $t$ considering the corresponding expression using infinitesimal quantities $dI$ and $ds$ in place of $\Delta I$ and $\Delta s$ is then calculated as:

$$dI = -\tau(s)I(s)ds \tag{2a}$$

$$\Rightarrow \frac{dI}{ds} = -\tau(s)I(s) \tag{2b}$$

To determine the cumulative intensity change resulting when the ray leaves the volume at position $t$, the following differential equation should be solved, giving:

$$I(t) = I_0 e^{-\int_0^t \tau(s)ds} \tag{3}$$

where $I_0$ is the intensity of the light ray before it enters the volume.

**Emission**

Emission is the opposite phenomenon to absorption. When light traverses a participating medium in which the particles diffusively emit light, the intensity of the light may increase [31]. Given $\tau(s)$ (the probability per unit distance of the light being absorbed at position $s$), let $C(s)$ represent the glow or intensity of the light per particle at position $s$. The amount of emitted light after traversing a distance of $\Delta s$ within the participating medium is:

$$\Delta I = \tau(s)C(s)\Delta s \tag{4}$$

The term $\tau(s)C(s)$ represents the source, which can be thought of as the color that is emitted at position $s$. For simplification, this term will be replaced by $g(s)$, giving:

$$\Delta I = g(s)\Delta s \tag{5}$$

When infinitesimal values for $\Delta I$ and $\Delta s$ are used, then they could be replaced by $dI$ and $ds$ giving the following formula:

$$dI = g(s)ds \tag{6a}$$

$$\Rightarrow \frac{dI}{ds} = g(s) \tag{6b}$$

After solving this differential equation, the intensity of the light after traversing a distance of $t$ within the participating medium becomes:

$$I(t) = I_0 + \int_0^t g(s)ds \tag{7}$$

**Scattering**

Particles can scatter light by deflecting incident light [31]. This can produce a blurring effect and may even change the color of surrounding objects, for example in the case of color bleeding. However, since the goal medical volume rendering is to convey information in a clear and unambiguous manner, scattering is not usually implemented, thus we do not describe it here.

### 2.1.2 Volume Rendering Integral

Taking into account only absorption (i.e. attenuation by the particles) and emission (i.e. the source term), we get the following differential equation [1]:

$$\frac{dI}{ds} = g(s) - \tau(s)I(s) \tag{8}$$

Solving this differential equation gives:

$$I(t) = I_0 e^{-\int_0^t \tau(u)du} + \int_0^t g(s) e^{-\int_s^t \tau(u)du} ds \tag{9}$$

The equation can then be approximated using a Riemann sum [31]. Assuming that the integral is separated into $n$ pieces of length $\Delta s = \frac{t}{n}$, then $e^{(-\int_0^t \tau(u)du)}$ it can be rewritten as:

$$e^{(-\sum_{i=1}^{n} \tau(i\Delta s)\Delta s)} = \prod_{i=1}^{n} e^{-\tau(i\Delta s)\Delta s} \tag{10}$$

Similarly, $\int_0^t g(s) e^{(-\int_s^t \tau(u)du)} ds$ can be rewritten as:

$$\sum_{i=1}^{n} g(i\Delta s) * \Delta s * e^{(-\int_{i\Delta s}^{t} \tau(u)du)} \tag{11a}$$

$$\approx \sum_{i=1}^{n} g(i\Delta s) * \Delta s * (\prod_{j=i+1}^{n} e^{-\tau(j\Delta s)\Delta s}) \tag{11b}$$

Thus, the discrete volume rendering integral is represented as:

$$I(t) \approx I_0 \prod_{i=1}^{n} e^{-\tau(i\Delta s)\Delta s} + \sum_{i=1}^{n} g(i\Delta s) * \Delta s * (\prod_{j=i+1}^{n} e^{-\tau(j\Delta s)\Delta s}) \tag{12}$$

[1] Equations derived from [31] by Max, N.

where $e^{-\tau(i\Delta s)\Delta s}$ represents the opacity of the $i^{th}$ segment and can be replaced by $\alpha_i$ for simplification. At the same time, $g(i\Delta s) * \Delta s$ represents the emission of light at the $i^{th}$ segment and can be replaced by $g_i$. Thus, the formula can be simplified to:

$$I(t) \approx I_0 \prod_{i=1}^{n} \alpha_i + \sum_{i=1}^{n} g_i \prod_{j=i+1}^{n} \alpha_j \tag{13}$$

This function gives a discrete estimate of the volume rendering equation. The source term $g_i$ indicates the color of the sample at position $r(i\Delta s)$, which is calculated by applying the transfer function on $r(s)$. The opacity term $\alpha_i$ is used when compositing multiple samples. Transfer functions and compositing are further described in 2.2.2.

## 2.2 Volume Rendering Implementations

Although a number of different direct volume rendering methods exist, including splatting [32], shear-warp [33], ray casting [34] and texture mapping [35], here, we focus on the latter two as these are currently the most commonly used in practice.

### 2.2.1 Ray Casting

Ray casting is a direct volume rendering algorithm where rays are sent from the viewer position through the volume [24], i.e. from $I(t)$ to $I(0)$. For each pixel of the view plane (the plane where the final image will be created) at least one ray is sent throughout the volume, which is represented in the form of a 3D grid. Samples are taken along the ray, $r$, at small steps, $\Delta s$, inside the volume (Figure 8). The steps are typically of a fixed size although some adaptive-step algorithms such as the one described by Zuiderveld *et al.* [25] may be used. As we can see in Figure 9, the step size will impact the quality of the rendered image, with larger steps leading to more artefacts, and conversely smaller calculation time and smaller steps lead to a higher image quality with less artefacts, but a higher calculation time.

At each step, the volume is sampled using some interpolation technique. Then, a transfer function is applied on the sample to create an RGBA color. All the calculated RGBA colors from all samples of the ray are then composited together until the ray

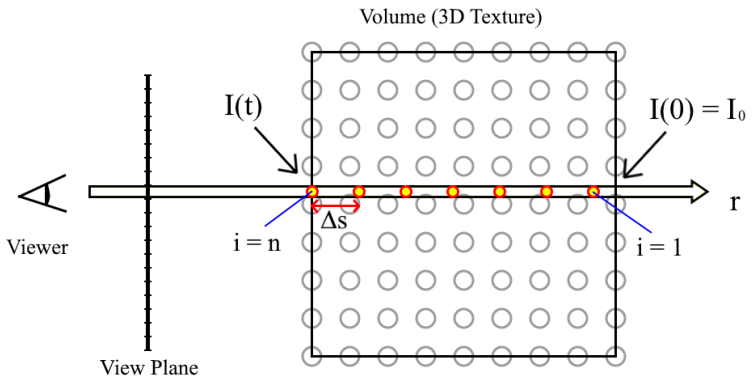
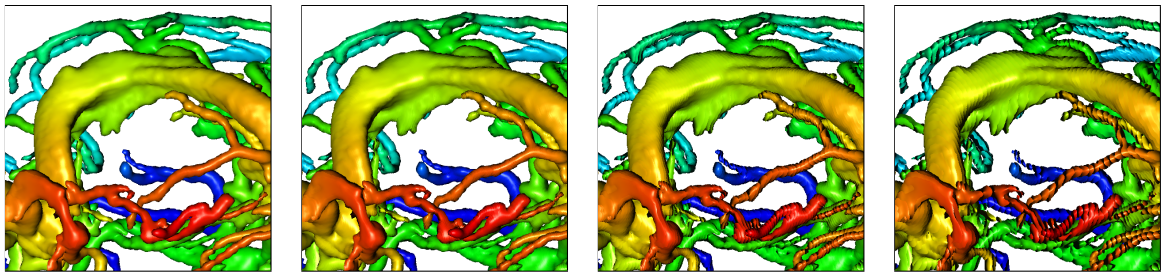



Figure 8: The volume is sampled along all rays, $r$, $n$ times, while performing a step of length $\Delta s$ between each sample.



(a) Step = 0.25 (b) Step = 0.50 (c) Step = 1.0 (d) Step = 1.5

Figure 9: Comparison of images rendered with different step sizes. These images were generated using the fixed-step algorithm. A $224 \times 212 \times 102$ texture was visualized using the chromadepth visualization cue.

exits the volume. In the next sections, we describe sampling, the commonly used interpolation methods, the transfer function and the compositing process.

### Sampling

To create the rendered image, the volume is sampled along all rays, $r$ and the accumulated color and opacity are computed based on Equation 13. To ensure smoothness, the values of the samples along a ray are interpolated. This is typically done using nearest-neighbor or trilinear interpolation although more complex algorithms such as triquadratic or tricubic interpolation may be used [28, pp.16-17].

In the case of nearest neighbor, the sampled value is equal to the value of the

closest voxel. In the case of trilinear interpolation [28, p.15], a linear interpolation is performed along all three axes ($x$, $y$ and $z$). Assume that there exists a 3D cell that represents the internal space between 8 voxels, creating a cube. Let $V_{000}, ..., V_{111}$ denote the values of these 8 voxels (Figure 10). Given a volume $f$, $f(x, y, z)$ is a $\mathbb{R}^3 \rightarrow \mathbb{R}$ function such that $x, y, z \in [0, 1]$ maps the the location of the sampling point at position $x, y, z$ inside the volume so that $f(0, 0, 0)$ samples the value of the voxel $V_{000}$ and $f(1, 1, 1)$ samples the value of the voxel $V_{111}$. Then for tri-linear interpolation, four linear interpolations are performed for the $X$ axis, giving four temporary variables:

$$\begin{aligned} t_{00} &= (1 - x) * V_{000} + x * V_{100} \\ t_{01} &= (1 - x) * V_{001} + x * V_{101} \\ t_{10} &= (1 - x) * V_{010} + x * V_{110} \\ t_{11} &= (1 - x) * V_{011} + x * V_{111} \end{aligned} \tag{14}$$

Secondly, two linear interpolations are performed for the $Y$ axis using the temporary variable calculated previously, giving two more temporary variables:

$$\begin{aligned} t_0 &= (1 - y) * t_{00} + y * t_{10} \\ t_1 &= (1 - y) * t_{01} + y * t_{11} \end{aligned} \tag{15}$$

Lastly, the interpolated value is calculated by performing a linear interpolation along the $Z$ axis:

$$f(x, y, z) = (1 - z) * t_0 + z * t_1 \tag{16}$$

An example of a volume sampled with nearest neighbor and trilinear interpolation is presented in Figure 11. As can be seen in the figure, trilinear interpolation results in a much smoother surface because it considers the values of the 8 surrounding voxels which are weighted according to their distance from the sampling point, unlike nearest neighbor which only considers the value of the closest voxel.

### 2.2.2 Transfer Function and Classification

After obtaining the sampled values, a transfer function, which maps the sampled data of the volume to optical properties, i.e. $RGB$ color and opacity ($A$ or $alpha$), is used

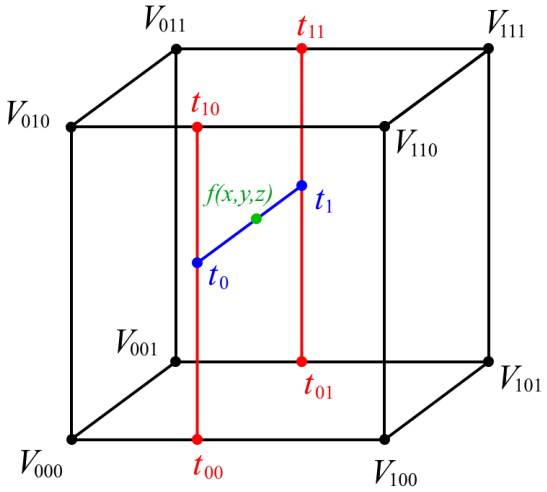
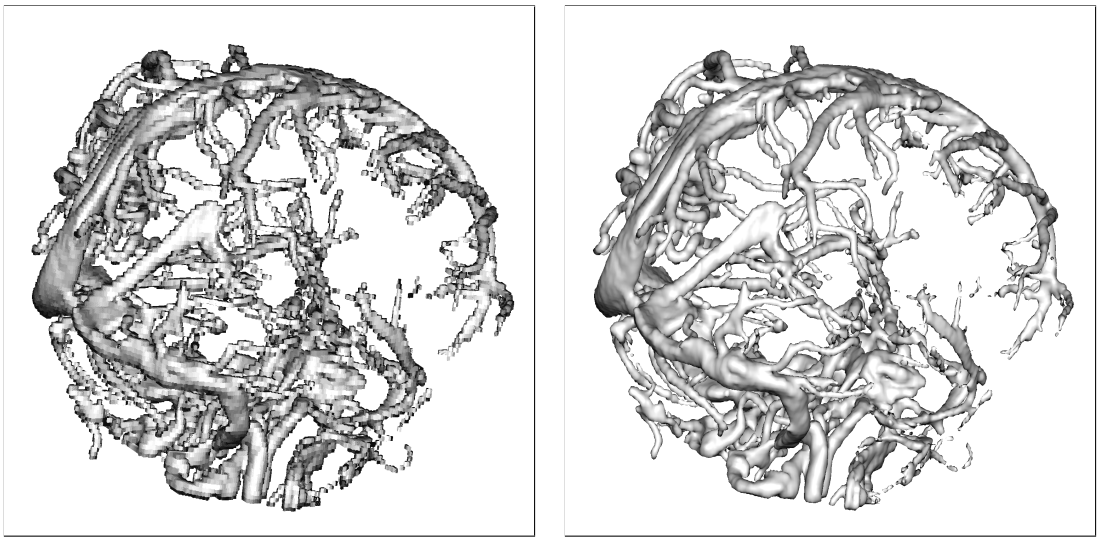



Figure 10: Trilinear Interpolation. The result of the interpolation is stored in $f(x, y, z)$.



(a) Nearest Neighbor (b) Trilinear Interpolation

Figure 11: A $224 \times 212 \times 102$ texture sampled using two different interpolation techniques. Blinn-Phong shading was used for both images.

[8]. A transfer function can take as an input, data related to the current ray position, such as the sampled intensity, the current depth of the ray or the gradient (i.e. the orientation of local surfaces within the volume). The samples can then be colored and illuminated based on the surface orientation and the lighting in the scene using

an illumination model such as Blinn-Phong [20], which will be described in Section 2.2.5.

The transfer function can be used to emphasize interesting features and to classify different parts of the volume [36]. Transfer functions are usually implemented using a texture lookup table that is sampled in the fragment shader, however some simpler transfer functions can be computed programmatically. In Figure 12, we show how the transfer function is used to accentuate different parts of a CTA volume.

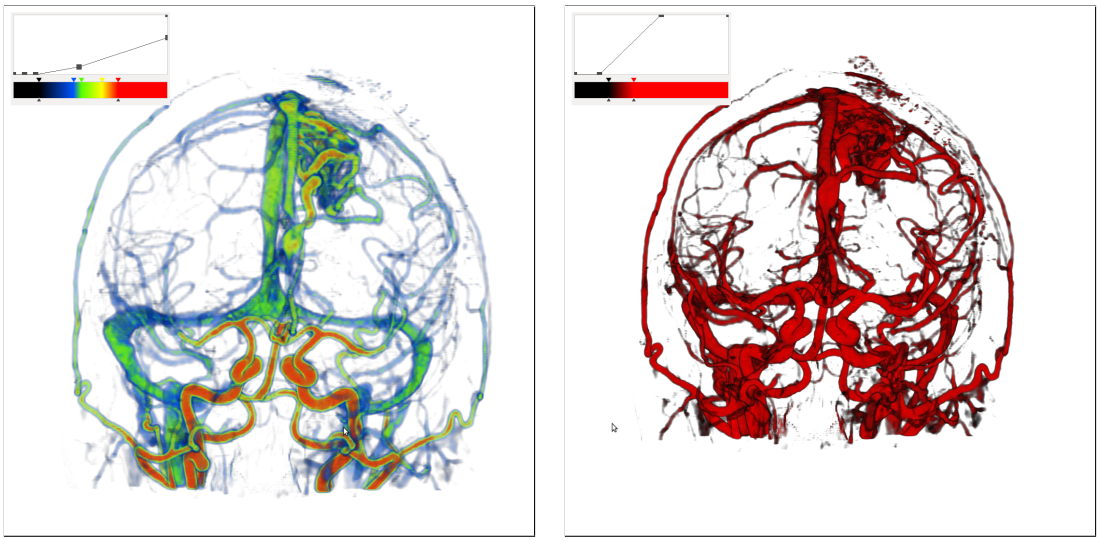


Figure 12: Examples of transfer functions created using the Intraoperative Brain Imaging System (http://ibisneuronav.org/). Transfer functions can emphasize different aspects of the volume. In the left image the color pertains to the flow of blood through the vessels, on the right edges are added using the transfer function.

### Compositing

Every sample along the ray has to be composited onto the accumulated color and opacity to give the final color value for the current pixel through which the ray is being sent [1]. This process is called compositing or alpha blending, and it is usually performed using the discrete volume rendering (Equation 13). Using this formula, the final color of the pixel can be calculated using a recursive function. Assume that the starting color of the ray before it enters the participating medium is $I_0$ and that the ray performs $n$ steps of equal size within the volume before exiting the participating

medium and reaching the view plane. Then, the accumulated color after $i$ steps is defined in the following way [28, p.190]:

$$I_i = I_{i-1} * \alpha_i + C_i, i \in \mathbb{N} \cap [1, n] \tag{17}$$

where $\alpha_i$ is the opacity of the volume at the $i^th$ step for the current ray.

Equation 17 is considered to be performing *back to front* composition since the ray enters the volume at the back of the volume and ends close to the view plane. Conversely, there also exists *front to back* composition that performs the calculations in the opposite order, starting from the end of the ray (close to the view plane). Assume that $I_n = C_n$ is the starting color and $o_n = \alpha_n$ is the starting opacity of the ray entering the volume. Then, the resulting color and opacity will be stored in $I_0$ and $o_0$ using the following formulas [28, p.190]:

$$\begin{aligned} I_{i-1} &= I_i * C_i + o_i, i \in \mathbb{N} \cap [0, n-1] \\ o_{i-1} &= \alpha_i * o_i, i \in \mathbb{N} \cap [0, n-1] \end{aligned} \tag{18}$$

A number of other compositing algorithms exist that emphasize or visualize different parts of the volume [28, p.190]. *First Hit* is a compositing algorithm that terminates the ray cast at the first sample that is above a certain fixed threshold, which produces a result similar to isosurface rendering. *Averaging* traverses the whole volume and calculates the average between all samples. *Maximum Intensity Projection* finds the sample with the biggest intensity and discards all the other samples.

### Alpha Blending

Alpha blending (also called alpha compositing) is used for compositing different samples in texture mapping or ray casting where multiple semi-transparent sampled values have to be blended together [1]. As described above, in ray casting alpha blending is used to composite the voxel colors from back to front when rays are cast through a given pixel. In texture mapping, alpha blending is used to composite the slices of the volume.

Specifically, alpha blending refers to combining a translucent foreground color with that of a background color [37]. The opacity of a certain surface or object is usually denoted by the letter $A$ or $\alpha$. Thus, the color of an object in computer

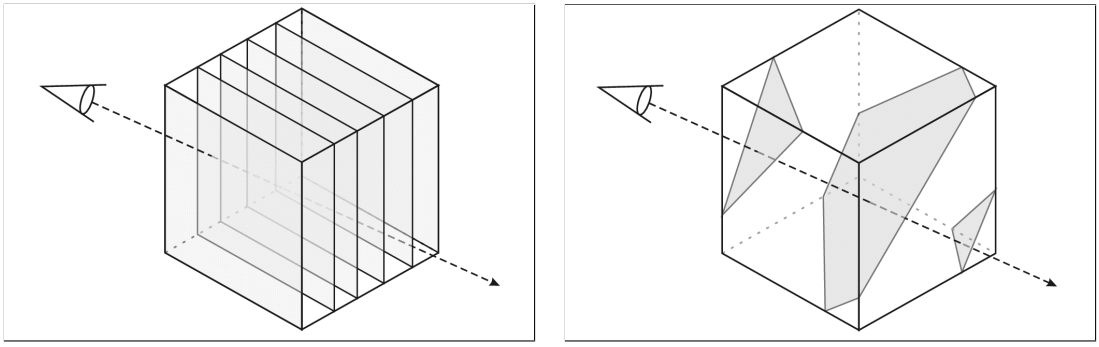
graphics is often stored as $RGBA$ where $RGB$ corresponds to the visible colors red, blue and green and $A$ indicates how opaque it is. Although $A$ does not represent any color, it comes into play when colored surfaces are blended together. The most widely used blending technique is called alpha blending. If $src$ represents the color in the foreground and $dst$ represent the color in the background, then the formula to alpha-blend these colors is the following[2]:

$$RGB_{out} = RGB_{src} * A_{src} + (1 - A_{src}) * RGB_{dst} \tag{19}$$

### 2.2.3 Texture Mapping

Volume rendering using ray casting is computationally expensive and depending on the size of the volume and other factors may not allow for real-time rendering. An alternative method is to use texture maps and exploit modern graphics hardware, i.e. graphical processing units (GPUs)[24]. Texture Mapping [38] is a volume rendering method that takes advantage of hardware acceleration. In this case, the volume is uploaded to GPU memory as a set of stacked 2D slices where it is interpreted as a 3D texture and can be sampled using the built-in trilinear interpolation function, and the slices can be blended together using hardware acceleration [39]. During rendering, slices of this texture are sampled at an equal distance between each other. In older hardware that didn't support 3D textures, the slices were object-aligned (Figure 13a), that is, they were stored three times aligned along the $x$, $y$ and $z$ axes, and the dataset whose slices were the most aligned with the view plane was used [35]. On newer hardware that supports 3D texturing, the slices are independent of the object space, and are calculated instead in image space [38]. The slices are positioned parallel to the view plane, i.e. the normal vector of the slices/planes points towards the viewer (Figure 13b). Each slice is computed by performing hardware-accelerated trilinear interpolation at the desired locations of the slicing plane, which map to pixel locations when projected on the view plane. It should be noted that the distance between different slices has a major impact on the quality of different images, with bigger distances resulting in a staircase effect [28, p.213].

[2]https://www.opengl.org/archives/resources/code/samples/advanced/advanced97/notes/node111.html (Accessed on October 13, 2020)



a) Object-Aligned Slices b) Viewer-Aligned Slices

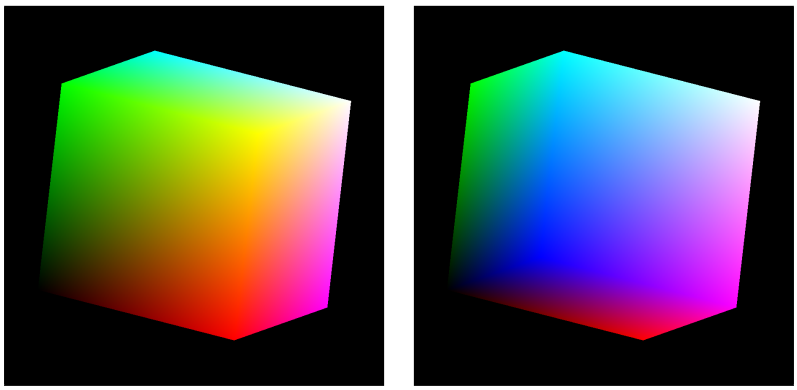
Figure 13: Comparison between object-aligned (a) and viewer-aligned (b) slices in texture mapping. Figure adapted from [40] © [2015] IEEE

All the sampled values from different slices are then combined together by performing compositing or alpha blending. Because of the nature of alpha blending, the slices are blended starting from the furthest from the viewer to the closest.

### 2.2.4 Two-Pass Volume rendering algorithm

One of the most popular ray casting algorithms is the two-pass algorithm described by Kruger *et al.* [24]. This method allows the dataset to be viewed from any angle, but unlike texture mapping, it doesn't require any resampling within the 3D texture. The algorithm consists of two passes, where the first determines the starting and ending position within the volume that the ray has to traverse and the second pass performs ray casting within the determined range.

During the first pass, a colored rectangular prism representing the bounding box of the volume is rendered twice to two different textures. The prism is colored so that the $RGB$ color with each component within the range $[0, 1]$ corresponds to a $(x, y, z)$ normalized coordinate in 3D space. One of the vertices is colored with solid black color, while the opposite vertex is colored in white (as demonstrated in Figure 14), marking the minimum and maximum $(x, y, z)$ values within the 3D space. Other vertices are colored with intermediate colors and the color on the faces of the prism between different voxels is calculated using bilinear interpolation. The prism is rendered first with only the front faces visible (Figure 14a), and then with only the back faces visible (Figure 14b).



a) Front Faces b) Back Faces

Figure 14: The colored cube used to determine the start and the end position of the ray for each pixel

During the second pass, a ray is cast for every pixel, performing standard ray casting (Equation 13). The two textures from the first pass are used to determine for every pixel the starting and ending positions where the ray should be cast. The ray performs small steps of constant size and samples the volume with trilinear interpolation at each step.

### 2.2.5 Blinn-Phong Reflection Model

Illumination models are used to improve and make the visual appearance of an object more realistic. The most popular illumination models are the Phong [41] and the Blinn [20] reflection model (also known as the Blinn-Phong reflection model [42, p. 257]). These illumination models work by approximating how light illuminates an object in the real world and how an object that is exposed to one or more light sources is perceived by the human eye[41].

Specifically, these models describe the way a surface reflects light as a combination of three independent light components: ambient, diffuse and specular [20], as demonstrated in Figure 15. These components are calculated for every pixel that displays a fragment of the illuminated object. Thus, the final color of these pixels is calculated as the sum of these three intermediate components. These models assume that it is possible to determine the normalized vector for any point on the surface of the volume. As the normal vector is not explicitly defined for every point, it has to

be calculated by linearly interpolating between the neighboring points for which the normal was calculated explicitly [41].

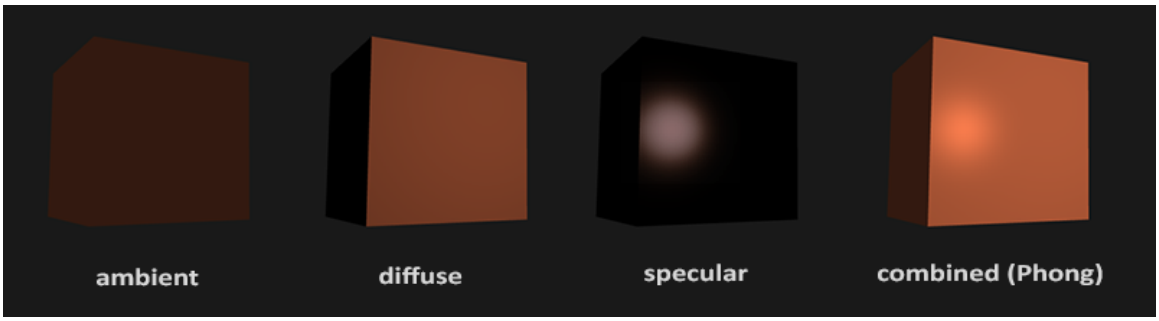



Figure 15: Components of Phong shading
Source: https://learnopengl.com/Lighting/Basic-Lighting by Joey de Vries (https://twitter.com/JoeyDeVriez), licensed under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/)

Although the Phong and Blinn-Phong illumination models are very similar, the Blinn illumination improves over the original Phong model [42, p. 257] in terms of the specular component (the ambient and diffuse components are the same). The details about the difference between these two models are discussed in Section 2.2.5.

### Ambient Component

One of the assumptions of the Phong or Blinn-Phong model is that in addition to the explicitly defined light sources that illuminate an object, there is also light present in the scene coming from indirect bounces of light [42, p. 295]. Thus, ambient light can be viewed as a very simplified version of global illumination; without it, any surface that is not directly illuminated by a source of light (i.e. is within a shadow) would be completely black. The intensity of ambient light, $I_a$, is defined as a constant such that it illuminates the surface of the object equally from all directions and the object reflects this light equally in all directions [20]. If only the ambient component is considered, the same point on an object would have the same color regardless of the viewpoint.

### Diffuse Component

The diffuse component represents the light that comes from a defined light source and that is reflected equally in all directions by the surface of the object [42, p. 106]. Unlike the ambient component, the diffuse light that the object reflects comes from a specific direction [42, p. 295]. For this component, the Blinn and Phong models assume a Lambertian surface[42, p. 240], i.e. a surface that reflects light equally in all directions. Thus only the direction of the light source and the normal vector of the surface affect the color of a point while the viewing angle has no impact [43].

The intensity of the diffuse component, $I_d$, is calculated as:

$$I_d = max(0, \hat{N} \cdot \hat{L}) \tag{20}$$

where $N$ is the interpolated normal vector of the surface and $L$ is the vector towards the light source.

### Specular Component

The specular component is used to represent the highlights that appear on the surface of the object, giving a "glossy" look to the object. This is unlike the diffuse component that gives a "matte" look. The specular highlights appear when the surface is located so that it reflects or mirrors the light source towards the viewer. For this component, the light direction, the normals of the surface and the viewer direction are taken into account.

As mentioned above, the specular component, $I_s$, is the only component that is calculated differently in the Phong and the Blinn-Phong models. The Phong model performs a dot product between the reflected light and the vector towards the viewer:

$$I_s = max(0, \hat{R} \cdot \hat{V})^s \tag{21}$$

where $R$ is the direction that a mirror-reflected ray from the light will point to and $V$ is the vector towards the viewer.

The following equation is used to calculate $R$, the normalized direction that a mirror-reflected ray from the light will point to:

$$R = 2(\hat{L} \cdot \hat{N}) \cdot \hat{N} - \hat{L} \tag{22}$$

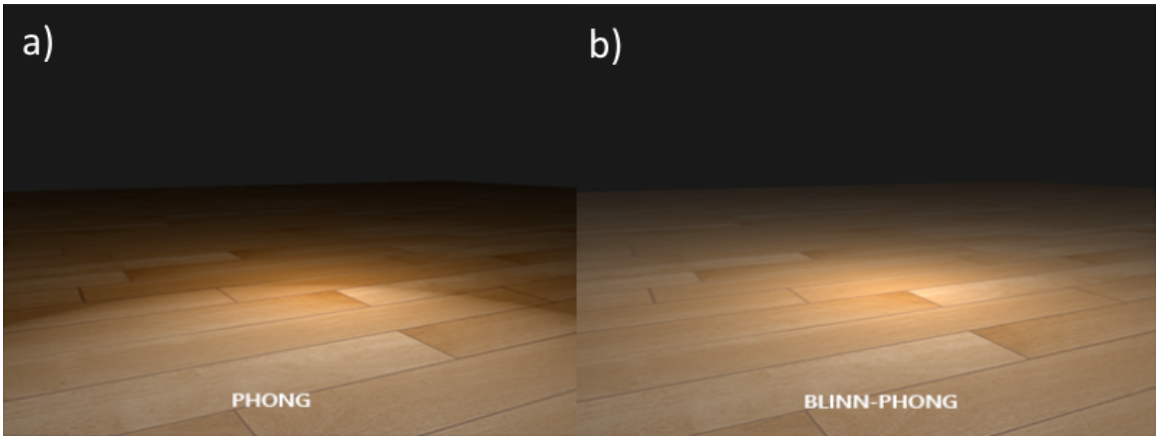



Figure 16: The cut-off problem in Phong shading and its fix in Blinn-Phong shading Source: adapted from https://learnopengl.com/Advanced-Lighting/Advanced-Lighting by Joey de Vries (https://twitter.com/JoeyDeVriez), licensed under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/)

where $N$ and $L$ are as defined above.

One drawback of the Phong model is that if the angle between $V$ and $L$ is less than 90 degrees, then the angle between $V$ and $R$ could become more than 90 degrees, resulting in an intensity of 0 for the specular component [42, p. 252]. This results in a very sudden color cut-off for the rendered object which is not physically accurate, as demonstrated in Figure 16 a). The Blinn-Phong reflection model uses the dot product between the normal $N$ of the object and the the halfway vector $H$ so that the angle between the $N$ and $H$ is never bigger than 90 degrees, thus removing the cut-off problem (see Figure 16 b)). The updated Blinn formula for the specular component, $I_s$, is as follows:

$$I_s = max(0, \hat{N} \cdot \hat{H})^s \tag{23}$$

As mentioned above, $H$, the halfway vector, represents the average vector between the vector towards the light and the vector towards the viewer. It can be calculated for any point on the surface of the object using the following formula:

$$H = \frac{\hat{L} + \hat{V}}{\|\hat{L} + \hat{V}\|} \tag{24}$$

### Shading Function

Given, the three lighting components, the final color of any given pixel, $C$, is calculated by adding the ambient, $I_a$, diffuse, $I_d$, and specular, $I_s$, components. Blinn [20] described the formula for the final color (with the assumption that the light is white) as:

$$C = p_a I_a c + p_d I_d c + p_s I_s \tag{25}$$

where $p_a$ is the proportion of the ambient reflection, $p_d$ is the proportion of the diffuse reflection, $p_s$ is the proportion of the specular reflection and $c$ is the color of the object.

## 2.3 Limitations

Understanding the spatial layout or depth between different elements in a complex 3D volume may be difficult depending on the rendering technique used and if the volume is rendered on a 2D display. In the next chapter, we explore different volume visualization techniques to improve spatial and depth understanding of medical data, and specifically angiography data. Further, we describe a game that was developed to evaluate the effectiveness of these techniques.

# Chapter 3

# Connect Brain: A mobile game to study the efficacy of vascular volume visualization techniques

A version of this chapter was submitted to IEEE Transactions on Visualization and Computer Graphics:

> Titov A., Kersten-Oertel, M. Connect Brain: A mobile game to study the efficacy of vascular volume visualization techniques. Submitted to *IEEE Transactions on Visualization and Computer Graphics*.

## 3.1 Introduction

In the field of medical imaging, angiography is used to visualize vascular structures inside the body. This is typically done by injecting a contrast substance into a patient and imaging the patient with X-ray, Magnetic Resonance (MR), Computed Tomography (CT) [5]. For 3D X-ray, MR or CT angiography, the result is a 3D volumetric representation of the scanned patient's vascular anatomy. This 3D volume can be visualized using such methods as volume rendering, maximum intensity projection (MIP) or surface rendering [7].

Cerebral angiography, specifically, depicts the blood vessels inside the brain. The goal of this type of angiography is to help radiologists and surgeons understand the

cerebral vasculature and detect abnormalities such as stenosis, arteriovenous malformations (AVMs) and aneurisms [4]. However, visualizing angiography data in a manner so that it is well understood spatially presents certain challenges [4, 5, 6]. First, cerebral vasculature is very complex, with intricate branching and many overlapping vessels, hindering the understanding of the data in 3D [5]. Second, due to variations in anatomy, from patient to patient, surgeons may not always be able to rely on past experience to understand a new dataset [5]. Third, depending on the environment (e.g. the operating room), not all visualization methods can be used to render the data. For example, 3D viewing requires specialized equipment (e.g. stereoscopic display or augmented reality glasses), which are not always available. Perspective rendering, may also be inconvenient to use when displaying the data as surgeons may want to perform measurements on the angiographic image, so most commonly orthographic projection is used for medical image visualization.

To improve depth perception and spatial understanding of vascular volumes, numerous perceptually-driven vessel visualization methods have been developed [7, 4, 9, 10, 11, 12]. In addition, shading techniques that are not specific to medical visualization [41, 20, 21, 22] have also been tested. In all of these works, user studies to determine the effectiveness of the different visualization techniques were performed in a laboratory environment under the supervision of a researcher [13]. This type of laboratory study has a number of disadvantages: the lack of diversity between the participants (which are often young college students) [13] and a limited pool of participants or conversely a high monetary cost for studies which have many participants [14].

To address these issues, we used the gamification paradigm to collect data on the effectiveness of different perceptually-driven vascular volume visualization techniques. Building on previous work [5, 8, 4], we developed a mobile app, Connect Brain [1] [2], with two different games (distributed on the App Store and on Google Play). The purpose of this app, was not only to determine the effectiveness of differing vessel visualization techniques and general purpose shading techniques in understanding

[1] Google Play link:
https://play.google.com/store/apps/details?id=ca.andreytitov.connectbrain
(Published on May 2020)

[2] App Store link:
https://apps.apple.com/us/app/id1524359191
(Published on Aug 2020)

the spatial and 3D layout of cerebral vasculature volumes but also to see how well the results of small laboratory studies generalize to a larger population.

### 3.1.1 Gamification

Gamification is similar to crowdsourcing and shares its advantages [13]. Crowdsourcing is a method of conducting user studies which distributes a given task to a larger network of participants [13]. One example of a platform for crowdsourcing is the Amazon Mechanical Turk (AMT) [23], which has been used in studies for a variety of topics such as perceptual effectiveness of line drawings to depict shapes [44], natural language processing [45] and audio transcription [46]. Crowdsourcing enables a larger study population in comparison to traditional studies because the task can be distributed online. In addition to this, the subject pool becomes more diverse since the study is no longer limited to a physical environment (e.g. a university lab). Finally, crowdsourcing is less time consuming for each individual subject and allows a lower per-participant cost [23]. This model also has some disadvantages, the main one being a lower data quality because researchers don't have as much control over the unfolding of the experiment and because participants may only be motivated by the monetary gain [14, 13].

The main difference between gamification and crowdsourcing is that gaming elements are added to the study [13]. With gamification a study is transformed into a game that is fun to play, and the gameplay data is collected and analyzed. The most important advantage of gamification is that users are motivated to perform well, which consequently may increase the quality of the collected data compared to crowdsourcing. Further, players are motivated to perform well not because of monetary incentives, but because they enjoy playing the game [14]. As gamification scales well with a high number of participants (since players download and play the games on their own devices) these types of studies have an even lower runtime cost than crowdsourcing [14]. However, there are several disadvantages, firstly not every study can be transformed into a game that is fun to play. Furthermore, developing and publishing a game requires more time and effort, above and beyond, creating an experimental study. Lastly, for success, the researcher should develop interesting game mechanics that follow the rules of game design [14].

## 3.2 Implemented Visualizations

In this section, we describe the specific vascular volume visualization techniques (Figure 17) that were implemented in our game, Connect Brain.

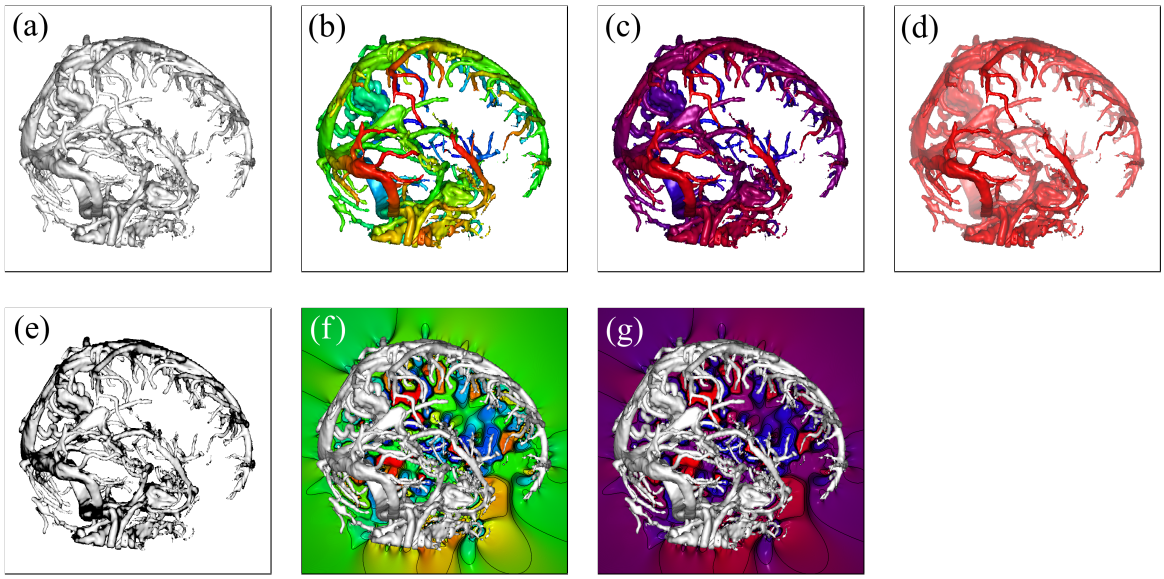



Figure 17: All implemented vessel visualization techniques: (a) Shading (Blinn-Phong), (b) Chromadepth, (c) Pseudo Chromadepth, (d) Aerial Perspective, (e) Edge Enhancement, (f) VSS Chromadepth, (g) VSS Pseudo Chromadepth

### 3.2.1 Edge Enhancement

Edge enhancement has been used to emphasize the occlusion depth cue, where a viewer determines the relative depth between different objects due to the way they overlap [47]. In vessel visualization, the contours of vessels are emphasized, typically by rendering dark lines around the edges of the vessels [22] (see Figure 17(e)). This cue is especially helpful when the transfer function produces a translucent result. In this case, the highly-contrasted black silhouettes occlude the silhouettes of the vessels that are further away from the viewer, thus giving a better understanding of the depth ordering of vessels.

Following, the work of Drouin *et al.* [47], in our implementation edge enhancement was combined with Blinn-Phong shading. To do this, the volume is rendered using Blinn-Phong shading and the pixels that form the silhouette are darkened based on

the interpolated normal vector for each pixel. Pixels with a normal that is almost perpendicular to the viewer are considered as being part of the silhouette. Drouin *et al.* [47] described the formula to calculate the intensity of the edge enhancement for a given pixel as:

$$\alpha = smoothstep(stepMin,\ stepMax,\ \|\vec{N}\| \cdot (1 - |\vec{N} \cdot \vec{V}|))$$

where $\alpha$ is the intensity of the edge enhancement factor, $\vec{N}$ is the gradient (normal vector) of the surface, $\vec{V}$ is the direction of the ray (from the volume towards the viewer) and $stepMin$ and $stepMax$ are user defined parameters.

### 3.2.2 Aerial Perspective

Aerial perspective is a monocular depth cue caused by the atmosphere and the way in which light scatters. Specifically, the further the distance between an object and a viewer, the less contrast there is between that object and the background. With this technique, the vessels that are closer to the viewer appear as more saturated and more contrasted while further vessels fade into the background [5, 6] (see Figure 17(d)). By comparing the saturation of two vessels, it is possible to deduce which one is closer and which one is further away.

To render a dataset with aerial perspective cue, the pixels representing the color should be correctly blended with the background. Ebert *et al.* [48] described the formula for distance color blending:

$$C = (1 - d) \cdot c_o + d \cdot c_b$$

where $d$ is the depth of the volume at the current pixel in the range of $[0, 1]$, $c_o$ is the color of the object and $c_b$ is the color of the background. Preim *et al.* [6] noted, the relationship between depth of the projected vessel and the saturation of the pixel does not need to be linear but can rather be exponential (by replacing $d$ with an exponential function). In order to ensure, the visualization of the entire volume (such that no vessels are blended completely into the background), Kersten *et al.* [19] determined that the best upper bound for $d$ was between 0.75 and 0.85. In our implementation, we used the original linear formulation with values $d = 0.8$.

### 3.2.3 Chromadepth

Chromadepth, a technique developed by Richard Steenblik [49], encodes depth using colour. Specifically, the colour of the pixels in depth follows the colors of the visible light spectrum, starting from red, orange, yellow, green, cyan, to blue [21]. Thus for a vascular volume, the closest vessels are red, the furthest ones blue and the ones in between have a color that is linearly interpolated between these values (see Figure 17(b)).

Bailey *et al.* [21] described the chromadepth transfer function. Given, a one-dimensional texture containing all the colors (from red to blue), $s$ is defined as the sampling parameter. Where $D_1$ and $D_2$ are parameters defined by the viewer such that $D_1 \geq 0$, $D_2 \leq 1$ and $D_1 < D_2$, then, for any depth $d$ such that $d \in [0, 1]$, the transfer function is defined as:

if $d < D_1$, then the color of the pixel is red

if $d > D_2$, then the color of the pixel is blue

else, the parameter $s$ is calculated such that $s = \frac{d}{D_2 - D_1} - \frac{D_1}{D_2 - D_1}$.

### 3.2.4 Pseudo Chromadepth

Ropinski *et al.* [4] noted that one of drawbacks of chromadepth is that the large number of hues presented in a chromadepth image can distract from the understanding of the depth. To alleviate this issue, pseudo chromadepth, which uses only two colors (red and blue) instead of the full color spectrum can be used [4] (see Figure 17(c)). Red and blue colors are used because of the visual phenomena of chromostereopsis, which is caused by light of different colors refracting into different parts of the retina in eye depending on the wavelength [50]. Chromostereopsis can be used to make red objects appear closer in depth then blue ones.

When using pseudo-chromadepth for vasculature, the closest vessels are red, the farthest are blue, and for any intermediate depth, the color of the pixel is calculated by interpolating between red and blue. Thus, using the pseudo-chromadepth depth cue, a depth comparison between two shaded objects can be simplified to a simple comparison of the hue, with warmer hues representing closer objects and colder hues representing further ones.

The pseudo-chromadepth cue was implemented in the same way as chromadepth, with the only difference being that the 1D rainbow-like texture was replaced by one where the color is linearly interpolated between red and blue colors.

### 3.2.5 Void Space Surface

Void Space Surface (VSS), a technique used in vessel visualization, was developed by Kreiser *et al.* [12] (see Figure 17(f,g)). Unlike many other vessel visualization techniques that are based on shading the vessels in a certain manner, VSS concentrates on shading the area around the vessels; the background is coloured to indicate the relative depth of surrounding vessels. Therefore, to understand the relative depth of a certain vessel, one looks at the color of the background that surrounds this vessel. The motivation behind VSS is that in the more traditional depth rendering methods there is a lot of unused empty space. Therefore, instead of being limited by the area that vessels take on the screen, the entirety of the screen can be used, allowing the vessel pixels to represent any other information that may be deemed necessary.

To determine the color of each pixel, a weighted average of the depths of the surrounding border pixels are calculated. In order to do this, a rendered image of a vessel structure in the form of a depth map on which the filled pixels (representing the volume) can be distinguished from the empty ones (representing the background) is needed. The Suzuki [51] border following algorithm is then executed on the depth map, creating a hierarchy of borders of the depth map. This hierarchy indicates what border pixels contribute to what part of the background. After this is done, the interpolated depth for each background pixel is calculated using Inverse Distance Weighting [52]:

$$Depth = \frac{\sum_{i=1}^{N} w(p_i) \cdot d(p_i)}{\sum_{i=1}^{N} w(p_i)}$$

where $D_b$ is the calculated depth of the background pixel, $p_i$ is the i-th border pixel whose depth is used in the weighted average calculation, $N$ is the total number of border pixels which affect the depth of $p_b$, $w(p_i)$ is the weight of the border pixel $p_i$ and $d(p_i)$ is the depth of the border pixel $p_i$.

The weight $w(p_i)$ of a border pixel $p_i$ is calculated in the following way:

$$w(p_i) = \frac{1}{m(p_b, p_i)^s}$$

where $p_b$ is the background pixel for which the depth calculation is performed, $p_i$ is the i-th border pixel whose depth is used in the weighted average calculation, $m(p_b, p_i)$ is the magnitude of the vector between the position of the pixel $p_b$ and $p_i$, and $s$ is a user-defined smoothing parameter that results in closer border pixels giving exponentially more weight.

After calculating the depth of every background pixel, a transfer function is then applied to the depths transforming them into a color. Usually, chromadepth (see Figure 17(f)) and pseudo-chromadepth (see Figure 17(g)) are used [12]. In addition to this, VSS implements an approximated version of global illumination in the form of Screen Space Directional Occlusion (SSDO) [53]. SSDO darkens some regions of the generated VSS that may be occluded from the light by neighboring parts of the VSS and performs an indirect light bounce. Finally, isolines are generated on the surface of the VSS in the form of black lines to improve understanding of the generated shape by the VSS.

In our case, because of the hardware limitations of mobile devices, we used Screen Space Ambient Occlusion (SSAO) instead of SSDO, which doesn't include the indirect bounce.

## 3.3 Related Work

Multiple user studies have been done to compare the effectiveness of different perceptually-driven and illustrative vessel visualization techniques [6]. An overview of the most related studies and their results is presented in Table 1.

Ropinski *et al.* [4] performed a user study where 10 subjects were to determine, between two indicated points on the vasculature of a liver, which was closer to them. A number of different visualization techniques were explored: standard rendering, stereoscopy, chromadepth, pseudo chromadepth, depth of field, a combination of depth of field and pseudo chromadepth, as well as multiple techniques related to edges (overlaid, blended, perspective edges and edge shading). The results of the study showed that overall pseudo chromadepth was the most effective cue, although it was tied with the depth of field cue in terms of accuracy ($M = 93\%$), it generally had the

| **Reference** | **Visualizations** | **Subjects** | **Trials / Sample Points** | **Goal** | **Result** |
|---|---|---|---|---|---|
| Ropinski *et al.* [4] | Phong, Stereo, Chroma, Pseudo Chroma, Overlaid, Blended, & Perspective Edges, Edge Shading, DoF, DoF + Pseudo Chroma | 14 | 50 x 14 = 700 | Depth Comparison<br><br>**Metrics:** Correctness, Time, and User feedback | • Pseudo chroma better than chroma and best overall<br>• Edge best in terms of %correct<br>• Stereo least effective overall<br>**User feedback:**<br>• Pseudo Chroma got highest score<br>• DoF got the lowest score |
| Kersten *et al.* [5] | No cue, Kinetic, Stereo, Edge, Pseudo Chroma, Fog + combined cues (for novice experiments only) | 2 studies: 13 novice & 6 experts | 160x13= 2080 (Novice)<br>6 x 50 = 300 (Expert) | Depth Comparison<br><br>**Metrics:** Correctness, Time, and User feedback | **Single cues:**<br>• Pseudo Chroma best overall<br>• Edge cue more helpful to experts<br>**Combined cues:**<br>• Best cues overall are Pseudo Chroma + Stereo and Pseudo Chroma + Stereo + Edge.<br>**All cues:**<br>• Large $xy$ and large $z$ (depth) difference between points increased %correct<br>**User feedback:**<br>• Novices: Pseudo Chroma ranked best<br>• Experts: Fog ranked best |
| Kreiser *et al.* [12] | Phong, Chroma, Pseudo Chroma, VSS Chroma, VSS Pseudo Chroma | 19 | 150 x 19 = 2850 | Depth Comparison<br><br>**Metrics:** Correctness, Time | • VSS and Direct: higher %correct than Phong<br>• VSS and direct have similar %correct.<br>• In terms of time, Direct better than Phong, but VSS worse than Phong<br>• Pseudo Chroma generally not better than Chroma. |
| Abhari *et al.* [8] | Shading, Edge | 10 | 60 x 10 = 600 | Connectivity<br>**Metrics:** Correctness, Time and Expert Feedback | • Edge generally better than shading<br>• Response time directly proportional to the $xy$ distance between points<br>**Expert feedback:**<br>• Shading better at allowing to understand lesions<br>• Edge better at allowing to understand vessel continuity |
| Drouin *et al.* [11] | Shading, Pseudo Chroma, Fog, Dynamic Shading, Dynamic Pseudo Chroma, Dynamic Fog | 20 | 80 x 20 = 1600 | Target-ing/Reaching<br>**Metrics:** Correctness, Time, Pointer-Target Distance and User Feedback | • Dynamic cues helped understanding local structures, but not global ones<br>• Dynamic pseudo chroma and dynamic Fog higher decision time than static version<br>**User feedback:**<br>• Pseudo chroma preferred cue for both static and dynamic rendering<br>• Subjects said that dynamic cues allowed to perform better in terms of time, not confirmed by experiment<br>• High rating for dynamic shading despite poor results<br>• Lower rating for dynamic fog despite better results than with dynamic shading |

Table 1: Table with the related works on volume rendering vascular visualization techniques. Short forms used in table: Stereo = Stereoscopy, Chroma = Chromadepth, Pseudo Chroma = Pseudo Chromadepth, DoF = Depth of Field, Overlaid = Overlaid Edges, Blended = Blended Edges, Kinetic = Kinetic Depth, Edge = Edge Enhancement, VSS = Void Space Surface, Direct = Directly Applied Cues, Base = Baseline

fastest decision time ($M = 2.1s$). The authors also found that stereoscopic rendering with an autostereoscopic display was the least effective cue with the highest decision time ($M = 3.9s$) and the worst accuracy ($M = 72\%$). Additionally, a was survey conducted after the study asking subjects to rate the perceived effectiveness of each depth cue using a Likert scale. Subjects gave the highest rating to pseudo chromadepth and the lowest one to depth of field cue, despite good results in terms of accuracy in the experiment. The authors posited that the low score could be caused by subjects feeling the depth of field images were out of focus.

Kersten *et al.* [5] performed a similar user study with a slightly larger pool of participants: 13 novices and 6 experts (e.g. neurosurgeons). The subjects performed an analogous experiment to that described by Ropinski *et al.* [4], looking at the following depth cues: no cue, kinetic depth, edges, pseudo chromadepth, fog, and some combination of the cues (only in the case of novices). The results showed that pseudo chromadepth and fog were some of the best in terms of both time and the correctness, for the novices ($M_{PChroma-\%Correct} = 90.4\%$, $M_{PChroma-Time} = 4.83s$, $M_{Fog-\%Correct} = 78.3\%$, $M_{Fog-Time} = 4.76s$) and experts ($M_{PChroma-\%Correct} = 88.3\%$, $M_{PChroma-Time} = 7.36s$, $M_{Fog-\%Correct} = 85.0\%$, $M_{Fog-Time} = 5.96s$) alike. However, for these cues the task of determining depth is a simple hue and/or color saturation comparison, thus it does not necessarily reflect that subjects spatially understood the 3D volume. The stereoscopic ($M_{\%Correct-Novices} = 64.8\%$, $M_{Time-Novices} = 6.50s$) and the kinetic depth ($M_{\%Correct-Novices} = 44.3\%$, $M_{Time-Novices} = 6.53s$) cues yielded poor results, with low accuracy and the highest decision times. For the combined cues none performed significantly better than the single cues. The authors also looked at the relation between the distance of the selected points and found that having a larger $z$ difference (depth) and a larger $xy$ difference (pixel position) increased accuracy. Finally, experts took more time to make a decision than novices. In terms of subjective results, novices rated pseudo chromadepth and kinetic depth received a relatively high rating too despite its poor performance. Among the experts, the aerial perspective cue received the highest rating.

Kreiser *et al.* [12] introduced the Void Space Surface (VSS) cue and performed a study comparing it with other vessel visualization techniques. Twenty subjects participated in the study, but one was excluded because of high error rates. The experimental task was the same as the one used in Ropinski *et al.* [4] and Kersten *et*

*al.* study [5]. The authors compared VSS to cues applied directly to the volume and found that VSS achieved a similar accuracy ($M_{VSS-CD} = 92\%$, $M_{VSS-PCD} = 89\%$) to directly applied cues ($M_{CD} = 91\%$, $M_{PCD} = 95\%$), and both performed better than the standard shading ($M = 73\%$) in terms of accuracy. However, in terms of response time, while directly applied depth cues performed better than shading, VSS performed worse. The authors stated that this may be caused by the indirect nature of the VSS cue. Unlike the study performed by Ropinski *et al.* [4], Kreiser *et al.* [12] could not confirm that pseudo chromadepth is more effective than full-color chromadepth. It was also noted that isolines were very helpful when the difference between the points was small and the points shared the same void space surface.

Abhari *et al.* [8] performed a study where the goal was to determine the connectivity between two vessels rather than their depth relationship. Specifically, the participants were presented with volume-rendered images on which two points were selected and the subject had to determine if there exists a path using the visible vessels that connects the two points. The reasoning behind this type of task is that in the context of surgery or preoperative planning, clinicians may want to trace the blood vessels inside of a medical dataset to better understand it. In the study, the visualizations were limited only to shading (non-enhanced visualization) and edge enhancement. It was determined that edge enhancement improves accuracy ($M_{Shading} = 75.60\%$, $M_{EE} = 84.00\%$) and response time ($M_{Shading} = 13.4s$, $M_{EE} = 11.17s$). The results also showed that the response time is directly proportional to the $xy$ distance between the selected points. In addition to the user study, a neurosurgeon provided feedback on the effectiveness of the depth cues when conducting pre-operative planning using the provided cues. The surgeon found that shading was better for understanding the lesions in the presented vessel data, while edge enhancement was better for understanding the continuity of the vessels. Further, stereoscopic rendering was better in both cases than monoscopic.

Drouin *et al.* [11] introduced dynamic rendering methods in which the viewer could use a pointer (instead of a keyboard or a mouse) to interact with the visualized vascular data. For the chromadepth and the aerial perspective cues, subjects could changing the location of the planes between which the color or the transparency is interpolated. For the Blinn-Phong shading [20], subjects could modify the location of the point light that illuminates the object, which dynamically changed the shading

of the volume. A study with 20 subjects was done to compare the effectiveness of the static and dynamic versions of pseudo chromadepth, aerial perspective and Blinn-Phong shading. The task of the user was to move a handheld pointer device so that its tip is as close as possible to one of the two select points that looks closer to them in terms of depth. It was determined that generally, both Chroma and Fog result in a significantly better accuracy (in terms of mm length to the target 3D point) than Shading (chroma was $5.8mm$ closer than shading, fog was $6.6mm$ closer than shading), but there was no significant difference between the cues in terms of correctness. When it comes to the comparison between static and dynamic version of the cues, dynamic cues allowed for better understanding of the local structures of the volume, but they didn't help or hinder the understanding of the global structure. In terms of the time taken, dynamic chromadepth and aerial perspective took more time compared to their static version, while for shading there was no difference. Subjective ratings showed that pseudo chromadepth was favored under both the static and the dynamic rendering condition. Subjects also felt that dynamic cues allowed them to be faster and more accurate, but this was not confirmed by the quantitative measures.

Although we did not consider any illustrative techniques in our study, a number of groups have looked at how these methods can be used for visualizing vascular data. These types of techniques included, hatching, shadows, and anchors. Ritter *et al.* [7] proposed a vessel visualization technique that consists of drawing hatching strokes and shadows on the volume. This method conveys the shape of the vessels as well as the connectivity between them; hatching strokes indicate how fast the depth is changing on the surface of the vessel structure, while the thickness of shadows indicates the relative depth between vessels. In a study on the effectiveness of these two cues for depth perception the authors found that the distance-encoded shadows allowed for a significant decrease in the decision time compared to Gouraud shading, while hatching didn't show any improvement upon Gouraud shading.

Lawonn *et al.* [9] proposed three depth cues for better understanding of vessel structures: illustrative shadows, supporting lines and depth-dependent contours. In a study, where the combination of these three techniques was compared to pseudo-chromadepth and Phong shading, the authors found that their illustrative technique resulted in a higher accuracy and higher confidence, with pseudo chromadepth having the second best result for both criteria. However, in terms of decision time, pseudo

chromadepth performed the best ($M = 10.95s$), followed by Phong shading ($M = 13.21s$) and then the illustrative technique ($M = 14.04s$). In another study, Lawonn *et al.* [10] presented another illustrative technique in the form of anchors that indicate the relative depth of some visible vessels on the screen. In this study, the results were very similar to the illustrative shadows, supporting lines and depth-dependent contours technique. The illustrative technique resulted in the best accuracy ($M = 0.84\%$) and best confidence ($M = 3.85/5$), followed by pseudo chromadepth ($M = 0.54\%$ for accuracy, $M = 2.60/5$ for confidence) and then by Phong ($M = 0.26\%$ for accuracy, $M = 2.78/5$ for confidence). For the decision time, pseudo chromadepth resulted in the fastest answers ($M = 9.38s$), followed by Phong ($M = 10.65s$) and then the illustrative technique ($M = 11.39s$).

## 3.4 Methodology

Connect Brain was developed using the Unity engine [3] for the Android and iOS platforms and published on Google Play and in the App Store in Aug 2020. The visualizations described in Section 3.3, in addition to Blinn-Phong [20] which was used as the base case, were implemented using using real-time direct volume rendering (DVR). Note that all visualizations are also shaded using the Blinn-Phong shading model on top of the specified method.

### 3.4.1 Direct Volume Rendering on the Mobile Device

To visualize the volumes, the DVR technique described by Drouin *et al.* [47], which is based on a well-known two pass rendering algorithm described by Kruger *et al.* [24] was used. This technique describes a real-time ray casting algorithm that consists of two rendering passes. In the first pass, the front and the back faces of a colored cube representing the bounding box of the volume are rendered into two different textures. The RGB color encodes the start and end positions (as 3D coordinates) of the ray for each pixel. In the second pass, for each pixel, a ray is sent through the volume and the opacity is accumulated while sampling the volume using trilinear interpolation. The ray stops and the traveled distance of the ray is recorded into a third texture.

[3] Unity Engine: https://unity.com/
(Last accessed on November 11, 2020)

Next a compute shader scans the third texture to determine the smallest and the highest nonzero depths of the texture, such that the visible interval of the volume inside the 3D texture is known. Finally, in the last pass, the final image of the volume is rendered using the recorded pixel depths, which are adjusted using the minimum and maximum values calculated previously, so that the whole range of depth values (from 0 to 1) lies within the visible part of the volume. A transfer function maps the adjusted depth values to the RGBA color for each pixel. This transfer function is encoded as a one-dimensional texture that is passed to the shader.

As mobile device GPUs are typically slower compared to their desktop equivalents, additional optimizations were made to allow for real-time rendering. First, the ray casting algorithm was simplified so that instead of accumulating opacity, the ray stops immediately when the sampled value in the volume reaches a given threshold. Second, the ray casting algorithm was modified to reduce the frequency at which the volume is sampled. To do this, the 3D Chamfer distance approach described by Zuiderveld *et al.* [25] was used. This method speeds up ray casting without compromising the quality of the rendered image by determining for every voxel, the distance to the closest non-zero voxel and storing it in a 3D texture. The distance corresponds to the number of voxels that have to be traversed to create a path in 3D space, assuming a 26-cell cubic neighborhood. Here, a small threshold value was defined to distinguish the "empty" voxels from the non-empty ones. When performing ray casting, the value from the Chamfer distance 3D texture, which indicates the distance that the ray could safely travel without missing any interesting voxels, is used. Thus, empty areas of the volume are traversed faster. It should be noted, that although the algorithm doesn't compromise the quality of the volume, it requires more space to store the additional volume.

Lastly, to save the battery life of the mobile device and also have a smoother user interface, when the volume is not being rotated, it is rendered once to a texture and then displayed in future frames. In addition, when the volume is rotated, we reduce the number of ray casts that have to be performed by downscaling and then scaling back up during the rendering to the visible frame buffer. The intensity of the downscaling is directly proportional to the speed of the rotation of the volume, making the downsampling less perceptible to the viewer.

Using these optimizations, real-time rendering was achieved on the mobile devices

tested, for all cues except Void Space Surfaces. Despite attempts to improve the calculation time of VSS, only rendering times of a few seconds/per frame were achieved. As a result a static version of VSS that can't be interacted with was used in Connect Brain.

### 3.4.2 Connect Brain Game Play

Connect Brain consists of two mini-games: (1) the "Near-Far Game", a game where players compare the relative depth between indicated vessels and (2) the "Blood Circulation Game", a game where the player must understand the connectivity between different points in the vascular volume (see Figure 18). Both mini-games are separated into a tutorial level that teaches the player the basics of the mini-game and 11 levels that can be played in any order after the completion of the tutorial. Each level is defined by four parameters: the computed tomography angiography (CTA) data, the threshold, the depth of the near and far planes, and the number of points selected on the volume (2 or more). Three CTA volumes were used in the game, for each each level one volume is used, however it is randomly rotated. Furthermore, to increase the variety between the levels different threshold values, cutting planes and number of selected points were used. Thus a slightly different dataset is presented in each round due to the fact that different parts of the volume are visible or invisible.

The cutting of the plane was implemented by modifying the colored cube as described above. Parts of the final volume were clipped by simply slicing parts from the colored cube mesh, filling the empty spaces with new faces and then coloring the faces in the correct color. To do this, the Unity ProBuilder[4] tool was used.

Each level in the game consists of 14 rounds in total, with each round showing a single visualization randomly from the 7 ones that were implemented. To reduce the player confusion, but also to avoid any biases, we decided to show each visualization twice in a row, for two consecutive rounds.

#### Depth Game

The "Near-Far Game" focuses on understanding the relative depth between vessels. This game is based on the experimental task that was described and used by Ropinski

[4]https://unity3d.com/unity/features/worldbuilding/probuilder (Accessed on September 24, 2020)

*et al.* [4], Kersten *et al.* [5] and Kreiser *et al.* [12]. The typical experimental task involves subjects determining the nearest vessel between two selected vessels rendered with a given visualization technique. The depth game in our app uses this same principle, but introduces some game play elements to make it more fun for players.

Players are presented with a CTA dataset on which two points are selected. The task of the player is to connect, using their finger, the point that looks closer to them with the one that looks further away in terms of the depth. The selected points are indicated on the volume with a contrasting color that is not present in the current visualization. Because these points are small and could be difficult to see, a black and white circle is placed around the selected point (see Figure 18(a)). This circle also indicates the region where the player can touch the screen to select the point. To further help indicate the positions of the points, arrows appear on the screen indicating the location of the points during the first second of the round. The selected points are randomly chosen, and recall that the rotation at each round is also random, meaning the player can't simply learn the correct answers. This makes replaying a level more interesting as the player will always have new data to view and interact with. Despite the points on the vessels being randomly chosen, a number of rules are applied. The chosen points are always clearly visible from to the player perspectives and always have a small minimum depth difference between them. The points also have a minimum $xy$ pixel position difference between them equal to the $diameter_{b\&wcircle} \times 1.5$ to avoid the overlapping of two indicator circles.

By connecting the indicated vessels in the correct order, the player can gain score points and additional bonus points for doing it quickly. The amount of bonus points is calculated by applying a reciprocal function to the round time. However, if the player makes an incorrect decision, the bonus is subtracted from their current score. This gives players an incentive to complete rounds as fast as possible, while simultaneously motivating them to be accurate in these decisions. Further, the score accumulates through the rounds and is saved in a global leaderboard where players can compare their score to others.

Some levels have rounds where more than two points are indicated to the player. In these rounds, the player can connect any number of points at once. The goal in that case is that all the connected points are selected in the ascending depth order, starting from the closest point in terms of the depth (similar to the work of Ritter *et.*

*al.* [7]). However, if a point with a bigger depth is selected before one with a smaller depth, then the whole selection is considered to be incorrect and the player loses the bonus time points. If all points are connected in the right order, the player will receive significantly more points than if they connected each pair of points individually. Thus, selecting multiple points at once is a high-risk, high-reward strategy.

During game play, players can rotate the volume with an offset of up to 45 degrees from the initial position. If $x$ and $y$ are the rotation in degrees around the $x-$ and $y-axis$ from the initial, then the rotation of the volume always follows the formula $\sqrt{x^2+y^2} <= 45$. To discourage rotation (as we want players to understand the data using the given visualization technique), players lose score points for rotating the volume. The amount of points lost is directly proportional to the rotation of the volume in degrees.

In a preliminary in-lab study with Connect Brain, we learned that some users wanted to know how they were wrong when they made an incorrect decision. Thus, if the feedback feature is enabled for the game, at the end of each incorrectly completed round, the volume is rotated by 90 degrees around the $x-axis$ so that the points that were closer to the viewer are position on the bottom of this view and the points that were further are positioned on the top. Vertical lines are then drawn like a ruler to demonstrate the relative depth between the points (Figure 18(b)).

### Connectivity Game

The "Blood Circulation Game" focuses on the connectivity between different vessels in the vascular volume. This game is an adaptation of the experiment that was described by Abhari *et al.* [8] where subjects were presented with static 2D images and were asked to determine if a path between two selected points on the vessel structure exists. We built on this experiment, adding motivating game play features to it.

As in the "Near-Far Game", players are presented with two or more points selected on the vascular volume. However, the goal in this game is to determine which points are directly connected, in other words does a path between the two vessels exist. As each selected point on the 2D image is associated to a specific voxel in the 3D, connectivity thus refers to the path between the two voxels inside the 3D volume. When the player finds two connected points, they link them with their finger in any

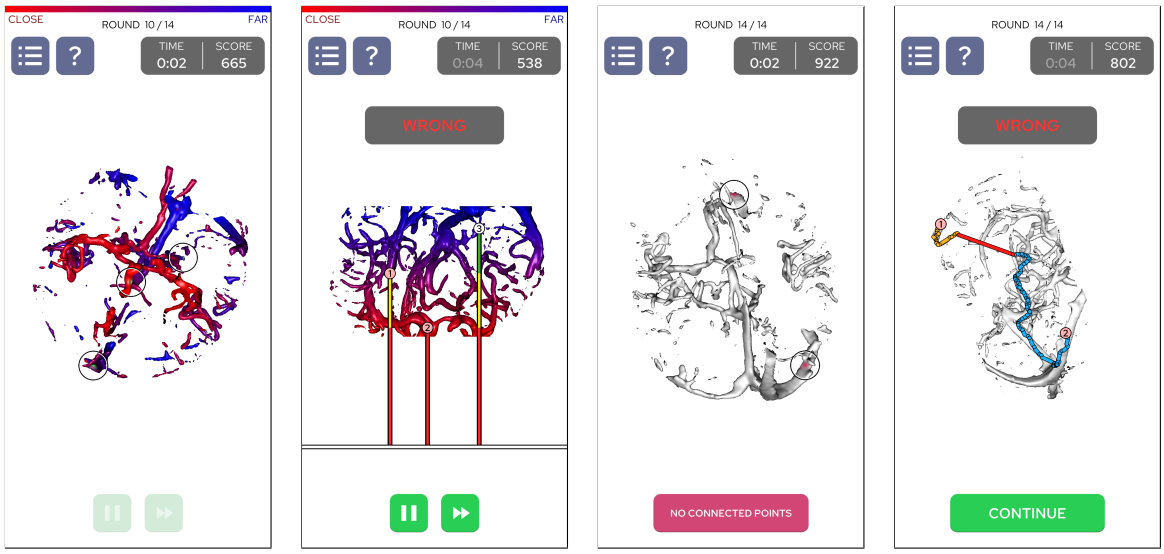



Figure 18: Screenshots from the mobile game: (a) Gameplay of the depth game, (b) Feedback for the depth game, (c) Gameplay of the connectivity game, (d) Feedback for the connectivity game

order. However, if no two points seem to be connected with each other, players should press the "No Connected Points" button that is located on the bottom of the screen (see Figure 18(c)).

As described in the first game, the initial rotation of the volume at the beginning of each round and the selection of the points are performed randomly. This means that we need to compute at runtime if two voxels are connected with each other. To to this, the A* search algorithm [54], which determines the path (if it exists) between two voxels inside the 3D texture, was used. A* is an informed search algorithm, which takes into account both the distance traversed so far and an estimation (heuristic) of the remaining path, allowing it to perform very quickly and find the optimal path in the case that the heurisitc function is admissible (never overestimates the cost to reach the goal). This algorithm requires a priority queue data structure to function, and we chose the Fiboancci Heap [55] owing to its efficient performance.

The score system works in the same way as in the "Near-Far Game", with points awarded for correct decisions about whether a path exists and for fast decision response times. The rotation of the volume also works in the same manner, resulting in points being lost.

The connectivity game also features a feedback system; if the player decides that

two points are connected, but in fact they are not, the feedback view shows the minimum distance that separates two independent parts of the vessel structure (see Figure 18(d)). Conversely, if the player decides that no points are connect with each other, but some of them are, then this view demonstrates the path between the two points.

The connectivity between neighboring voxels presents an ambiguity problem; even if two neighboring voxels are above a certain threshold, they are not necessarily visually connected. This is a well-known problem that was studied in the context of implementation of the marching cubes algorithm [56] [57]. We solve the face ambiguity problem using the asymptotic decider described by Nielson *et al.* [56]. For the internal ambiguity, we solve the problem iteratively by casting a high number of distinct Bezier splines throughout the "cube" between the two opposing vertices for which we perform the calculation. Each spline samples the intensity of the volume throughout its length, recording the smallest intensity that it could sample. After this, splines are aggregated by finding the biggest value among all smallest values that each spline could sample. This value then represents the minimum threshold that can be used during segmentation for the voxels to be connected.

## 3.5 Results

Correctness and response time were measured and analyzed using an analysis of variance (ANOVA) and post-hoc Tukey honestly significant difference (HSD) tests using the IBM SPSS Software v. 26 [5].

At the time of our analysis (Nov 1), a total of 78 subjects (49 male, 27 female, 2 other) had downloaded and played the mobile game. In addition to the 78 subjects that played the game, 14 others downloaded it but did not play. Over half the participants (59%) played on Android and the remainder on iOS. Due to the regulation for iOS apps age was only collected from Android version where the age range of participants was between 16-62 (M=31, SD=10). Of our participants, 40% had experience with medical visualization, 26% were familiar with angiography and 28% had experience with vessel visualization techniques. All 78 users participated in the depth game, completing on average 44 rounds (SD=71), but only 40% of the players

[5]https://www.ibm.com/analytics/spss-statistics-software (Last visited on Nov 12, 2020)

participated in the connectivity game, completing on average 47 rounds (SD=54).

Similar to Kersten *et al.* [5] and Lawonn *et al.* [9], for both games in addition to correctness and response time, we looked at the effect of both the distance of the indicated vessels on the screen ($xy$-distance) and the distance in depth between the indicated vessels ($z$-distance). Both $xy$ and $z$ distances were equally divided into three categories "near", "medium" or "far" measured in Unity's world coordinate space. For the $xy$ variable, the ranges are defined in the following way: near [0.162, 0.305], medium [0.305, 0.447], and far [0.447, 0.950]. For the $z$ variable, they are defined as: near [0.021, 0.076], medium [0.076, 0.147], and far [0.147, 0.792].[6]

Due to a lack of control over the timing and how the game was played (e.g. a person might get interrupted during the game increasing time) we removed all extreme outliers equal to $Q_3 + 3 * IQR$, where $Q3$ represents the value at the third quartile and $IQR$ represents the inter-quartile range, equal to $Q_3 - Q_1$.

### 3.5.1 Depth Game

A three-way repeated measures ANOVA was used to examine the main effects, as well as, the interactions of visualization method, $xy$-distance, and $z$-distance as they relate to correctness.

#### Correctness

A total of 4935 entries were collected for the depth game correctness analysis. Correctness was represented by either 1 (correct) or 0 (incorrect) determined by whether the connection between between points was done in the right order. In the case where multiple points (3 or 4) were connected at the same time, each individual pair of connected points is treated as an individual entry. The mean correctness and standard error for each visualization method is shown in Figure 19(left).

The ANOVA showed that the visualization method had a significant effect on correctness ($F(6, 4872) = 19.483, p < 0.0005$). A Tukey post hoc test showed that pseudo chromadepth ($M = 83\%, SE = 1.4\%$), aerial perspective ($M = 81\%, SE = 1.5\%$) and chromadepth ($M = 81\%, SE = 1.5\%$) allowed for significantly better

[6]Note that $z$ distances are distributed in this way because the close and the far clipping planes in some levels greatly limit the total depth range of the volume, resulting in a smaller possible depth distances.

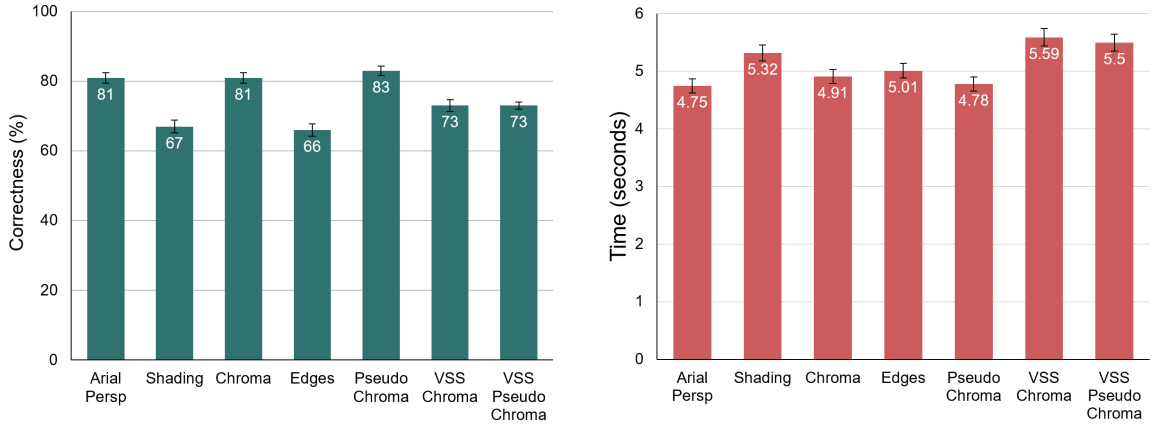



Figure 19: Mean correctness (left) and decision time (right) for for the depth game, depending on the visualization that was being used. The error bars represent the standard error.

depth perception than VSS chromadepth ($M = 73\%, SE = 1.7\%$), VSS pseudo-chromadepth ($M = 73\%, SE = 1.7\%$), shading ($M = 67\%, SE = 1.8\%$) and edge enhancement ($M = 66\%, SE = 1.8\%$). Although both VSS versions performed better than shading and edge enhancement, only the difference with edge enhancement was found to be statistically significant.

We found a significant main effect of $z$-distance on correctness ($F(2, 4872) = 18.807, p < 0.0005$). A Tukey post hoc test showed that a near $z$-distance ($M = 70\%, SE = 1.2\%$) resulted in significantly worse correctness compared to both a far ($M = 78\%, SE = 1.0\%$) and medium ($M = 76\%, SE = 1.1\%$) $z$-distances. The difference between a medium and a far $z$-distances was not significant.

We found no main effect of the $xy$ distance on correctness ($F(2, 4872) = 0.792, p = 0.453$). Also, there was no significant two-way interaction between $xy$-distance and visualization method on correctness ($F(12, 4872) = 1.246, p = 0.245$). Nor was there a significant two-way interaction between $z$-distance and visualization on correctness ($F(12, 4872) = 1.164, p = 0.303$), or between the $xy$ and $z$ distances ($F(4, 4872) = 0.230, p = 0.922$).

A significant 3-way interaction was found between the variables ($F(24, 4872) = 1.570, p = 0.038$). An interesting pattern was found where for every visualization, at each individual $xy$ distance level, a high $z$ distance produced a better correctness than a low z distance, with the only exception being the aerial perspective visualization

technique, where for a medium $xy$ distance, a near $z$ distance produced a result of $M = 85\%$ and a far $z$ distance produced a result of $M = 79\%$. Another interesting pattern was found for a near $z$ distance, where a medium $xy$ distance resulted in significantly worse results than low or high $xy$ distances for pseudo chromadepth $M = 67\%$ and for VSS chromadepth $M = 59\%$. At the same time, aerial perspective $M = 85\%$ and chromadepth $M = 84\%$ had significantly better results for medium $xy$ distances than low or medium ones at a near $z$ distance.

### Decision Time

A total of 4436 decision time entries were collected for the depth game. There are fewer time entry points (in comparison to correctness) as the tutorial levels could not be used as they had no time limit. The mean decision time and standard error for each visualization method is shown in Figure 19(right).

The decision time for levels with two points corresponds to the interval between the moment when the round started $T_0$ and the moment when the finger of the player reached the second point $T_2$. When more than two indicated vessels (i.e. $n$) are connected in the same level, the time for connecting $n-1$ with $n$ is calculated as $T_n = T_1 + T_n - T_{n-1}$. Thus, we consider the time to touch the first indicated vessel, which we consider the time the player is making decisions about the spatial layout of the vasculature as a whole, plus the time interval to connect the two indicated vessels $n-1$ and $n$.

A three-way repeated measures ANOVA was used to examine the main effects and interactions of visualization methods, $xy$-distance and $z$-distance on decision time.

The ANOVA showed that the visualization method had a significant effect on response time ($F(6, 4373) = 6.948, p < 0.0005$). A post-hoc Tukey test showed aerial perspective ($M = 4.75s, SE = 0.123s$), pseudo-chromadepth ($M = 4.78s, SE = 0.123s$) and chromadepth ($M = 4.91s, SE = 0.119s$) resulted in the fastest decision times and performed significantly better than VSS chromadepth ($M = 5.59s, SE = 0.153s$) and VSS pseudo chromadepth ($M = 5.50s, SE = 0.148s$). However, only aerial perspective performed significantly better than shading ($M = 5.32, SE = 0.140s$), which had the third worst decision time. Edge enhancement ($M = 5.01, SE = 0.125s$) was significantly faster than VSS pseudo chromadepth, but not VSS chromadepth.

There was a significant main effect of $xy$-distance ($F(2, 4373) = 4.914, p = 0.007$) on decision time. A Tukey HSD test determined that a far $xy$-distance ($M = 5.34s, SE = 0.088s$) resulted in significantly longer decision times than medium ($M = 5.04s, SE = 0.090s$) or near ($M = 4.97s, SE = 0.085s$) distances. There was no significant difference between near and medium distances.

There was a significant main effect of $z$-distance ($F(2, 4373) = 6.754, p = 0.001$) on decision time. A Tukey HSD test found that a far $xy$-distance ($M = 4.88s, SE = 0.080s$) resulted in a significantly longer decision time than medium ($M = 5.17, SE = 0.091s$) and near ($M = 5.32s, SE = 0.091s$) distances. There was no significant difference between near and medium distances.

There was no significant two-way interaction between the visualization method and the $xy$ distance ($F(12, 4373) = 0.556, p = 0.878$), the visualization method and the $z$ distance ($F(12, 4373) = 1.512, p = 0.112$), the $xy$ distance and the $z$ distance ($F(4, 4373) = 1.516, p = 0.194$). There was also no three-way interaction ($F(24, 4373) = 0.790, p = 0.753$).

### 3.5.2 Connectivity Game

A three-way repeated measures ANOVA was used to examine the main effects, as well as, the interactions of visualization method, $xy$-distance, and $z$-distance as they relate to correctness and response time for the Connectivity game.

#### Correctness

The total number of entries collected was 1680. Correctness in this game corresponds to whether the player correctly identified the indicated vessels as connected or not. The mean correctness and standard error for each visualization method is shown in Figure 20(left).

The ANOVA showed that there was a significant effect of $xy$-distance on correctness ($F(2, 1617) = 4.526, p = 0.011$). A Tukey HSD test determined that a medium $xy$ distance had the lowest correctness ($M = 79\%, SE = 1.7\%$) and performed significantly worse than both the far $xy$-distance ($M = 85\%, SE = 1.5\%$) and the near $xy$-distances ($M = 84\%, SE = 1.6\%$). However, the Tukey post-hoc test found no significant difference between the near and the far $xy$ distance.

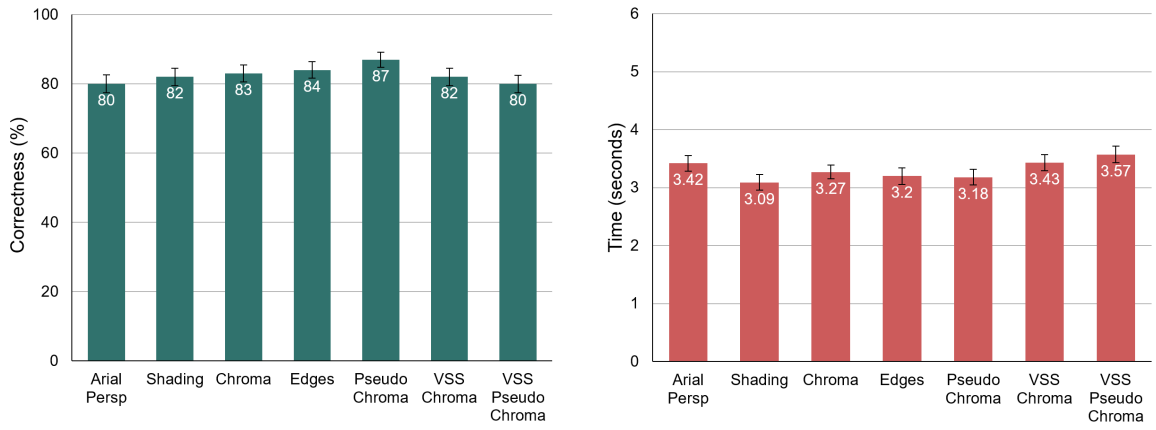



Figure 20: Mean correctness (left) and decision time (right) for for the connectivity game, depending on the visualization that was being used. The error bars represent the standard error.

There was no main effect of the visualization technique ($F(6, 1617) = 0.950, p = 0.458$) or $z$-distance ($F(2, 1617) = 0.542, p = 0.582$) on correctness. Furthermore, there was no significant two-way interaction between visualization method and $xy$ distance ($F(12, 1617) = 0.767, p = 0.686$) and visualization method and $z$ distance ($F(12, 1617) = 1.104, p = 0.352$).

The ANOVA showed a significant three-way interaction between the visualization method, $xy$ and $z$-distances ($F(24, 1617) = 2.449, p < 0.0005$). A pattern was found that a combination of far $xy$ distance and far $z$ distance resulted in the highest correctness for 5 of the 7 visualization techniques: shading $M = 100\%$, edge enhancement $M = 94\%$, pseudo-chromadepth $M = 95\%$, VSS chromadepth $M = 100\%$ and VSS pseudo chromadepth $M = 93\%$. Another interesting pattern was observed for the near $z$ distance. For all the non-VSS visualization techniques, a near $xy$ distance resulted in a higher correctness: shading $M = 93\%$, edge enhancement $M = 86\%$, aerial perspective $M = 84\%$, chromadepth $M = 94\%$, pseudo chromadepth $M = 94\%$. However, for both VSS techniques, a near $xy$ distance caused the lowest correctness among near $z$ distances: $M = 73\%$ for VSS chromadepth (while medium and far $xy$ distances both had an correctness of $M = 88\%$) and $M = 73\%$ for pseudo-chromadepth (while medium and far $xy$ distances have accuracies of $M = 83\%$ and $M = 81\%$ correspondingly).

### Decision Time

The number of entries collected for statistical analysis of decision time for the connectivity game is 1680. The mean decision time and standard error for each visualization method is shown in Figure 20(right).

There was no significant main effects of visualization method ($F(6, 1617) = 1.246, p = 0.280$), $xy$ distance ($F(2, 1617) = 0.793, p = 0.453$) or $z$ distance ($F(2, 1617) = 2.551, p = 0.078$) on decision time. No significant two-way interactions were found for the visualization technique and the $xy$ distance ($F(12, 1617) = 0.581, p = 0.859$) nor for the visualization technique and the $z$ distance and ($F(12, 1617) = 0.479, p = 0.751$). Finally, no three-way interaction was found ($F(24, 1617) = 1.225, p = 0.208$).

## 3.6 Discussion

The results of our study showed that the aerial perspective, chromadepth and pseudo chromadepth allow for the best relative depth perception. These techniques led to the most correct responses and the quickest times. For vessel connectivity no cue performed significantly better than the others.

Similar to the study done by Kersten *et al.* [5], we found that for depth perception, the aerial perspective and pseudo-chromadepth visualization techniques performed very well for both the correctness and the decision time. However, unlike Kersten *et al.* [5] and Ropinski *et al.*[4] who found pseudo-chromadepth to be significantly better that chromadepth, we found no difference between the cues. This is however, inline with the results reported by Kreiser *et al.* [12], who also found no difference between these two cues.

For the VSS cues, we found that they performed slightly worse compared to the results obtained by Kresier *et al.* [12]. Although VSS chromadepth and VSS pseudo chromadepth resulted in a higher accuracy than shading, we did not determine that this difference was statistically significant. Also, rather than having an accuracy similar to non-VSS versions of chromadepth and pseudo-chromadepth, we found the accuracy of the VSS counterparts to be worse. However, both VSS versions performed significantly better than edge enhancement (which was not studied by Kresier *et al.*). In terms of the decision response time, a similar result was found; VSS had longer times in comparison to directly applied visualization methods. This is expected due

to the indirect nature of this vessel visualization technique. The correctness results may be explained by the fact that the visualized vasculature is complex and on small devices (e.g. smartphones) there is a limited amount of background, which is needed for VSS. In addition to this, because of the hardware limitations of mobile devices, VSS was the only cue that was not being adjusted in real time when the player was rotating the volume. Also, as the study was not done in a laboratory setting, it is possible some players did not take the time to understand the VSS visualization technique properly during the tutorial levels.

Edge enhancement was not found to be an effective cue. In terms of depth perception, it resulted in the lowest correct responses, similar to shading. In terms of decision response times, it was only significantly better than VSS chromadepth, and VSS techniques are known for taking a significant amount of time to understand. In terms of vessel connectivity understanding, unlike Abhari *et al.* [8], edge enhancement did not improve the accuracy or the decision time. In fact, visualization technique had no significant impact on either correctness or response time in terms of understanding vessel connectivity. We posit this is the case because we tended to demonstrate simpler vessel structures in the connectivity game, which was achieved by using closer cutting planes to avoid having all vessels connected with each other. The negative side-effect of this was that accuracy was high across all visualizations, and the decision times were generally similar. We posit that the similarities in time could be explained by the fact that players were rotating the volume with their finger, but after removing all entries where players rotated the volume, still no effect was observed on the decision time.

In terms of distances between the indicated vessels, as expected having a far or medium $z$ distance between the vessels improves relative depth perception. Although $xy$ distance had no effect on accuracy, it did have an effect on the decision response times, with bigger $xy$ distances resulting in a higher decision time. This may be caused by the fact for higher $xy$ distances players had to perform a longer gesture when connecting the indicated vessels. On the other hand, in the connectivity game where players had to perform a similar gesture, no link was found between the $xy$ distance and the decision time, which could mean that the hand gesture doesn't have a big impact on the decision time and that a bigger $xy$ actually affects the decision time. This may because players may look back and forth between indicated vessels

more often in case of bigger distances.

For the connectivity game, medium distances resulted in the lowest correctness. This may be explained by the fact that most vessels near in $xy$ distance were connected, which is easy to discern, and most vessels far in $xy$ distance were disconnected, which was also easy to discern. Therefore, only points located at a medium distance present a challenge.

## 3.7 Conclusion

In this chapter, we described the results of a study related to the comparison of the effectiveness of cerebral blood vessel visualization techniques which was conducted using a mobile game, rather than in a traditional lab setting.

Similar to previous works, we found that aerial perspective, chromadepth and pseudo chromadepth allow for the best relative depth perception. In terms of determining the connectivity between two vessels, we found that the visualization method did not affect the result.

What differentiates our study from related works is the gamification paradigm that was used to conduct the study. Rather than having subjects perform an experiment in a lab, we created a mobile game that was distributed using mobile app distribution platforms. The main advantage of gamification is that it allowed us to have more participants than a usual lab study, and the cost of performing the study was independent of the number of participants. The participants that we got were also highly diverse, with a wide range of ages and different backgrounds. However, gamification presented some important disadvantages too, both during the development of the game and with the data collection. First, transforming the experiment into a game that is fun to play required more development time and required additional research to create interesting game mechanics. Second, implementing volume rendering so that it allows real-time rendering on mobile devices required additional optimizations to the rendering code. Third, ensuring the game worked on different devices and operating systems, GPUs, resolutions and aspect ratios also required additional development. And even though we tested our game on a variety of Android and iOS devices, we still couldn't guarantee that our game worked perfectly on all hardware configurations, as we got feedback from one participant that one of the

rendering techniques crashed on their device.

The lack of a controlled environment may also have impacted the collected data. As we could not observe how the game was played, we cannot be sure if players were motivated to try and do their best. At the same time, we think that by adding a competitive element to the study in the form of a leaderboard, did indeed motivate most players to perform well, which should have resulted in a higher quality of collected samples. We also had little control over the credibility of the data that users filled when creating their account and we could not create a detailed pre-test or post-test questionnaire on iOS due to privacy concerns. But even if this wasn't the case, having a detailed questionnaire could possibly lead to a player abandoning the game before they even start to play.

Despite some of the drawbacks of gamification, using this paradigm allowed us to collect more data samples than many similar studies [4, 5, 12, 8, 11]. Furthermore, it showed that our results were more similar to studies with more samples and subjects (2380 for Kersten *et. al.* [5] and 2850 for Kreiser *et. al.* [12]) than those with less samples (700 for Ropinski *et. al.* [4] and 600 for Abhari *et. al.* [8]).

Gamification is a promising technique for collecting large samples of data, however, it is important to have fun games that users will continue to play. We found that some players played many levels while others were perhaps not as interested in this type of game and only did the tutorial. In the future, we could improve our study by randomizing the order of the games and also by adding some illustrative techniques such as hatching [7], illustrative shadows [9] or anchors [10]. Furthermore, these games may be ported to mixed reality environments which may also attract more players.

## 3.8 Postfix: Statistical Analysis

Here, we define the terms relating to the statistical analysis that were used in this chapter. This analysis was done in order to determine not only if there were differences between the tested visualization techniques, but also to see if the $xy$ and/or $z$ distance between the vessels had an impact on depth or connectivity perception.

When describing the definitions of the related statistical measures, we assume there are $N$ discrete numerical samples. The definitions and formulas are based on those given in Tabachnick *et al.* [58].

### 3.8.1 Variance

Variance measures the spread of the numerical values from their mean. It is defined as the average of squared differences:

$$\sigma^2 = \frac{\sum(x_i - \bar{x})^2}{N}$$

This formula measures the variance of $N$ values. However, if $N$ represents the number of samples of a larger population, then $N - 1$ is used in the denominator to account for the fact that in a smaller list of samples, values are too close to their own mean. This value is called the sample variance and is calculated as follows:

$$s^2 = \frac{\sum(x_i - \bar{x})^2}{N - 1}$$

### 3.8.2 Standard Deviation

The standard deviation measures how, on average, the entries are different from their mean. Unlike variance which gives the sum of squared differences, standard deviation additionally applies the square root to the variance. This is done to obtain the answer in the original units since all the differences were squared during the variance calculation:

$$\sigma = \sqrt{\frac{\sum(x_i - \bar{x})^2}{N}}$$

Similarly, the sample standard deviation is defined in the following way:

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{N - 1}}$$

### 3.8.3 Standard Error

The standard error of the mean ($SEM$, or sometimes simply $SE$) represents an estimate of the standard deviation of the distribution of sampled means. It is calculated by dividing the standard deviation of the sample mean by the square root of the number of entries:

$$SEM = \frac{s}{\sqrt{N}}$$

The fundamental difference between standard deviation ($SD$) and standard error ($SE$) is that $SD$ measures the dispersion of the data samples (meaning, how well the mean represents the sample data), while $SE$ gives the accuracy of the sample mean. Thus, assuming uniform sampling, with a higher number of samples, $SE$ will always approach 0, meaning that the sample mean gives a more accurate representation of the population mean. However, in the case of a sample standard deviation, with a higher number of samples, the sample $SD$ will approach more and more the $SD$ of the population.

Therefore, in our study, when talking about the mean decision time or mean correctness, we always indicated $SE$ to show how accurately our obtained sample mean represents the population mean.

We used the standard deviation only in places where we wanted to demonstrate how dispersed the data was, such as when we indicated the age of the subjects or the number of rounds played in each game. In these cases, we were more interested in demonstrating the age diversity between the participants and how much they played the game.

### 3.8.4 Hypothesis Testing with Probability Density Functions

Hypothesis testing is a procedure that can be done on sampled data, where two or more mutually exclusive statements (hypotheses) about a dataset are made. The two hypothesis used with PDF functions are the following:

- The null hypothesis, which states that samples from a certain group come from a certain precise population.
- The alternative hypothesis, which states that the samples come from a different population than the one that we thought about initially.

In our case, the null hypothesis was that the time or correctness of different visualization techniques or different $xy$ or $z$ distances are equal, which means that those variables (or a combination of them) do not affect the correctness or the decision time. The alternative hypothesis was that the not all visualization technique were equal and that different visualization techniques and/or $xy$ or $z$ differences would produce better or worse results in terms of depth and spatial perception of the vessels.

In the case of statistical analysis, hypothesis testing is used with probability density functions (PDFs). These type of functions indicate how a certain sample statistical variable is distributed in multiple groups of samples. For example, if a population is uniformly sampled multiple times and the sample mean is calculated for each group of samples, then the sample means would be distributed using the normal distribution, but if we consider the variance of multiple groups of samples instead, then these variances will be distributed using the Chi-square distribution.

The key idea behind hypothesis testing with probability density functions is that if a certain statistical variable produces a result that is very unlikely to happen when looking at its PDF, then it can be concluded that the samples were not taken from the same population.

A value associated with hypothesis testing is the significance level denoted by $\alpha$. $\alpha$ represents the probability of committing a Type I error, that is, deciding that the null hypothesis is false when it is actually true. A closely related value, called the significance level, is calculated as $C = (1 - \alpha)$.

The confidence level is chosen by the researcher, although most often, a 95% confidence level is used with probability density functions. The compromise with the confidence level chosen is that while increasing the level of significance decreases the chance of committing a Type I error, it also results in sometimes accepting the null hypothesis incorrectly.

### 3.8.5 F-test

One may want to determine, given two populations $x$ and $y$, if these populations have the same variance. This procedure is used in an analysis of variance (ANOVA) (described in Section 3.8.6) to determine if three or more populations are significantly different. For the F-test, the null hypothesis is that two populations have the same variance, while the alternative hypothesis states that they have a different variance. To determine this, an F-test with a given confidence level $C$ (usually 95%) can be performed. Multiple samples are taken from both populations and the sample variances, $s_x^2$ and $s_y^2$, are calculated. Having these two sample variances, the F-ratio can be calculated as:

$$F = \frac{s_x^2}{s_y^2}$$

Note that the bigger sample variance is always located in the numerator of the formula and the smaller one is always located in the denominator.

After this, the two degrees of freedom (DoF) for the F-function are calculated for samples of both populations. The degrees of freedom correspond to the number of samples whose value could vary, when calculating a statistic, before the value of the last sample gets fixed. Assuming that $N_x$ and $N_y$ are the number of samples obtained for populations $x$ and $y$ correspondingly, the degrees of freedom are then calculated as $d_1 = N_x - 1$ and $d_2 = N_y - 1$. The number of degrees of freedom is calculated because as it changes the Chi-square distribution of variances depending on the number of samples.

Using these degrees of freedom and a specific level of significance $\alpha = (1 - C)$, it is then possible to find the critical value using an F-distribution. The F-distribution represents the ratio of two chi-square distributions. Since the distribution of sample variances for the same population follows the chi-square distribution, the F-distribution is used to represent the distribution for the F-ratio. However, because the formula of the F-distribution is complex and contains a non-elementary integral, an F-table can be used instead with pre-calculated critical values.

The critical value gives the boundary of confirmation or rejection of the hypothesis that both populations have the same variance, using the F-ratio. If the F-ratio is smaller than the critical value, both populations $x$ and $y$ are considered to have the same variance. Conversely, if the F-ratio is bigger than the critical value, populations

$x$ and $y$ are considered to have a different variance.

### 3.8.6 ANOVA

An analysis of variance (ANOVA) is a statistical test that is used to determine whether a significant difference is present among three or more sample means. Thus, it tests the null hypothesis: $\mu_1 = \mu_2 = \mu_3 = ... = \mu_M$ where $\mu_i$ corresponds to the population mean of group $i$ and $M$ corresponds to the total number of groups.

The general idea of an ANOVA is to perform an F-test using the variance between different groups and the variance within each individual group.

First, the nominator of the variance between different groups, called Sum of Squares Between ($SSB$) is calculated. Assuming there are $M$ groups with a total of $T$ entries in all groups combined, SSB is calculated in the following way:

$$SSB = \sum_{i=1}^{M} (\bar{x}_i - \bar{x})^2$$

where $\bar{x}_i$ represents the mean of the samples of group $i$ and $\bar{x}$ represents the mean of all samples from all groups combined.

Second, the nominator of the variance within each individual group called Sum of Squares Within ($SSW$) is calculated:

$$SSW = \sum_{i=1}^{M} \sum_{j=1}^{N_i} (x_{i,j} - \bar{x}_i)^2$$

where the number of samples in each group is noted by $N_i$ and that the sample at index $j$ in the group $i$ is noted by $x_{i,j}$.

The denominators for the variance, which will be referred to as the number of degrees of freedom, are calculated as $df_b = T - M$ for the "between" variance and $df_w = M - 1$ for the "within" variance. The between variance corresponds to the variance between different groups, while the within variance corresponds to the variance between samples in each group. Generally speaking, ANOVA results in rejecting the null hypothesis when the "between" variance is high and the "within" variance is low, because this means that values in every group are close to each other, but the groups themselves have very different values, which means that the groups don't have the same means.

Having this information, we can calculate the "between" variance called Mean Squares Between ($MSB$) as:

$$MSB = \frac{SSB}{df_b}$$

and the variance "within" called Mean Squares Within ($MSW$) as:

$$MSW = \frac{SSW}{df_w}$$

With these two values, the F-ratio can be calculated as

$$F = \frac{MSB}{MSW}$$

Next, the F-test is calculated on the obtained F-ratio to see if the groups have the same variance. If the variances differ, it can be deduced that at least one of the groups has samples that come from a different population.

In a one-way ANOVA, each sample belongs to a certain group defined by a discrete independent variable (DIV or simply IV) stored with each sample. However, it is possible for the samples to be partitioned in a number of alternative ways using multiple different IVs. In this case, for each independent variable, the sample stores the group to which it belongs within the IV. It is then possible to perform an ANOVA while partitioning the data in a multitude of ways. This type of analysis is called an $N$-way ANOVA, where $N$ corresponds to the number of independent variables. A three-way ANOVA was performed in our analysis where our discrete independent variables were the vessel visualization technique, the $xy$ and the $z$ distance between the vessels.

In an $N$-way ANOVA, in addition to performing the analysis for each independent variable individually, it is also possible to combine some IVs together to create a bigger number of smaller groups that can also be compared between each other. Such combination of IVs is called an interaction. Combining $M$ independent variables results in performing an $M$-way interaction.

### 3.8.7 Tukey's HSD Test

If is determined using an ANOVA that at least one of the groups has samples that don't come from the population, a Tukey's HSD Test can be used to determined which

groups precisely are different from other groups in a statistically significant manner. This test is performed pairwise on all combinations of groups. The calculation is as follows:

$$HSD = \frac{\bar{x_1} - \bar{x_2}}{SE}$$

Where $\bar{x_1}$ and $\bar{x_2}$ are means of the two groups, and $SE$ is the standard error of all the samples of all groups, using the previously calculated values in the ANOVA, the standard error is:

$$SE = \sqrt{\frac{MSW}{T}}$$

After the $HSD$ value is calculated, it is compared with the level of significance $\alpha$. If $HSD$ is smaller than $\alpha$, the difference between the groups is statistically significant and the samples of each group come from different populations. If $HSD$ is bigger than $\alpha$, then the difference is not statistically significant and samples of both groups come from the same population.

For example, the results of our ANOVA showed that there were differences between visualization methods, and to determine which methods were significantly different, a pairwise comparison using Tukey's HSD test was performed.

# Chapter 4

# Conclusion

Advances in commodity computer graphics hardware have provided new opportunities for developing and studying visualization techniques in medical imaging. We have leveraged these advances to study the use of different vascular visualization techniques in the human perception of 3D CT angiography images. Specifically, in this thesis, we explored the use of gamification to determine the effectiveness of certain cerebral vessel visualization techniques. We developed "Connect Brain", a game for both Android and iOS platforms that allowed us to collect gameplay data that was related to the effectiveness of the varying visualization techniques on spatial and depth understanding of CTA data. The visualization techniques were compared in terms of both accuracy and decision time for both deciding upon the relative depth relationship and the connectivity between two or more vessels.

Prior to publishing the game on Google Play and in the App Store, we ran a preliminary study in a supervised environment. This allowed us to gather feedback from players on what interesting game mechanics we could use to encourage longer game play. Based on these preliminary results, we added a leaderboard at the end of each level to have players compete between each other and we also added a feedback view at the end of each incorrect round so that the player could visualize the correct answer.

After analyzing the data collected data from the "Connect Brain" app, we found that for determining the depth relationship between vessels, aerial perspective, chromadepth and pseudo chromadepth performed the best, both in terms of the correctness and the decision response time. Conversely, shading and edge enhancement

performed the worst, resulting in the lowest accuracies and mediocre decision times. VSS cues performed slightly worse than expected, with VSS chromadepth and VSS pseudo chromadepth performing worse than their non-VSS counterparts. For the connectivity game, we found that the visualization technique that was used had no significant impact on either the correctness or the decision time.

In terms of how are results compare to other studies, in general, we found that they were more similar to studies that also had a significant number of participants and samples points (like the studies by Kersten *et. al.* [5] and Kreiser *et. al.* [12]) rather than studies with few participants and samples (Abhari *et. al.*[8] and Ropinski *et. al.*[4]). This suggests the importance of having an appropriate number of subjects and trials when evaluating visualization paradigms.

## 4.1 Future Work

Using gamification in the context of medical visualization has significant potential, and with the improvements of mobile device hardware and the accessibility of modern app marketplaces, these kind of studies will be increasingly easier to perform and more effective to gather research data.

In the future, we could improve the game by making it more appealing to a broader audience, rather than only the people interested in medical imaging. Sound and music could be added and in general the graphical assets and special effects could be improved. Another, idea would be to introduce some concept of currency that the player accumulates by playing the game, which could then be use to unlock temporary "cheats" in the game. An advertisement campaign could also be used to promote the app more broadly and encourage more players to download the game.

Regarding the study itself, in the future, illustrative techniques could be added to compare an even higher number of visualizations. Some good candidates are the hatching and distance-encoded shadows technique described by Ritter *et al.* [7], the illustrative shadows, supporting lines and contours technique described by Lawonn *et al.* [9] and the anchors technique also described by Lawonn *et al.* [10]. Furthermore, visualizing other types of medical images for spatial and depth understanding could also be explored.

Additionally, we could incorporate the capabilities of the gyroscope and/or the

accelerometer sensors, available on modern smartphones and tablets by. For example, instead of rotating the volume with their finger, players could try and rotate the device itself. Alternatively, we could transform the input system so that instead of selecting the points with the finger in the depth game, players would rather have to tilt their phone in the direction of the points that looks further away in terms of depth. Additionally, we could take advantage of the camera of the mobile device and create some mixed reality visualizations.

As mobile hardware continues to improve, we believe performing gamified studies would become easier and common, even in the context of computationally costly domains such as medical imaging. The results of these types of studies can not only be used to inform what the most appropriate medical visualization techniques are in general but also to tailor more specific clinical studies (e.g. clipping an aneurysm) with domain experts (e.g. neurosurgeons).

# Bibliography


[1] Steven P. Callahan, Jason H. Callahan, Carlos E. Scheldegger, and Claudio T. Silva. Direct volume rendering: A 3d plotting technique for scientific data. *Computing in Science and Engineering*, 10(1):88–91, January 2008.

[2] Dirk Bartz and Michael Meiner. Voxels versus polygons: A comparative approach for volume graphics. 01 2000.

[3] Bernhard Kainz, Rupert H Portugaller, Daniel Seider, Michael Moche, Philipp Stiegler, and Dieter Schmalstieg. Volume visualization in the clinical practice. In *Workshop on Augmented Environments for Computer-Assisted Interventions*, pages 74–84. Springer, 2011.

[4] Timo Ropinski, Frank Steinicke, and Klaus Hinrichs. K.h.: Visually supporting depth perception in angiography imaging. pages 93–104, 07 2006.

[5] Marta Kersten-Oertel, Sean J. Chen, and D. Louis Collins. An evaluation of depth enhancing perceptual cues for vascular volume visualization in neurosurgery. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):391–403, March 2014.

[6] Bernhard Preim, Alexandra Baer, Douglas Cunningham, Tobias Isenberg, and Timo Ropinski. A survey of perceptually motivated 3d visualization of medical image data. In *Computer Graphics Forum*, volume 35, pages 501–525. Wiley Online Library, 2016.

[7] F. Ritter, C. Hansen, V. Dicken, O. Konrad, B. Preim, and H. Peitgen. Real-time illustration of vascular structures. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):877–884, 2006.

[8] Kamyar Abhari, John S. H. Baxter, Roy Eagleson, Terry Peters, and Sandrine de Ribaupierre. Perceptual enhancement of arteriovenous malformation in MRI angiography displays. In Craig K. Abbey and Claudia R. Mello-Thoms, editors, *Medical Imaging 2012: Image Perception, Observer Performance, and Technology Assessment*, volume 8318, pages 70 – 77. International Society for Optics and Photonics, SPIE, 2012.

[9] Kai Lawonn, Maria Luz, Bernhard Preim, and Christian Hansen. Illustrative visualization of vascular models for static 2d representations. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 399–406, 10 2015.

[10] Kai Lawonn, Maria Luz, and Christian Hansen. Improving spatial perception of vascular models using supporting anchors and illustrative visualization. *Computers & Graphics*, 63:37 – 49, 2017.

[11] Simon Drouin, Daniel DiGiovanni, Marta Kersten-Oertel, and Louis Collins. Interaction driven enhancement of depth perception in angiographic volumes. *IEEE Transactions on Visualization and Computer Graphics*, PP:1–1, 12 2018.

[12] Julian Kreiser, Pedro Hermosilla, and Timo Ropinski. Void space surfaces to convey depth in vessel visualizations. *arXiv: Graphics*, 2018.

[13] Kristen Dergousoff and Regan L. Mandryk. Mobile gamification for crowdsourcing data collection: Leveraging the freemium model. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 1065–1074, New York, NY, USA, 2015. Association for Computing Machinery.

[14] Nafees Ahmed and Klaus Mueller. Gamification as a paradigm for the evaluation of visual analytics systems. pages 78–86, 11 2014.

[15] Marcos Balsa Rodríguez and Pere Pau Vázquez Alcocer. Practical volume rendering in mobile devices. In *International Symposium on Visual Computing*, pages 708–718. Springer, 2012.

[16] José M Noguera, Juan-Roberto Jiménez, Carlos J Ogáyar, and Rafael Jesús Segura. Volume rendering strategies on mobile devices. In *GRAPP/IVAPP*, pages 447–452, 2012.

[17] Movania Muhammad Mobeen and Lin Feng. Ubiquitous medical volume rendering on mobile devices. In *International Conference on Information Society (i-Society 2012)*, pages 93–98. IEEE, 2012.

[18] Tomasz Hachaj. Real time exploration and management of large medical volumetric datasets on small mobile devices—evaluation of remote volume rendering approach. *International Journal of Information Management*, 34(3):336–343, 2014.

[19] Marta Kersten-Oertel, A Stewart, Nikolaus Troje, and Randy Ellis. Enhancing depth perception in translucent volumes. *IEEE transactions on visualization and computer graphics*, 12:1117–23, 10 2006.

[20] James F. Blinn. Models of light reflection for computer synthesized pictures. *SIGGRAPH Comput. Graph.*, 11(2):192–198, July 1977.

[21] Michael Bailey and Dru Clark. Using chromadepth to obtain inexpensive single-image stereovision for scientific visualization. *Journal of Graphics Tools*, 3(3):1–9, 1998.

[22] Eric B. Lum and Kwan-Liu Ma. Hardware-accelerated parallel non-photorealistic volume rendering. In *Proceedings of the 2nd International Symposium on Non-Photorealistic Animation and Rendering*, NPAR '02, page 67–ff, New York, NY, USA, 2002. Association for Computing Machinery.

[23] Winter Mason and Siddharth Suri. A guide to conducting behavioral research on amazon's mechanical turk. *Behavior research methods*, 44:1–23, 06 2011.

[24] J. Kruger and R. Westermann. Acceleration techniques for gpu-based volume rendering. In *IEEE Visualization, 2003. VIS 2003.*, pages 287–292, 2003.

[25] Karel J. Zuiderveld, Anton H. J. Koning, and Max A. Viergever. Acceleration of ray-casting using 3-D distance transforms. In Richard A. Robb, editor, *Visualization in Biomedical Computing '92*, volume 1808, pages 324 – 335. International Society for Optics and Photonics, SPIE, 1992.

[26] Robert A. Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *SIGGRAPH Comput. Graph.*, 22(4):65–74, June 1988.

[27] Sanjana Patrick, N. Birur, Keerthi Gurushanth, A. Raghavan, and Shubha Gurudath. Comparison of gray values of cone-beam computed tomography with hounsfield units of multislice computed tomography: An in vitro study: Official publication of indian society for dental research. *Indian Journal of Dental Research*, 28(1), Jan 2017. Copyright - Copyright Medknow Publications & Media Pvt. Ltd. Jan/Feb 2017; Last updated - 2020-09-09.

[28] Bernhard Preim and Dirk Bartz. *Visualization in medicine: theory, algorithms, and applications*. Elsevier, 2007.

[29] Horst K. Hahn, Bernhard Preim, Dirk Selle, and Heinz Otto Peitgen. Visualization and interaction techniques for the exploration of vascular structures. In *Proceedings of the Conference on Visualization '01*, VIS '01, page 395–402, USA, 2001. IEEE Computer Society.

[30] 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.

[31] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[32] K. Mueller, N. Shareef, Jian Huang, and R. Crawfis. High-quality splatting on rectilinear grids with efficient culling of occluded voxels. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):116–134, 1999.

[33] 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*, SIGGRAPH '94, page 451–458, New York, NY, USA, 1994. Association for Computing Machinery.

[34] Peter M. Hall and Alan H. Watt. *Rapid Volume Rendering Using a Boundary-Fill Guided Ray Cast Algorithm*, page 235–249. Springer-Verlag, Berlin, Heidelberg, 1991.

[35] Timothy J. Cullip and Ulrich Neumann. Accelerating volume reconstruction with 3d texture hardware. Technical report, USA, 1994.

[36] Randima Fernando et al. *GPU gems: programming techniques, tips, and tricks for real-time graphics*, volume 590. Addison-Wesley Reading, 2004.

[37] Tom McReynolds and David Blythe. *Advanced graphics programming using OpenGL*. Elsevier, 2005.

[38] Orion Wilson, Allen VanGelder, and Jane Wilhelms. Direct volume rendering via 3d textures. Technical report, USA, 1994.

[39] Henning Scharsach. Advanced gpu raycasting. In *In Proceedings of CESCG 2005*, pages 69–76, 2005.

[40] Jose M Noguera and J Roberto Jimenez. Mobile volume rendering: past, present and future. *IEEE transactions on visualization and computer graphics*, 22(2):1164–1178, 2015.

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

[42] Tomas Akenine-Moller, Eric Haines, and Naty Hoffman. *Real-time rendering; 3rd ed.* A K Peters, Wellesley, MA, 2008.

[43] Sanjeev J. Koppal. *Lambertian Reflectance*, pages 441–443. Springer US, Boston, MA, 2014.

[44] Forrester Cole, Kevin Sanik, Doug DeCarlo, Adam Finkelstein, Thomas Funkhouser, Szymon Rusinkiewicz, and Manish Singh. How well do line drawings depict shape? *ACM Trans. Graph.*, 28(3), July 2009.

[45] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast—but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, page 254–263, USA, 2008. Association for Computational Linguistics.

[46] M. Marge, S. Banerjee, and A. I. Rudnicky. Using the amazon mechanical turk for transcription of spoken language. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5270–5273, 2010.

[47] Simon Drouin and Louis Collins. Prism: An open source framework for the interactive design of gpu volume rendering shaders. *PLOS ONE*, 13:e0193636, 03 2018.

[48] David Ebert and Penny Rheingans. Volume illustration: Non-photorealistic rendering of volume models. In *Proceedings of the Conference on Visualization '00*, VIS '00, page 195–202, Washington, DC, USA, 2000. IEEE Computer Society Press.

[49] Richard Arend Steenblik. The Chromostereoscopic Process: A Novel Single Image Stereoscopic Process. In David F. McAllister and Woodrow E. Robbins, editors, *True Three-Dimensional Imaging Techniques & Display Technologies*, volume 0761, pages 27 – 34. International Society for Optics and Photonics, SPIE, 1987.

[50] P. Thompson, K. A. May, and Robert Stone. Chromostereopsis: a multicomponent depth effect? *Displays*, 14:227–234, 1993.

[51] Satoshi Suzuki and Keiichi Abe. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30:32–46, 1985.

[52] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, ACM '68, page 517–524, New York, NY, USA, 1968. Association for Computing Machinery.

[53] Tobias Ritschel, Thorsten Grosch, and Hans-Peter Seidel. Approximating dynamic global illumination in image space. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*, I3D '09, page 75–82, New York, NY, USA, 2009. Association for Computing Machinery.

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

[55] Michael L. Fredman and Robert Endre Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34(3):596–615, July 1987.

[56] G. M. Nielson and B. Hamann. The asymptotic decider: resolving the ambiguity in marching cubes. In *Proceeding Visualization '91*, pages 83–91, 1991.

[57] Evgeni Chernyaev. Marching cubes 33: Construction of topologically correct isosurfaces. Technical report, 1995.

[58] Barbara G Tabachnick and Linda S Fidell. *Experimental designs using ANOVA*. Thomson/Brooks/Cole Belmont, CA, 2007.