

Professur für
Kognitive Sensorsysteme

Exploring Trajectory Planning in Volume Rendering alongside Transfer Function Interpolation based on Perceptually Uniform Color Spaces

Masterarbeit von

Jorge Korgut Junior

1. PRÜFER

2. PRÜFER

3. PRÜFER

Prof. Dr. Christoph Heinzl Prof. Dr. Harald Kosch Dr. Eric Guérin

June 11, 2025

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Structure of the Thesis	5
1.3	Main Contributions	6
2	Background	7
2.1	Orientation	7
2.2	Position Interpolation	9
2.3	Color Spaces	13
2.4	Color Distances	15
2.5	Volume Rendering	17
3	Related Work	23
3.1	Color Appearance Models	23
3.2	Perceptual Color Metrics	27
4	Methodology	33
4.1	Camera Trajectory Interpolation	34
4.2	General Transfer Function Interpolation Design	42
4.3	Transfer Function Interpolation in a Timeline	46
5	Results & Evaluations	48
5.1	Evaluation Design for Camera Trajectory Interpolation	49
5.2	Evaluation Results regarding Camera Trajectory Interpolation	52
5.3	Evaluation Design for Perceptually Uniform Transfer Functions	55
5.4	Evaluation Results regarding Perceptually Uniform Transfer Functions	59
6	Discussion	68

Contents

7 Conclusion	70
7.1 Limitations and future work	72
Appendix A Color Conversions	74
A.1 CIERGB and CIEXYZ	74
A.2 CIEXYZ and CIELUV	74
A.3 CIEXYZ and CIELAB	75
Appendix B Color Distances	77
B.1 $\Delta E_{CIE1994}$	77
B.2 $\Delta E_{CIE2000}$	77
Bibliography	80
Eidesstattliche Erklärung	84

Abstract

Creation of fluid and visually appealing animations is a crucial factor for the effective communication of ideas and concepts. In this context, this thesis visualises and animates volumetric data used, for example, in pre-operative planning, material analysis and promotional content, among others. For this purpose techniques for camera path planning and for the export of high-quality videos were designed and implemented using the Visualisation Toolkit (VTK) in C++. Furthermore, a method for interpolation of transfer functions is presented that allows to isolate regions of interest that may change over time during the animation. Another research focus addresses a gap in the literature regarding the use of perceptually uniform color spaces in this context. Such color spaces have the potential to produce more accurate and perceptually consistent color transitions. We compare transfer functions defined in CIERGB with those using perceptually uniform color spaces such as ProLAB and CIELAB, as well as custom spaces designed specifically to minimize the CIEDE2000 color difference. The evaluation of the presented techniques shows the effectiveness of this framework in different scenarios and provides a solid basis for future research on perceptually uniform colour spaces in volume rendering.

Résumé

La création d'animations fluides et esthétiques constitue un facteur crucial pour la communication efficace d'idées et de concepts. Dans ce contexte, ce mémoire vise à visualiser et animer des données volumiques utilisées, par exemple, dans la planification préopératoire, l'analyse de matériaux ou encore le contenu promotionnel, entre autres. À cette fin, des techniques de planification de trajectoires de caméra et d'exportation de vidéos de haute qualité ont été conçues et implémentées à l'aide de la Visualisation Toolkit (VTK) en C++. En outre, une méthode d'interpolation des fonctions de transfert est présentée, permettant d'isoler des régions d'intérêt susceptibles d'évoluer au cours de l'animation. Un autre axe de recherche aborde une lacune de la littérature concernant l'utilisation d'espaces colorimétriques perceptuellement uniformes dans ce contexte. De tels espaces ont le potentiel de produire des transitions de couleur plus précises et plus cohérentes du point de vue perceptuel. Nous comparons les fonctions de transfert définies dans l'espace CIERGB avec celles utilisant des espaces perceptuellement uniformes tels que ProLAB et CIELAB, ainsi que des espaces personnalisés conçus spécifiquement pour minimiser la différence de couleur CIEDE2000. L'évaluation des techniques présentées montre l'efficacité de ce cadre dans différents scénarios et fournit une base solide pour des recherches futures sur les espaces colorimétriques perceptuellement uniformes en rendu volumique.

Zusammenfassung

Die Erstellung flüssiger und visuell ansprechender Animationen ist ein entscheidender Faktor für die effektive Kommunikation von Ideen und Konzepten. In diesem Zusammenhang werden in dieser Arbeit volumetrischen Daten visualisiert und animiert, die beispielsweise in der präoperativen Planung, der Materialanalyse und in Werbeinhalten eingesetzt werden. Zu diesem Zweck wurden Techniken für die Kamerapfadplanung und für den Export qualitativ hochwertiger Videos unter Verwendung des Visualization Toolkit (VTK) in C++ konzeptioniert und implementiert. Darüber hinaus wird eine Methode zur Interpolation von Transferfunktionen vorgestellt, die es ermöglicht, interessante Regionen zu isolieren, die sich während der Animation mit der Zeit verändern können. Ein weiterer Forschungsschwerpunkt befasst sich mit einer Lücke in der Literatur bezüglich der Verwendung von perzeptuell uniformen Farbräumen in diesem Zusammenhang. Solche Farbräume haben das Potenzial, genauere und perzeptiv konsistente Farbübergänge zu erzeugen. Wir vergleichen Transferfunktionen, die in CIERGB definiert sind, mit solchen, die perzeptuell uniforme Farbräume wie Pro-LAB und CIELAB verwenden, sowie mit benutzerdefinierten Räumen, die speziell zur Minimierung der CIEDE2000-Farbdifferenz entwickelt wurden. Die Evaluierung der präsentierten Techniken zeigt die Effektivität dieses Frameworks in verschiedenen Szenarien und bietet eine solide Grundlage für künftige Forschungen hinsichtlich perzeptuell uniformer Farbräume im Volume Rendering.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to Christoph Heinzl and Eric Guerin for their consistent support throughout the development of this master's thesis. The journey was long, challenging, and filled with obstacles, but their guidance enabled me not only to complete this work, but also to grow both technically and emotionally in ways that will stay with me for life.

I would also like to thank Ophelie Coueffe and Harald Kosch for their valuable administrative assistance during this period.

My sincere appreciation goes to Fraunhofer for welcoming me as an intern and inspiring many of the ideas and concepts explored in this thesis. In particular, I would like to thank Thomas Lang for his support during the internship and Tomas Sauer for his brilliant mathematical insights.

I am also grateful to Alexander Gall for his technical support and for the useful advice he shared on how to structure and write this work.

I wish to thank my family for their unconditional support, which allowed me to focus fully on this project over an extended period. Their encouragement has been invaluable. I am especially thankful to my father, who has always believed in me; my mother, whose love has been a source of strength; to my girlfriend, whose support helped me persevere through challenging moments; and my sister, whose joy and smiles brought light to this journey and made the entire experience easier.

Finally, I would like to acknowledge the use of OpenAI's ChatGPT as a support tool in refining the phrasing of certain sections of this thesis. While the ideas and content are entirely my own, AI assistance was employed to enhance the clarity of my original wording and to generate illustrative images. This support was strictly limited to language improvement and visual representation, without any contribution to the conceptual or analytical aspects of the work.

List of Figures

1.1	Representation of the same sea surface temperature flux data using two different color palettes: one with five nodes (black, white, blue, yellow, and red) and another using only a grayscale gradient from black to white. [Buj23] [NAS]	3
2.1	Illustration of the construction process of a quadratic Bézier curve.	10
2.2	Illustration of the Catmull-Rom construction of the speed vectors V_1 and V_2 .	11
2.3	Construction of Catmull-Rom interpolation curves by deriving the intermediate variables V_1 and V_2 used in Hermite interpolation.	12
2.4	Just Noticeable Difference (JND) example: (a) Comparison of two red tones where the difference is not perceptible; (b) Comparison of two red tones where the difference is noticeable.	17
2.5	Illustration of four volume rendering methods applied to CT scan data from the VTK Examples Dataset [VTK25] using raymarching: (a) maximum intensity projection, (b) compositing accumulation, (c) averaging, and (d) first-hit detection.	18
2.6	Illustration of a 1D transfer function widget. The horizontal axis represents scalar values ranging from the minimum to the maximum of the volumetric dataset, while the vertical axis corresponds to opacity varying from 0 to 1.	20
2.7	Illustration of the 2D transfer function widget using a CT scan dataset from the VTK Examples collection [VTK25]. The horizontal axis represents the range of scalar values of the volume data, while the vertical axis represents the range of the gradient magnitude. The gray intensity of each pixel is logarithmically proportional to the number of voxels in the volume that match the corresponding scalar-gradient pair.	21

List of Figures

3.1	Illustration of unintended color appearance during interpolation between blue and gray. (a) Interpolation in the CIERGB color space. (b) Interpolation in the CIELAB color space. (c) Interpolation in the ProLAB color appearance model.	26
3.2	Color distance matrix for the defined transfer function in (a) using the three defined color distance metrics normalized between 0 and 1 and plotted in gray scale: (b) ΔE_{1976} , (c) ΔE_{1994} , and (d) ΔE_{2000}	30
3.3	Visualization of just noticeable differences (JND) for a transfer function color palette. Green areas represent perceptible differences according to the JND threshold, while gray areas indicate imperceptible differences. (a) JND visualization in the RGB color space. (b) Corresponding JND visualization in the CIELAB color space using the same set of control points. (c) JND visualization in the ProLAB color appearance model. . .	31
4.1	Diagram illustrating the volume visualization pipeline for the video export framework presented in this master thesis.	33
4.2	(a) Illustration of the construction of Hermite segments when two points (P_0 , P_8) coincide at the same position. (b) In the starting segment, P_2 and P_7 are the directional control points, while P_0 and P_1 are the start and end points, respectively. (c) In the end segment, P_6 and P_1 are the directional control points, while P_7 and P_8 are the start and end points, respectively.	35
4.3	Illustration of the construction of a starting Hermite segment with a straight line, where P_1 and P_2 are the starting and ending points, and P_0 and P_3 are the directional control points. (a) The speed V_1 is chosen to be null by setting $P_2 = P_0$. (b) The speed V_2 is calculated following the standard rules. (c) The curve is constructed using Hermite interpolation. (d) An example of undesirable behavior when the Hermite interpolation begins without aligning the initial segment along a straight path.	36
4.4	Illustration of camera orientation and position representation. (a) The camera's state is defined using its position, view-up vector, and focal point. (b) The camera's state is defined using a quaternion and its position.	37

List of Figures

4.5 (a) Illustration of an unintended loop caused by placing two nearby control points in a Catmull-Rom spline, resulting in a smooth but undesired motion path. (b) Desired behavior when setting a pause by placing two control points at the exact same position, producing a stationary segment in the animation.	38
4.6 Illustration showing camera misalignment caused by separate interpolation of position and orientation. The point of convergence for all orientations is not centered, and it varies between spline segments.	41
4.7 Widget displaying the timeline of our framework, encoding transfer function control points, color palette interpolation, and transfer function definition through hovering.	46
5.1 Volume-rendered images generated from user-defined transfer functions during the evaluation of perceptual uniformity in direct volumetric rendering.	48
5.2 (a) Image used in Phase 1 to familiarize participants with 1D transfer functions, guiding them to fine-tune the image and arrive at a desired outcome. (b) Reference brain image that participants aim to replicate by adjusting the transfer function [AI25].	55
5.3 Tutorial page used in the perceptual uniformity user study for volume rendering transfer function evaluation.	58
5.4 Time spent by participants across the three phases of the user study on perceptually uniform color spaces in transfer function design.	61
5.5 Aggregated color distance difference chart between ProLAB and CIERGB. The y-axis shows the difference in color distance (ΔE_{2000}) between two consecutive nodes, computed as ProLAB minus CIERGB.	64
5.6 Aggregated color distance difference chart between ProLAB and CIELAB. The y-axis shows the difference in color distance (ΔE_{2000}) between two consecutive nodes, computed as ProLAB minus CIELAB.	66
5.7 Illustration of the color distance difference chart, incorporating the opacity defined by the transfer functions.	67

1 Introduction

What we observe is not nature itself, but
nature exposed to our method of questioning.

Werner Heisenberg

This chapter outlines the motivation that inspired the development of this thesis. It provides an overview of the current technologies, their applications, and the limitations that this work aims to address. The research questions guiding the study are presented along with their interrelations. An outline of the thesis structure follows, and finally, the chapter clarifies the claims made and any disclaimers associated with our research.

1.1 Motivation

Volume rendering has been extensively used over the past decades to reveal internal structures in the data in realistic representations. With advancements in CT scans, X-rays, MRI, and other volumetric data acquisition techniques, direct volume visualization has become a powerful tool for dealing with real-world challenges from medical diagnosis to tasks in industrial quality control. Simple algorithms can be used transform volumetric data into respective renderings. For example, this can be generally achieved through a ray marching algorithm that traverses the volume, sampling scalar values at each position and accumulating optical properties to construct the final image.

At the core of direct volume visualization is a critical function responsible for converting raw data into visually interpretable and meaningful representations. These functions, known as transfer functions, act as mapping mechanisms that translate data values into perceivable attributes. The most common transfer functions map scalar values to a 3D color space incorporating opacity, enhancing visualization and interpretation for

domain-specific applications. This common approach, known as transfer functions and more specifically as 1D transfer functions, has been widely adopted in the field. Utilizing a 3D RGB color space, they fulfill the fundamental requirements for transforming raw data into visually meaningful, accurate, and aesthetically appealing representations. By definition, transfer functions may seem straightforward, but their complexity becomes evident when generating high-quality images. In practice, configuring the optical parameters of a transfer function remains highly domain-specific and challenging. Even for experienced users, fine-tuning these parameters to achieve the desired visual representation can take up to 30 minutes or more, making the process both time-consuming and complicated.

It becomes evident that aside opacity, especially color is a critical factor to consider. An open research question in the field concerns the often suboptimal nature of color palettes used in transfer functions, largely due to their lack of perceptual uniformity. This issue is not unique to volume rendering: in other domains, simple interpolated color palettes have been criticized for their non-uniformity, which limits their ability to reveal fine details and increases the risk of introducing interpretative biases. By examining the Figure 1.1, we can see how different color palettes (transfer functions) may yield varying interpretations of the same data. The differences in visualization arise primarily due to the non-uniformity of color interpolation, which affects how details are perceived. This observation leads to our first research questions:

(Q1) How can perceptually uniform color palettes be used in Volume rendering?

- (Q1.1) How can the perceptual uniformity of transfer functions be objectively evaluated?
- (Q1.2) What visual enhancements do perceptually uniform color spaces introduce in the context of volume rendering?
- (Q1.3) How do color interpolations in perceptually uniform spaces compare to traditional linear RGB interpolation when used in animations?

1 Introduction

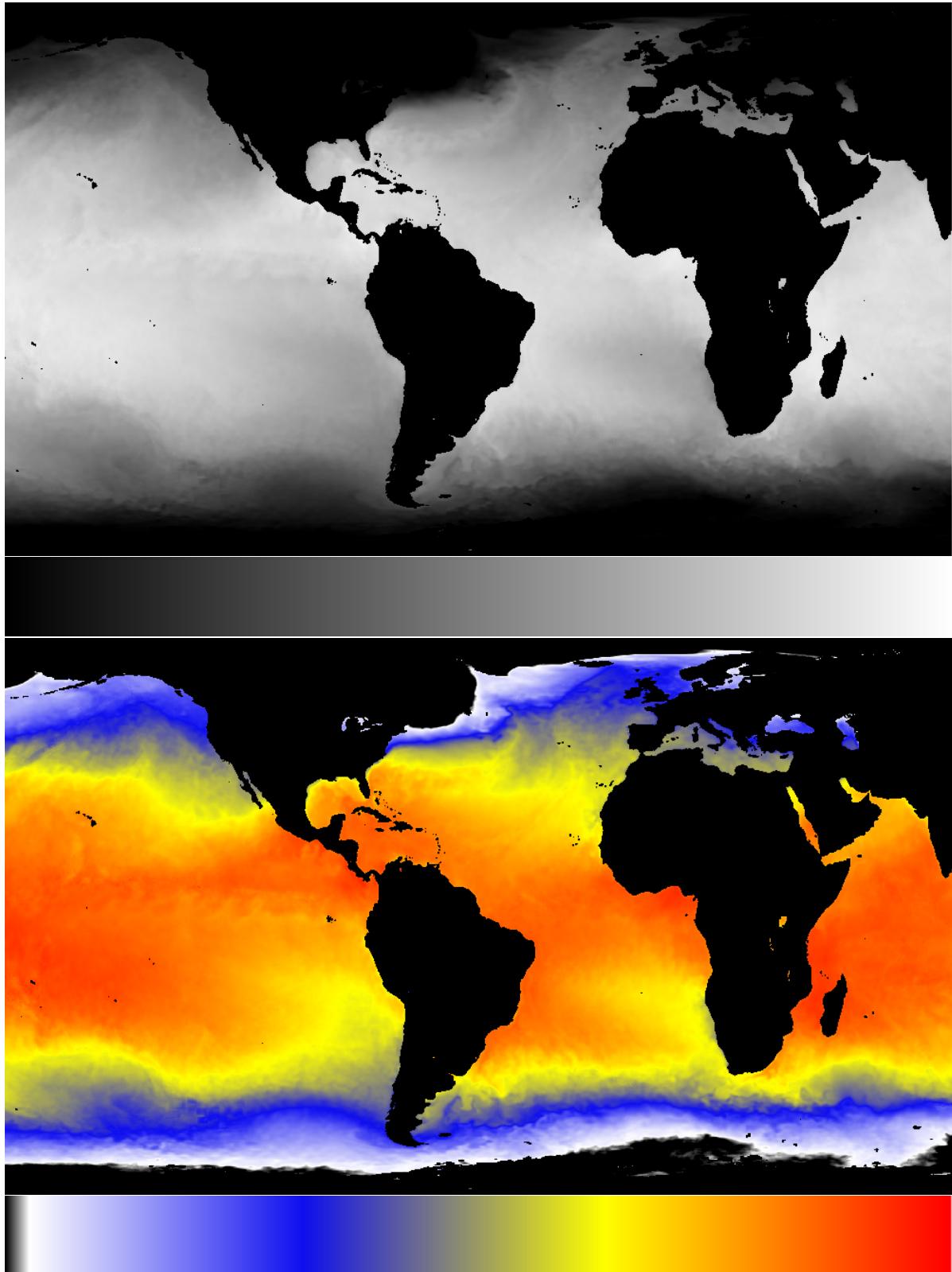


Figure 1.1: Representation of the same sea surface temperature flux data using two different color palettes: one with five nodes (black, white, blue, yellow, and red) and another using only a grayscale gradient from black to white. [Buj23] [NAS]

Beyond simply exploring images, regardless of the chosen color spaces, users in the field of volume rendering often wish to preserve and share the insights gained from refining transfer functions. Sharing these results, while also incorporating the dynamic appeal of movement and immersion, can greatly benefit a wide range of domains. To support this need, and in collaboration with Fraunhofer, we developed a plugin for their application that generates videos based on user-defined camera trajectories. The primary goal is to create seamless and visually engaging cut-scenes for presenting rendered volumes. This task, however, was far from simple. Numerous factors had to be considered: tracing smooth and accurate camera trajectories, dealing with transfer functions blending, interpolating camera orientations, and ensuring the tool accommodated a broad set of user requirements. The development process raised the next research questions:

(Q2) How can interpolation between control points for camera position and orientation ensure smooth and visually appealing animation?

- (Q2.1) What methods support intuitive specification of the camera's trajectory and orientation along the path?
- (Q2.2) How can pause frames be effectively encoded as control points within a trajectory definition framework?
- (Q2.3) What techniques are best suited for interpolating between distinct camera orientations to maintain continuity and visual appeal?

(Q3) How can transfer function interpolations be executed to produce smooth, semantically coherent, and visually appealing animations?

Ensuring continuous and visually appealing transitions is particularly important when shifting the region of interest, as it helps observers maintain context. Smooth, subtle transitions allow users to follow changes without abrupt shifts, preserving coherence in the visualization. This technique is widely employed in immersive analytics to enhance user engagement and reduce the risk of disorientation or motion sickness. Beyond camera movement, transfer functions can also be dynamically adjusted during animations, unlocking a whole new range of possibilities. By adapting color mappings and opacity throughout the animation, the visualization can smoothly highlight new regions of interest while maintaining visual consistency across frames.

1.2 Structure of the Thesis

In this thesis, we begin with the **Background** chapter. It introduces the foundational knowledge necessary to understand the key concepts addressed throughout this work. This includes the representation of orientations in 3D space, an overview of trajectory formation with an emphasis on smooth curve creation, and a discussion of color theory, with a focus on color spaces transformations. The chapter concludes with how volumetric data is typically rendered, as well as how our framework handles it.

After establishing the necessary background, we present the **Related Work** chapter, where we review related literature at the intersection of transfer functions in direct volume rendering as well as trajectory planning. In particular, we examine advancements in color appearance models and discuss both objective and subjective methods for evaluating the quality of color palettes, a crucial element in transfer functions, which directly map volume data to optical properties.

Next, the **Methodology** chapter details the techniques employed and challenges encountered in developing a smooth and visually appealing camera trajectory. We also introduce the use of perceptually uniform color spaces in volume rendering. This chapter brings together these elements into the creation of a comprehensive video export framework capable of interpolating both transfer functions and camera movements.

The **Results** chapter provides a comprehensive overview of the user study methodology employed to evaluate the use of transfer functions with perceptually uniform color spaces, as well as our video export framework. It details in-depth the design and execution of the studies, including the procedures followed during the evaluation phases. Furthermore, the chapter presents an in-depth analysis of the results obtained, highlighting key findings and insights derived from participants' feedback and performance.

Subsequently, the **Discussion** chapter provides an overview and broader analysis of the results, addressing the evaluation conclusions of the video export framework as well as the role of perceptually uniform color spaces in enhancing volume rendering.

Finally, the **Conclusion** chapter summarizes the insights gained from this work, directly addresses the research questions, acknowledges its limitations, and outlines potential directions for future research.

1.3 Main Contributions

Several important contributions are made to enhance the visualization of volumetric data. These contributions address key challenges in color representation and camera motion. The core contributions of this thesis are as follows:

- The exploration of perceptually uniform color spaces in volume rendering.
- The conceptualization and implementation of a framework that enables smooth and natural camera movements in volumetric data rendering.
- The definition of transfer function keyframes and their interpolation in the framework, allowing dynamic changes to the transfer function over time.
- An exploration of multiple techniques for interpolating camera orientation, accompanied by a preliminary evaluation of their effectiveness.
- An evaluation of the impact and quality of perceptually uniform color spaces on volume-rendered images.
- The contribution of a new implementation of the ProLAB color appearance model within the VTK open-source visualization framework.

In this work, we explicitly **do not** claim that perceptually uniform color spaces are universally superior to traditional color spaces for all volume rendering tasks. Additionally, we do not provide a fully quantitative or statistically validated user study. The evaluations conducted are preliminary and exploratory in nature.

2 Background

The eye sees only what the mind is prepared
to comprehend.

Robertson Davies, *The Manticore*

To understand the methods and techniques introduced in this work, we discuss in this chapter rotation and path representations with the goal of accurately defining, computing, and interpolating camera positions. Next, we explore various color space representations and the transformations between them, alongside the key domain-specific parameters essential for implementing these spaces in practical applications. Finally, we conclude by explaining the fundamentals of volume rendering as it is commonly applied in visualization systems.

2.1 Orientation

An object in space is typically represented by its position relative to an origin and its rotation. Rotation can be expressed using various methods, including axis-angle, Euler angles, or quaternions. While axis-angle and Euler angles provide an intuitive way to describe rotations using direct transformations, they lack robustness and become impractical for complex applications such as interpolation. One major issue with Euler angles is gimbal lock, where a specific configuration causes the loss of one degree of freedom, restricting the ability to represent certain rotations.

Quaternions were introduced to address the limitations of traditional rotation representations, particularly those based on Euler angles. While Euler angles are intuitive and easy to understand, they are prone to issues such as gimbal lock, which occurs when two of the three rotational axes become aligned, leading to a loss of one degree of freedom

2 Background

and resulting in unpredictable or constrained motion. Quaternions provide a more stable and efficient alternative for handling rotations, especially in 3D graphics and animation. Represented as four-dimensional numbers comprising one real part and three imaginary components 2.1, quaternions enable smooth and continuous interpolation between orientations.

$$Q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} \quad (2.1)$$

When working with quaternions, it is essential to understand that classical mathematical operations extend beyond the scope of standard arithmetic. Operations such as addition (2.1), multiplication (2.3), conjugation (2.4), and normalization (2.5) each follow specific mathematical rules unique to quaternion algebra. These rules often differ significantly from those used in conventional 3D transformations, which means that properly applying quaternions requires a solid grasp of their distinct properties to ensure accurate and stable rotational computations.

- Addition :

$$Q_1 + Q_2 = (a_1 + a_2) + (b_1 + b_2)\mathbf{i} + (c_1 + c_2)\mathbf{j} + (d_1 + d_2)\mathbf{k} \quad (2.2)$$

- Hamilton product :

$$\begin{aligned} Q_1 Q_2 &= a_1 a_2 + b_1 b_2 + c_1 c_2 + d_1 d_2 \\ &\quad + (a_1 b_2 + b_1 a_2 + c_1 d_2 - d_1 c_2)\mathbf{i} \\ &\quad + (a_1 c_2 - b_1 d_2 + c_1 a_2 + d_1 b_2)\mathbf{j} \\ &\quad + (a_1 d_2 + b_1 c_2 - c_1 b_2 + d_1 a_2)\mathbf{k} \end{aligned} \quad (2.3)$$

- Conjugate :

$$Q^* = a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k} \quad (2.4)$$

- Norm :

$$\|Q\| = \sqrt{QQ^*} \quad (2.5)$$

2.2 Position Interpolation

Defining trajectories can be a challenging task, especially when constraints are imposed. Without restrictions, the degrees of freedom are infinite, for instance, an infinite number of lines can pass through two given points. To narrow down the possibilities and remain relevant to the intended use case, one common approach is to define the trajectory as the shortest path between the two points. In Euclidean space, this corresponds to a straight line.

A path can be created by linking a sequence of two-point line segments. However, this method can be restrictive, particularly when smooth transitions are needed between segments. Using only straight lines often leads to sharp changes in direction, which can break the sense of immersion in applications such as camera path planning. To address this issue, splines extend the concept of straight lines by using piecewise-defined polynomials. These allow greater control over the trajectory, as higher-degree polynomials and auxiliary points enable smoother curves. To ensure a continuous and natural flow constraints such as continuity of position, first derivative (tangency), and second derivative (curvature) can be imposed. These constraints help eliminate abrupt transitions and create fluid motion at the cost of more defined points.

To control the shape of a curve within a path segment, additional points can be introduced between the start and end positions. The number of these points is related to the degree of the curve, which in turn influences its final form. To construct such curves, Bézier curve algorithms can be used. But how exactly do these algorithms work? Consider for example a quadratic Bézier curve defined by three points: P_0 , P_1 , P_2 . The curve is generated through a sequence of linear interpolations: First, we interpolate linearly between P_0 and P_1 and between P_1 and P_2 , using a parameter t in the range $[0, 1]$. This results in two new points, Q_0 and Q_1 , given by the Equations in 2.6.

$$Q_0 = (1 - t)P_0 + tP_1 \quad Q_1 = (1 - t)P_1 + tP_2 \quad (2.6)$$

Next, we interpolate linearly between Q_0 and Q_1 using the same parameter t , producing a final point B , as shown in Equation 2.7.

$$B = (1 - t)Q_0 + tQ_1 \quad (2.7)$$

2 Background

As the parameter t varies continuously from 0 to 1, the trajectory of point B traces out the path of a quadratic Bézier curve. This elegant curve smoothly transitions between the control points, creating a visually appealing and mathematically well-defined shape. Figure 2.1 clearly illustrates this behavior, showing how the curve is generated by blending the positions of the control points based on the value of t .

What makes this formulation particularly interesting is the underlying mathematical structure, which is expressed in terms of Bernstein polynomials. This polynomial basis not only standardizes the representation of Bézier curves but also offers several practical advantages. Notably, Bernstein polynomials are known for their computational efficiency [KK24], allowing for quick and stable evaluation of the curve at any point t . Furthermore, their properties lend themselves well to graphical interpretation, making the analysis and visualization of Bézier curves intuitive and straightforward.

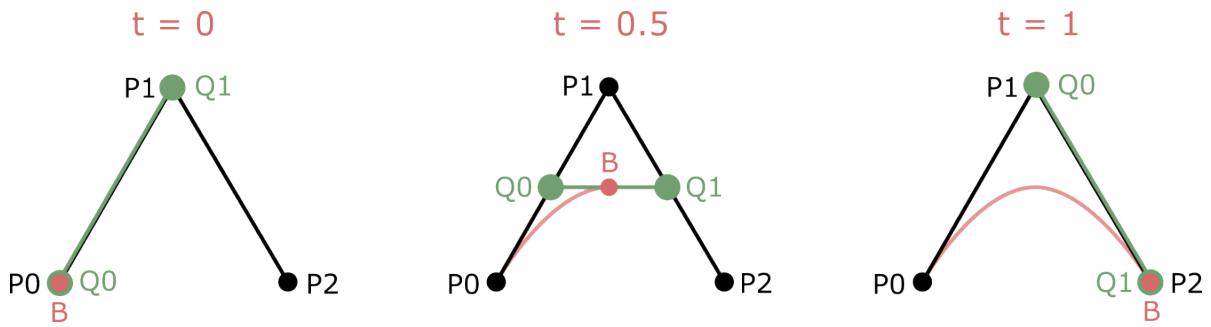


Figure 2.1: Illustration of the construction process of a quadratic Bézier curve.

The Bézier curves generation algorithm naturally extends to higher-degree Bézier curves by recursively applying linear interpolation across a greater number of control points. However, if the control points represent actual points that the curve must pass through, this approach becomes inappropriate. This case can particularly be seen when users define points of interest in a 3D space that the camera trajectory must follow. In such cases, manually placing additional intermediary control points to adjust the final form of the Bézier curve is neither ergonomic nor practical for the end user.

To meet the practical and visual requirements, we can introduce rules that ensure smooth transitions, pass through all control points, and avoid sharp edges. Smooth transitions are guaranteed by the continuity of the curve, while the absence of sharp edges is achieved by the derivative continuity of the curve. The conditions mentioned are met in the Catmull-Rom [CR74] cubic interpolation. Catmull-Rom interpolation utilizes four

2 Background

points: two central points (P_1, P_2), referred to as the start and end points, and two extreme points (P_0, P_3), acting as directional control points, as illustrated in Figure 2.2.



Figure 2.2: Illustration of the Catmull-Rom construction of the speed vectors V_1 and V_2 .

By examining the relationship between a directional control points and its neighboring points, we derive a property analogous to speed, which helps maintain a certain momentum when transitioning from one spline segment to another. Ideally, when selecting the next set of four points used to generate a curve segment, the resulting path should smoothly carry the shape and direction from the end of the previous segment into the beginning of the next one. This continuity is mathematically ensured by defining the derivative at the transition point (V_1, V_2), referred to as speed vectors, which are computed as shown in Equation 2.8.

$$V_1 = \frac{(P_1 - P_0) + (P_2 - P_1)}{2} \quad V_2 = \frac{(P_2 - P_1) + (P_3 - P_2)}{2} \quad (2.8)$$

When the two speed vectors are properly defined in conjunction with the defined central points, Hermite interpolation can be effectively applied to generate a smooth transition between curve segments. This interpolation method guarantees both positional and derivative constraints, ensuring that the resulting curve not only passes through the specified points but also respects the defined tangents at those points. By satisfying these constraints, Hermite interpolation produces visually coherent and fluid motion, also in camera path animations. This mathematical formulation allows for greater control over the shape and smoothness of the curve, making it particularly suitable for applications requiring continuous trajectory planning. The relationship governing this interpolation is given by:

$$\begin{aligned} P_1 &= f(x_1) & P_2 &= f(x_2) \\ V_1 &= \frac{df(x_1)}{dx} & V_2 &= \frac{df(x_2)}{dx} \end{aligned} \quad (2.9)$$

By using these four constraints specified in Equation 2.9, which typically involve posi-

2 Background

tions and derivatives at the endpoints, we can construct a third-degree polynomial that satisfies the given conditions. Solving this system of equations allows us to isolate the values of P_1 , P_2 , V_1 , and V_2 , ultimately producing the coefficients of a cubic polynomial that accurately defines the interpolated path. This isolation is shown in Equation 2.10.

$$f(x) = P_1(2x^3 - 3x^2 + 1) + P_2(3x^2 - 2x^3) + V_1(x^3 - 2x^2 + x) + V_2(x^3 - x^2) \quad (2.10)$$

In summary, the key formulations that will serve as the foundation for our interpolation framework are presented in Equation 2.11. These expressions, derived from Equations 2.8 and 2.10, establish a consistent mathematical basis for computing smooth transitions between camera positions and orientations. By clearly defining both velocity and position constraints, they enable robust and flexible path interpolation, which will be referenced throughout the remainder of this thesis.

$$\text{Catmull-rom}(t, P_0, P_1, P_2, P_3) = \text{Hermite}(t, P_1, P_2, V_1, V_2) \quad (2.11)$$

where :

$$\begin{aligned} \text{Hermite}(t, P_1, P_2, V_1, V_2) = & P_1(2t^3 - 3t^2 + 1) + P_2(3t^2 - 2t^3) + \\ & V_1(t^3 - 2t^2 + t) + V_2(t^3 - t^2) \end{aligned} \quad (2.12)$$

In this formulation, the parameter t ranges from 0 to 1. Figure 2.3 demonstrates the behavior and construction of a curve generated through Hermite interpolation. The black points (P_1 , P_2) represent the start and end positions of the curve, while the blue points (P_0 , P_3) serve as control points that influence the smoothness and shape of the interpolation. The tangents V_1 and V_2 , which guide the curvature at P_1 and P_2 , are computed based on Equation 2.8.

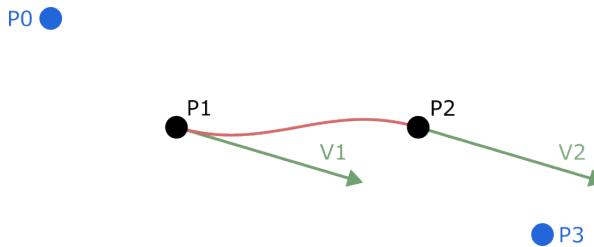


Figure 2.3: Construction of Catmull-Rom interpolation curves by deriving the intermediate variables V_1 and V_2 used in Hermite interpolation.

It is important to note that when applying all the spline constructions directly to real-world metrics and positions, one factor that can disrupt immersion is speed. Specifically, speed is not constant between segments. Since t varies from 0 to 1 regardless of the segment's length, both short and long segments will take the same amount of time to traverse. Achieving constant speed traversal is not a trivial task. However, there are approximations that can be used to create the perception of a constant speed throughout the spline path. One such approximation involves traversing the spline, in our case the Catmull-Rom spline, measuring the distance along the path, and then using a simple conversion to estimate the expected time spent on each segment. These time values are then weighted based on segment length to ensure a consistent global speed across the entire trajectory.

2.3 Color Spaces

Color representation on screens is typically achieved through the combination of three primary colors: Red, Green, and Blue. This choice is based on the human visual system, where the retina contains cones sensitive to these particular wavelengths of light. To align with human anatomy and the way we perceive colors, a common method for encoding colors is within a 3D Euclidean RGB color space, where each component (R, G, and B) represents a percentage of its respective color. However, this color representation is not universally consistent across devices, leading to variations in how colors are displayed on different supports. To address these limitations and provide a more standardized approach to color representation, the Commission Internationale de l'Éclairage (CIE) developed the CIERGB color space through experimental research. Their goal was to unify color display behavior across different devices and establish a consistent color representation.

Additionally, to simplify the mathematical handling and communication of color, the Commission Internationale de l'Éclairage (CIE) introduced the CIEXYZ color space [SG31], which provides a standardized and device-independent model for representing color. This model was a significant step forward in color science, as it enabled consistent color specification across different devices and media. In the CIEXYZ space, the X component primarily represents hue, Y corresponds to luminance (or perceived brightness), and Z accounts for aspects related to saturation. The values within this

2 Background

space were empirically derived through psychophysical experiments, where participants adjusted proportions of the red, green, and blue primaries to visually match a test color. These matches were then used to construct mathematical functions that map the spectral composition of CIERGB into the CIEXYZ space. As a result, this model serves as a foundational reference in colorimetry and is frequently used as an intermediate color space for conversions between other color models.

However, either the CIERGB or CIEXYZ color space have been found to lack perceptual uniformity. A perceptually uniform color space is one in which equal distances along any axis correspond to equal perceived differences in color by the human eye. This means that a shift of the same magnitude in any direction should appear equally noticeable. Achieving true perceptual uniformity is highly challenging due to the complex nature of human color perception, which is influenced by numerous factors such as the characteristics of the display device, surrounding background, ambient lighting conditions, and human variations in visual sensitivity. Consequently, while several modern color spaces aim to approximate perceptual uniformity, such as CIELAB or CIECAM-based models, none achieve perfect uniformity in all conditions. These spaces still offer significant practical advantages by reducing perceptual distortions and enabling more accurate color manipulation, even though they remain theoretical approximations rather than flawless models.

One of the earliest color spaces to aim for perceptual uniformity was CIELUV [Ada43]. Built upon the CIEXYZ color space, it departs from strict linearity to offer a more perceptually accurate representation of color differences. A key feature of CIELUV is its ability to define color distances in a way that more closely aligns with how humans perceive color variation, thus providing a more uniform response to changes in hue, brightness, and saturation. Additionally, these color models introduce the concept of standard illuminants, which represent typical lighting conditions based on correlated color temperature. Among them, illuminant D65 is the most widely adopted, particularly for RGB displays and general daylight simulations. In general color spaces conversions, this standard is used as the reference.

Following CIELUV, the next significant attempt at creating a perceptually uniform color space is CIELAB. While it involves more complex and computationally expensive transformations, CIELAB is widely regarded as the standard for detecting small differences in color. This color space represents color in three components: L , which indicates luminance, and a and b , which represent the opponent color pairs of human vision (red-green

and blue-yellow, respectively). The goal of CIELAB is to provide a more uniform perceptual scale, where the perceptual difference between two colors is approximately the same regardless of the region of the color space. However, while CIELAB offers significant improvements in perceptual uniformity over earlier color spaces, it is not perfectly uniform. Despite this limitation, it remains a standard for many color-related applications, particularly in the context of measuring and comparing subtle color differences.

The detailed formulae for color space conversions used in this work are provided in Appendix A. These transformations originate from the well-established color science resource maintained by Bruce Lindbloom [Lin17], specifically under the "Math" section of the website. For the purposes of this thesis, these mathematical models have been carefully adapted and refined to suit the specific requirements of digital volume rendering.

2.4 Color Distances

It would be mathematically useful to have a straightforward method for measuring the difference between two colors. While several approaches have been proposed to address this problem, it remains a complex task in practice. One of the key challenges lies in taking into account how humans actually perceive color differences. But what does it truly mean to measure the distance between one color and another? Ideally, such a distance should reflect the degree to which the two colors appear different to the human eye. However, due to the nonlinear and context-sensitive nature of human color perception, defining a perceptually accurate color distance is far from trivial.

A straightforward and commonly used approach to quantify the difference between two colors is to calculate the squared difference for each individual component in the CIELAB color space, which includes the lightness (L), and the color-opponent dimensions (a and b). These squared differences are summed together, and then the square root of this sum is taken to provide a single numerical value that represents the overall distance between the two colors. This method is known as the CIE 1976 color difference formula, often denoted as $\Delta E_{CIE1976}$ and is expressed in Equation 2.13.

$$\Delta E_{CIE1976} = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2} \quad (2.13)$$

2 Background

However, this method is not fully aligned with human perception. In other words, two colors that are mathematically equidistant from a reference color may not be perceived as equally different by the human eye. To address this limitation, more sophisticated and perceptually accurate color difference formulae have been developed. These methods derive intermediate parameters that better reflect the nuances of human color perception, resulting in differences that are more uniform perceptually. For standard display conditions, one such improved formula is expressed in Equation 2.14, with a detailed derivation of its variables provided in Appendix B.1.

$$\Delta E_{CIE1994} = \sqrt{(\Delta L)^2 + \left(\frac{\Delta C}{1 + 0.045C_1}\right)^2 + \left(\frac{\Delta H}{1 + 0.015C_1}\right)^2} \quad (2.14)$$

The previous formula still does not fully capture the complexities of human color perception. To overcome these limitations, the CIEDE2000 color difference formula was introduced and is now regarded as the current standard for perceptually coherent color distance measurements. This advanced formula improves upon earlier methods by incorporating not only lightness, chroma, and hue components but also an interactive term that models the interplay between chroma and hue [LCR01]. Consequently, it offers a significantly more accurate representation of perceived color differences, better aligning with human visual sensitivity. The formula is given in Equation 2.15, with a comprehensive derivation of its variables detailed in Appendix B.2.

$$\Delta E_{CIE2000} = \sqrt{\left(\frac{\Delta L'}{S_L}\right)^2 + \left(\frac{\Delta C'}{S_C}\right)^2 + \left(\frac{\Delta H'}{S_H}\right)^2 + R_T\left(\frac{\Delta C'}{S_C}\right)\left(\frac{\Delta H'}{S_H}\right)} \quad (2.15)$$

With all the relevant color distance formulas and color spaces now defined, another fundamental concept in color perception to consider is the Just Noticeable Difference (JND). The JND is formally defined as the smallest change in a stimulus that can be reliably detected by an observer at least 50% of the time. To illustrate, consider two RGB colors: (255, 0, 0) and (252, 0, 0). For most people, this slight difference is below the perceptual threshold and thus appears indistinguishable. In contrast, comparing (255, 0, 0) with (200, 0, 0) produces a color difference that is clearly noticeable to an average observer without color vision deficiencies. This notion of perceptual thresholds is visually represented in Figure 2.4, highlighting the subtle boundary where color differences become perceptible.

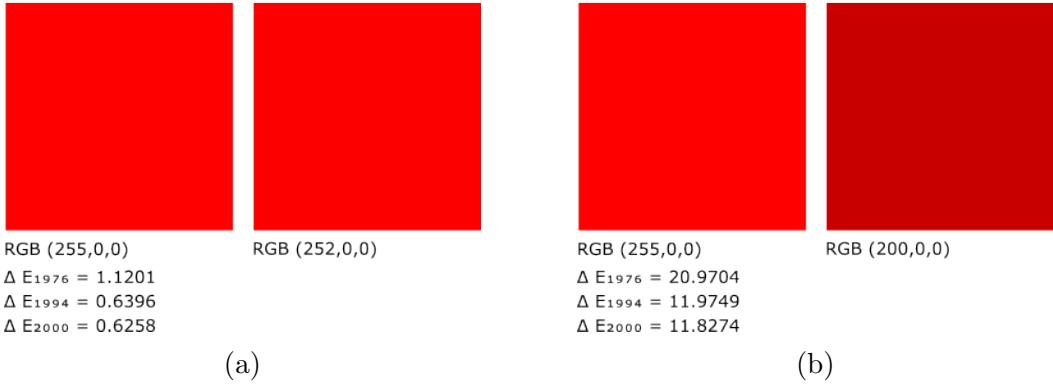


Figure 2.4: Just Noticeable Difference (JND) example: (a) Comparison of two red tones where the difference is not perceptible; (b) Comparison of two red tones where the difference is noticeable.

That said, through empirical experimentation, most of the color difference formulas, with the exception of ΔE_{1976} , have originally been designed such that values greater than 1 correspond, by definition, to perceptual differences that exceed the just noticeable difference (JND). In practical terms, this means that color distances exceeding 1 should be perceptible to the average human observer. Since the formal notion of JND was introduced after the definition of ΔE_{1976} , it was later discovered that the JND threshold varies with this color metric. Consequently, for CIELAB, a ΔE_{1976} value of 2.3 or higher indicates a perceptible color difference [Gau17].

2.5 Volume Rendering

Volume rendering is a set of techniques of data visualization that focuses on exploring volumetric data to generate images. In this thesis, we concentrate on direct volume rendering using transfer functions. Volumetric data is acquired using imaging technologies such as MRI and X-rays devices, which typically assign a scalar value to each precise point in space. These points, analogous to pixels in 2D space, are referred to as voxels in 3D. With this grid of points, various algorithms can be employed to transform abstract raw data into visually appealing 3D renders. The most common approach is ray marching. Simply put, multiple rays are cast from the camera and iterated step by step to construct the final image.

2 Background

Multiple approaches can be used in raymarching to generate the final rendered image. One common method is maximum intensity projection, which highlights the brightest values along the ray and is often used to enhance the detection of features such as cancers. Another approach involves accumulation with compositing, where contributions from all visible materials along the ray are combined to produce a more comprehensive image. Averaging can also be applied to simulate X-ray-like visuals by blending all sampled values uniformly. Additionally, the first hit method captures surfaces by taking the first intersection along the ray, providing an exterior view of the object. It is important to note that these are not exhaustive, and other techniques exist to convert ray information into meaningful visual representations. A summary of these transformation methods is illustrated in Figure 2.5.

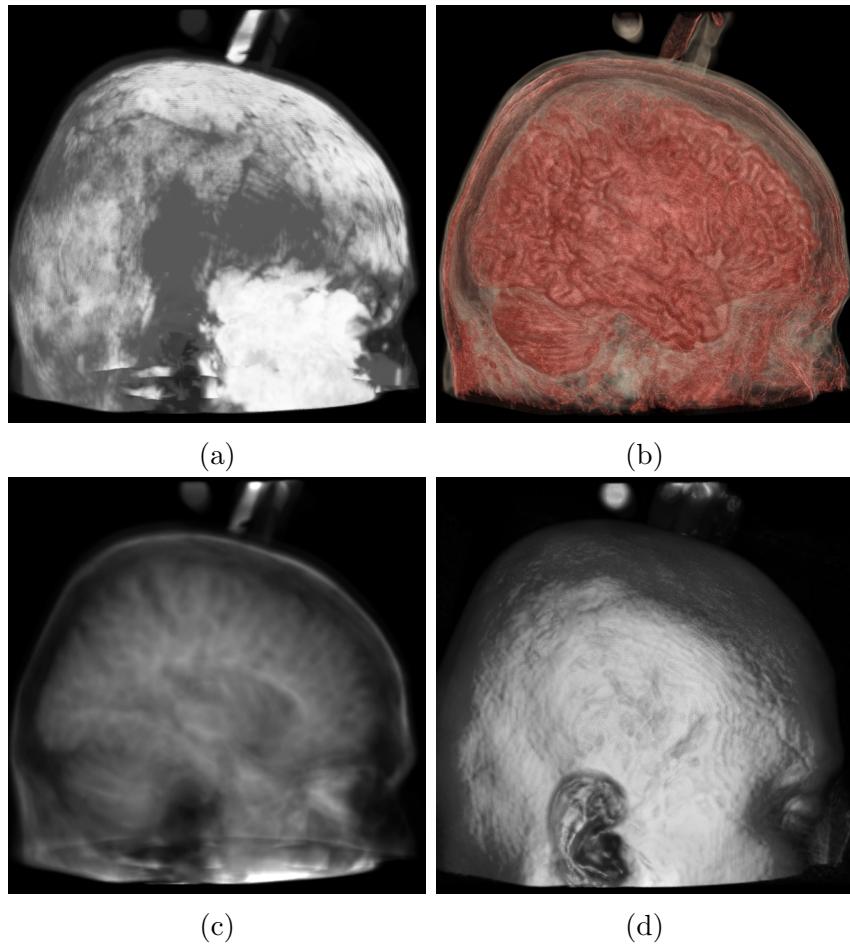


Figure 2.5: Illustration of four volume rendering methods applied to CT scan data from the VTK Examples Dataset [VTK25] using raymarching: (a) maximum intensity projection, (b) compositing accumulation, (c) averaging, and (d) first-hit detection.

2 Background

For this thesis, the focus will be on the alpha compositing technique used in volume rendering, specifically the mapping function that links scalar values to colors, which is crucial for generating the final 3D image. Before diving deeper into transfer functions, it is important to understand how the colors along each ray are obtained through alpha compositing. Typically, the colors and opacities are processed iteratively from front to back to optimize the blending process. When the accumulated alpha reaches 1 (fully opaque), the iteration stops, and the blended color is finalized. This process is mathematically expressed in Equations 2.16.

$$\begin{aligned} C'_i &= C'_{i+1} + (1 - \alpha'_{i+1})C_i \\ \alpha'_i &= \alpha'_{i+1} + (1 - \alpha'_{i+1})\alpha_i \end{aligned} \tag{2.16}$$

where C_i and α_i represent the color and opacity emitted by the current sample along the ray, and C'_i and α'_i denote the accumulated color and opacity observed during ray traversal.

During the iterative raymarching process, a transfer function is applied to convert voxel values into optical properties, typically assigning each voxel a color and its corresponding opacity (C_i, α_i). These optical values are accumulated along the ray's path to compute the final color for each ray. By casting rays for every pixel on the screen plane, a fully rendered 3D volume image is produced. Additionally, lighting effects such as Phong shading [Bui73] can be incorporated by using the ray directions and gradient information to simulate how light interacts with the volume, enhancing depth and surface perception. More advanced raymarching techniques may introduce secondary rays to simulate complex phenomena such as reflections, refractions, and occlusions. However, these effects are generally excluded from standard volume rendering pipelines because of their high computational cost, increased complexity, and limited necessity for most visualization applications.

There are various types of transfer functions, as discussed in the comprehensive state-of-the-art review by Ljung et al. [Lju+16]. The simplest among them directly map voxel values to optical properties based on predefined mappings and are commonly referred to as 1D transfer functions in the literature. When users define the mapping, the traditional approach involves a 2D chart, where the horizontal axis represents scalar values ranging from the minimum to the maximum of the volumetric dataset, while the vertical axis corresponds to the opacity of control points. Additionally, each control point is assigned

2 Background

a specific color, enabling intuitive customization of the rendering process. With this setup, users can efficiently explore and generate visually appealing 3D renders with reasonable effort [Lju+16]. To aid in identifying regions of interest, a histogram is typically displayed at the bottom of the chart, synchronized with the color palette. This histogram is often presented on a logarithmic scale to enhance the visibility of significant peaks. An illustration of this widget is shown in Figure 2.6.

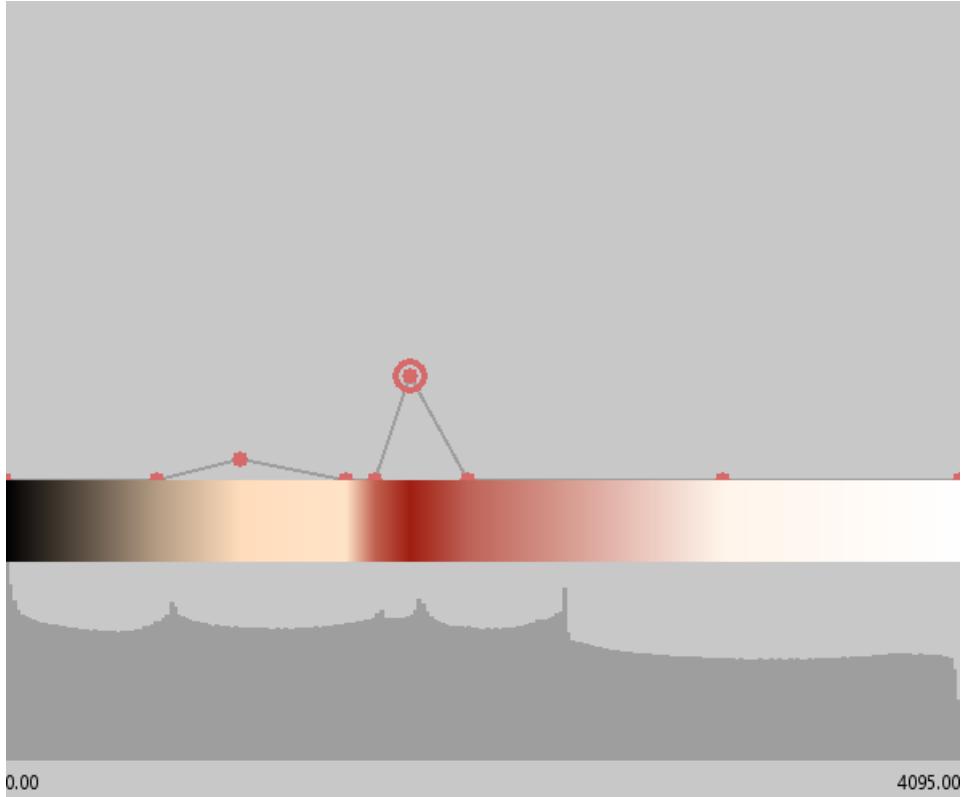


Figure 2.6: Illustration of a 1D transfer function widget. The horizontal axis represents scalar values ranging from the minimum to the maximum of the volumetric dataset, while the vertical axis corresponds to opacity varying from 0 to 1.

However, as noted in the state of the art by Ljung et al. [Lju+16], 1D transfer functions are not well suited for certain use cases, particularly when multiple regions of interest overlap within the scalar range of volumetric data, such as in medical scans. This overlap makes it difficult to isolate specific structures for clear visualization, thereby limiting the effectiveness of 1D transfer functions in these scenarios. To address this limitation, More complex transfer functions first transform the volumetric data to incorporate multiple dimensions, aiming for a more precise mapping of values. Among these transformations,

the gradient magnitude is the most commonly used, as it helps identify regions of transition within the volume. These advanced mappings can be used together with scalar values representing 2D transfer functions in the literature. A 2D transfer function is typically represented as a 2D histogram, where the x-axis corresponds to scalar values and the y-axis represents the derived values from the scalar ones. For the construction of each pixel in this histogram a gray color is assigned based on the logarithmic count of occurrences of the corresponding scalar-gradient pair, aiding in identifying material boundaries. This widget is illustrated in Figure 2.7.

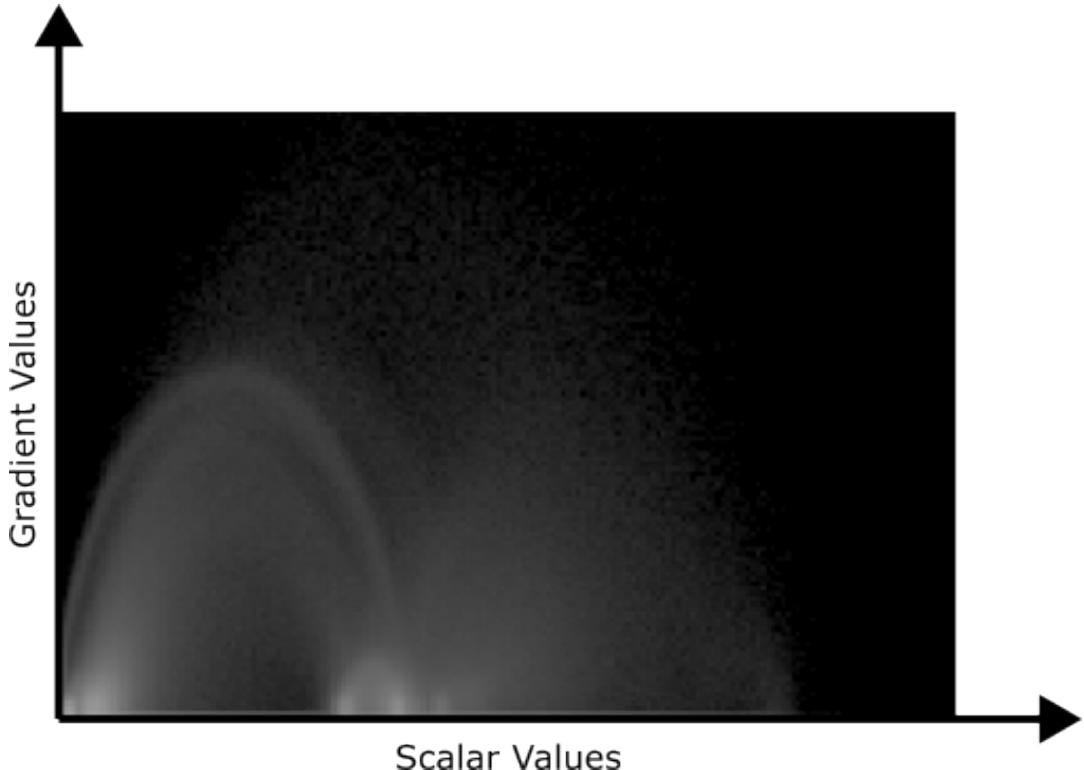


Figure 2.7: Illustration of the 2D transfer function widget using a CT scan dataset from the VTK Examples collection [VTK25]. The horizontal axis represents the range of scalar values of the volume data, while the vertical axis represents the range of the gradient magnitude. The gray intensity of each pixel is logarithmically proportional to the number of voxels in the volume that match the corresponding scalar-gradient pair.

Note that points at the bottom of the chart correspond to low gradient values, representing regions with similar scalar values. For example, if areas of high intensity appear at the bottom of the histogram, this may suggest the presence of homogeneous materials or different materials sharing similar scalar properties. Another feature of 2D transfer functions is the presence of arcs, which indicate regions where surface changes occur.

2 Background

The higher an arc is on the y-axis, the more significant the changes in the surface are. This allows the spot of surface boundaries on data, helping to better distinguish regions of interest based on their surrounding gradients.

Following user interaction, specific regions can be selected and assigned colors, which are then used as input for the ray marching algorithm to generate the final 3D rendering. In the literature [Lju+16], the selection of these regions ranges from basic faded-edge squares to more sophisticated approaches, such as clustering techniques aimed at highlighting areas of interest and producing visually engaging images. This technique ultimately results in a mapping between scalar values and their derived forms to corresponding colors, including associated opacity levels.

3 Related Work

Thus the task is not so much to see what no one yet has seen, but to think what nobody yet has thought about that which everybody sees.

Arthur Schopenhauer

In this chapter, we introduce the concept of color appearance models with CIELAB, one of the earliest and most commonly used models, while critically discussing its limitations. We explain how transformations from standard color spaces to the ProLAB color space are performed and briefly outline the construction of it. The chapter then details the mathematical foundations of perceptual color metrics relevant to this thesis, especially in the context of data visualization. We establish consistent naming conventions and define key concepts such as color distance, perceptual speed, and Just Noticeable Differences, addressing it both locally and globally. Finally, we explore how these perceptual metrics can be applied within transfer functions to enhance direct volume rendering and improve the visualization of volumetric data.

3.1 Color Appearance Models

The term "color appearance models" refers specifically to the way colors are perceived by human observers. Unlike general color models, which aim to mathematically represent colors in a structured space, color appearance models account for the effects of external factors that influence color perception. These factors may include ambient lighting, the surrounding environment, and the physiological characteristics of the observer. As seen in previous examples of color models, such as CIERGB, CIEXYZ, and CIELAB, each

3 Related Work

offers different approaches to representing color. Among them, CIELAB is considered one of the earliest and most basic forms of a color appearance model. It makes an initial attempt to account for external conditions, for instance by incorporating a reference white point to approximate the effect of illumination.

Recent models have been proposed that aim to improve perceptual uniformity while offering greater simplicity, speed, or efficiency, as demonstrated in the works of Li et al. [Li+17] and Konovalenko et al. [Kon+21]. ProLAB, one of these models go beyond merely modeling the physical properties of color and instead emphasize how colors are perceived under the influence of external factors. The conversion from CIEXYZ to ProLAB is achieved through a straightforward series of matrix multiplications and projections. These transformations are expressed in the Equations 3.1 and 3.3.

$$\begin{bmatrix} L \\ a \\ b \end{bmatrix} = \begin{bmatrix} \frac{L_{homog}}{w} \\ \frac{a_{homog}}{w} \\ \frac{b_{homog}}{w} \end{bmatrix} \quad \begin{bmatrix} L_{homog} \\ a_{homog} \\ b_{homog} \\ w \end{bmatrix} = Q \cdot \begin{bmatrix} \frac{X}{0.9505} \\ \frac{Y}{1} \\ \frac{Z}{1.089} \\ 1 \end{bmatrix} \quad (3.1)$$

where :

$$Q = \begin{bmatrix} 75.54 & 486.66 & 167.39 & 0.0 \\ 617.72 & -595.45 & -22.27 & 0.0 \\ 48.34 & 194.94 & -243.28 & 0.0 \\ 0.7554 & 3.8666 & 1.6739 & 1.0 \end{bmatrix} \quad (3.2)$$

Note that the X , Y , and Z components in the CIEXYZ color space are transformed based on the D65 standard illuminant. This illuminant represents average daylight conditions with a correlated color temperature of approximately 6500 Kelvin and serves as a widely accepted reference white point in the fields of colorimetry and color appearance modeling. By adopting D65 as the standard, color representations aim to simulate how colors appear under natural daylight, ensuring consistency and accuracy in visual perception across different devices and viewing environments.

The inverse transformation from ProLAB to CIEXYZ is accomplished using a corresponding inverse matrix operation. This allows for accurate and seamless bidirectional conversion between the two color spaces, ensuring that color data can be translated back

3 Related Work

and forth without significant loss of fidelity. Such reversibility is essential for maintaining consistency in applications that require both perceptual adjustments and compatibility with standard colorimetric models. In practical terms, this means we perform interpolation within the ProLAB space, taking advantage of its perceptual uniformity, and then convert the results to CIERGB for accurate visual display.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.9505X^* \\ 1.0000Y^* \\ 1.0890Z^* \end{bmatrix} \quad \begin{bmatrix} X^* \\ Y^* \\ Z^* \end{bmatrix} = \begin{bmatrix} \frac{X_{homog}}{w} \\ \frac{Y_{homog}}{w} \\ \frac{Z_{homog}}{w} \end{bmatrix} \quad \begin{bmatrix} X_{homog} \\ Y_{homog} \\ Z_{homog} \\ w \end{bmatrix} = Q^{-1} \cdot \begin{bmatrix} L \\ a \\ b \\ 1 \end{bmatrix} \quad (3.3)$$

where:

$$Q^{-1} = \begin{bmatrix} 0.0014 & 0.0014 & 0.0008 & 0.0 \\ 0.0014 & -0.0002 & 0.0010 & 0.0 \\ 0.0014 & 0.0001 & -0.0032 & 0.0 \\ -0.0086 & 0.0002 & 0.0010 & 1.0 \end{bmatrix} \quad (3.4)$$

The matrix Q and its inverse Q^{-1} , which define the transformation between the CIEXYZ and ProLAB color spaces, are composed of optical parameters obtained through a detailed numerical optimization process. These matrices were introduced and empirically validated in the original work by Konovalenko et al. [Kon+21]. The optimization was guided by experimental data and specifically aimed at ensuring that the transformation maintains the fundamental properties of perceptual uniformity. In other words, it aims to align mathematical distances in color space with human-perceived color differences, thereby enabling more accurate and meaningful color interpolation.

In addition to the color spaces discussed earlier, numerous other color appearance models have been developed and studied, each designed to address specific shortcomings in color representation and human perception. These limitations have motivated researchers to explore the underlying causes of such issues and to propose hypotheses that explain the observed inconsistencies [Mor03]. One example of undesirable color behavior is illustrated in Figure 3.1, where a noticeable and unintended shift toward purple occurs during interpolation in the CIELAB space, specifically in the transition from blue to gray. Notably, this type of artifact does not appear when using alternative color models such as ProLAB. ProLAB achieves its intended goal of perceptual uniformity while avoiding major undesired behaviors during interpolation. These findings highlight the

3 Related Work

importance of selecting an appropriate color space for tasks involving color interpolation or manipulation, particularly in applications where visual accuracy and perceptual consistency are essential.

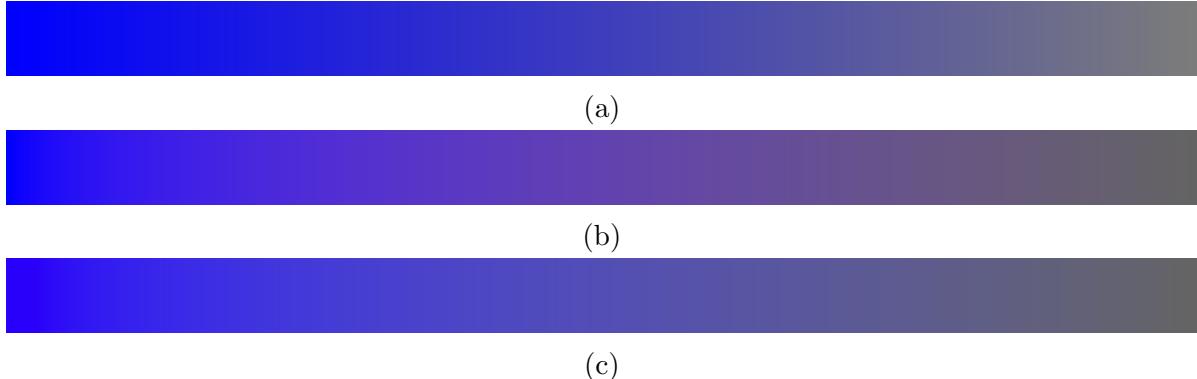


Figure 3.1: Illustration of unintended color appearance during interpolation between blue and gray. (a) Interpolation in the CIERGB color space. (b) Interpolation in the CIELAB color space. (c) Interpolation in the ProLAB color appearance model.

The work by Seymour [Sey22a] also highlights several critical issues in the CIELAB color space, including color inconsistency and more fundamental limitations. Rather than attempting to address these shortcomings individually and incrementally, Seymour argues that it is more effective to abandon CIELAB altogether. This recommendation is based on the fact that many of CIELAB’s core is constrained by the need to emulate device characteristics that have become obsolete. These legacy design choices remain in use primarily for convenience rather than technical necessity, as further discussed in his webinar talk *CIELAB is Icky! Do we have any other choice?* [Sey24].

One of the main issues identified in CIELAB, as highlighted in Seymour’s paper [Sey22b], is its reliance on tristimulus functions rather than directly modeling cone response functions in the construction of the color space. This foundational design choice was originally made to align with the capabilities of now-obsolete color measurement devices. However, it introduces several inconsistencies in how color is represented and perceived. The use of tristimulus values, which were intended to approximate average human vision through device-oriented calibration, does not accurately capture the nonlinear behavior of the human visual system, especially under varying luminance and chromatic adaptation conditions. As a result, CIELAB may produce perceptual distortions, including

3 Related Work

unintended hue shifts and non-uniform color transitions. These issues become particularly noticeable in tasks such as color interpolation and transformation.

An alternative suggested by the literature is to use modern color appearance models such as in the standard CIECAM, extending the capabilities of the CIELAB color space and handling the core issues of the color space [Li+02]. Among these, CIECAM02 significantly improves upon the major limitations found in both CIELAB and its predecessor, CIECAM97. According to Li et al., CIECAM02 provides a more accurate and consistent representation of color appearance under varying viewing conditions, making it a strong candidate to be considered as a new standard replacement for CIELAB in many applications.

The scope of this thesis is limited to the study of three color spaces: ProLAB, CIELAB, and the classical CIERGB color space. This selection was made to avoid a broad comparison aimed at determining the “best” color space, and instead to focus specifically on whether color appearance models have a meaningful role in direct volume rendering. ProLAB was included due to its relative simplicity of implementation and its improved perceptual uniformity compared to CIELAB [Kon+21].

3.2 Perceptual Color Metrics

Building upon the insightful study by Bujack et al. *The Good, the Bad, and the Ugly: A Theoretical Framework for the Assessment of Continuous Colormaps* [Buj+18], the authors made a significant contribution by consolidating and clarifying key terminology in the field. They identified and reconciled instances where different terms had been used to describe the same concept and where identical terms had taken on different meanings. In their words, they aim to “unify, clarify and standardize the taxonomy by distinguishing different uses of the same terms and summarizing different terms for the same concept.”

Their central goal was to bring objectivity to the evaluation of perceptual uniformity in colormaps, an area that has traditionally been dominated by subjective judgment, by introducing mathematical models and metrics. While not all concepts were fully formalized due to the limitations of their scope, their framework provides a robust foundation for future work. In our context, this framework offers valuable guidance as we attempt

3 Related Work

to objectify the perceptual uniformity effectiveness of color palettes used in transfer functions for volume rendering.

In this thesis, we will adopt the terminology and conceptual framework introduced in the aforementioned paper to maintain consistency with established standards and to prevent any potential misinterpretation arising from vague or undefined concepts. This alignment ensures clarity and precision throughout our discussion, particularly when addressing perceptual aspects of color in the context of transfer functions.

For answering **(Q1.1) How can the perceptual uniformity of transfer functions be objectively evaluated?**, we must delve deeper into the definition of transfer functions. As outlined in the previous section on volume rendering, there are several types of transfer functions, all sharing the same core objective: transforming volumetric data into a meaningful 3D visualization. For the sake of simplicity, and in alignment with the foundational studies that initially explored this field, we will focus our scope exclusively on 1D transfer functions. Although other properties, such as gradient magnitude or second-order derivatives, have proven to be more effective, intuitive, and capable of producing superior results in both manual and automatic configurations [Lju+16], they are considered beyond the scope of this thesis and are left for future exploration.

In the context of 1D transfer functions, the definition typically includes a set of color control points, which together form a color palette, as commonly implemented in mainstream software such as ParaView or Tomviz. However, a complete transfer function also incorporates an opacity channel, which plays a critical role in volume rendering techniques such as ray marching. This is especially important for optimizations that allow the rendering process to terminate early when opacity reaches its maximum value. Opacity can be conceptually understood as the blending between foreground and background colors based on its intensity. This allows us to approximate the result as a simple interpolation of a new blended color, assuming a fixed background. It is important to note, however, that this approximation is not always accurate in 3D rendering contexts. In real scenes, the background is rarely a uniform color, and internal structures can influence the final visual result. For example, in a volumetric rendering of a human head, the semi-transparent skin layer may be visually affected by the underlying muscles, which act as a variable background.

Another important consideration is that, according to the literature, perceptually uniform color spaces are non-Euclidean [HMO06] [RF12] and also non-Riemannian [Buj+22].

3 Related Work

Consequently, the distances between two colors in such spaces should not be computed using traditional Euclidean metrics. Instead, geometric properties must be adapted to account for the intrinsic curvature of the color space. For instance, the shortest path between two colors is given by the concept of a geodesic, which generalizes the notion of a straight line to non-Euclidean geometries. The implementation of transfer functions within perceptually uniform color spaces requires special handling of color distances. A preliminary version of such an implementation was proposed by Zeyen et al. [Zey+18] and later optimized in a subsequent work of Bujack et al. [Buj+24]. Their algorithm provides a method for computing interpolated points along geodesic paths between colors. This approach has proven particularly valuable for comparing nearly perceptually uniform color palettes with those derived from traditional color spaces. For simplicity, in this thesis, we refer to the resulting interpolated color set as belonging to the "CIEDE2000 color space." However, it is important to note that no such formal color space exists; this terminology merely denotes the collection of interpolated colors generated using the geodesic algorithm based on the CIEDE2000 color difference formula.

Before introducing the key concepts central to this thesis, it is important to establish some notational conventions that will be used throughout. A fundamental element in most of the perceptual metrics discussed later is the concept of distance between two colors. This is formally defined in Equation 3.5.

$$D_{i,j} := \Delta E_{\text{year}}(C_i, C_j) \quad (3.5)$$

Here, $D_{i,j}$ represents the perceptual distance between the colors C_i and C_j , and ΔE_{year} refers to one of the color difference formulae (e.g., ΔE_{1976} , ΔE_{1994} , ΔE_{2000}) described previously in the section 2.3. As highlighted in the referenced study, it is also useful to construct a visualization matrix of color distances for a given palette. From this matrix, various statistical properties, such as the mean, minimum, maximum, and standard deviation, can be computed. These metrics serve to quantitatively characterize the perceptual uniformity of a color palette and, by extension, the transfer function it defines.

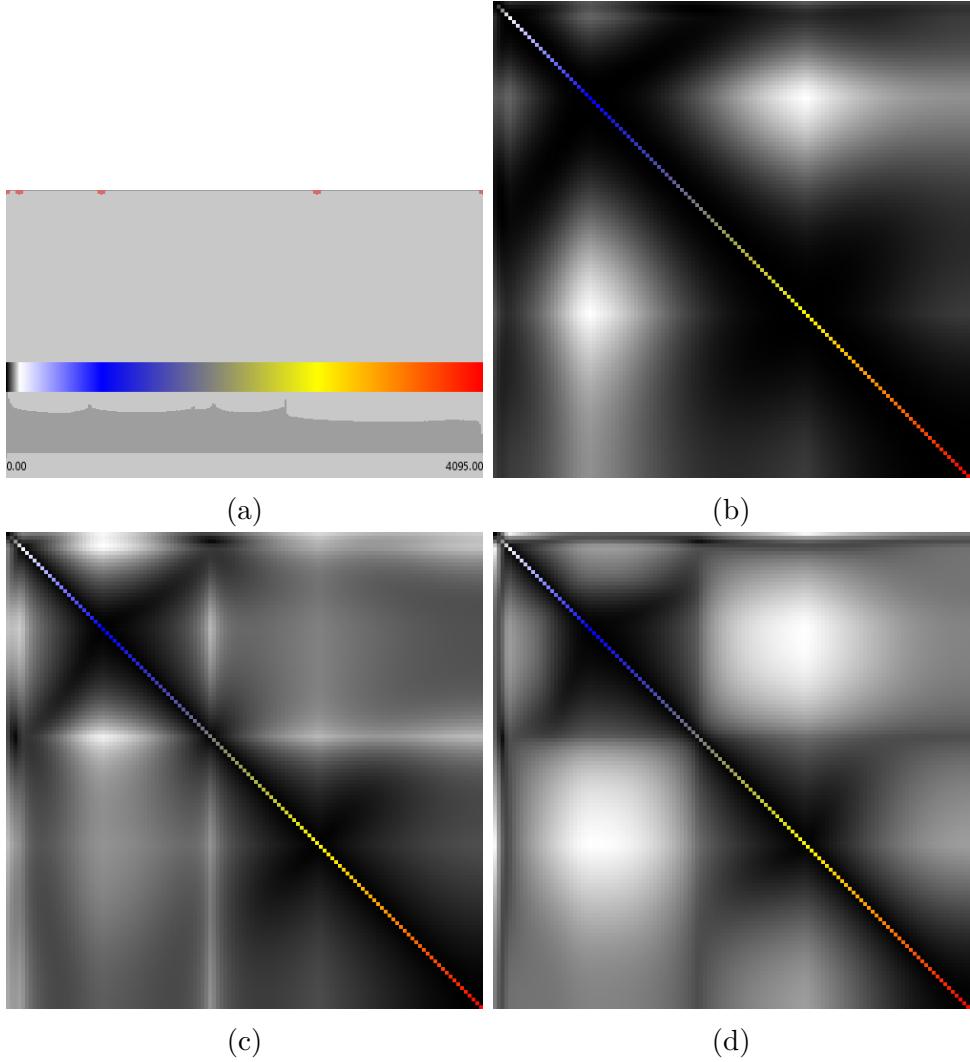


Figure 3.2: Color distance matrix for the defined transfer function in (a) using the three defined color distance metrics normalized between 0 and 1 and plotted in gray scale: (b) ΔE_{1976} , (c) ΔE_{1994} , and (d) ΔE_{2000} .

Figure 3.2 illustrates the distance between the corresponding row and column on the diagonal. Additionally, it is important to highlight that the color difference between black and white is not the highest in non-perceptual color distances (top right or bottom left in the color matrices), which supports the argument for using perceptual color distances. These are better suited for detecting real differences on the screen.

In addition, by integrating the concept of just noticeable difference (JND) into the visualization matrix of color distances, we can generate a new matrix-based image associated with the evaluated color palette. Given that a distance value greater than 1 is considered

3 Related Work

the threshold at which a typical observer can perceive a color difference, this new matrix can explicitly highlight the perceptual variations within the palette. Specifically, it reveals the number of distinguishable differences, helping assess the palette's effectiveness in visual discrimination. An example of this visualization is shown in Figure 3.3.

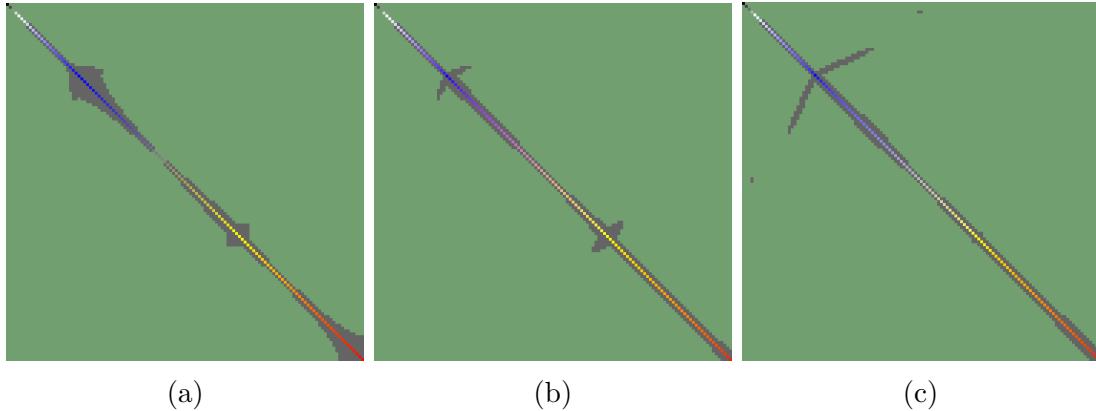


Figure 3.3: Visualization of just noticeable differences (JND) for a transfer function color palette. Green areas represent perceptible differences according to the JND threshold, while gray areas indicate imperceptible differences. (a) JND visualization in the RGB color space. (b) Corresponding JND visualization in the CIELAB color space using the same set of control points. (c) JND visualization in the ProLAB color appearance model.

The green mask highlights regions within the color palette where differences are perceptible, while the gray areas correspond to indistinguishable zones. These regions are determined using the empirically defined just noticeable difference (JND) threshold introduced in the previous chapter, and they are applied here using recent perceptual color difference formula. This visualization clearly illustrates the non-uniformity inherent in the CIERGB color space. In contrast, the CIELAB representation, when combined with a transfer function constructed through CIEDE minimization, demonstrates a more uniformly distributed perceptual space.

With the concept of distance established, the authors also introduce the notion of velocity, which we later adapt to our specific context of transfer functions. To calculate this value, they define the spatial distance between the positions of two colors as $|t_j - t_i|$. It is important to emphasize that two distinct types of distances are involved. The first is the color distance, previously defined using perceptual metrics, and the second is the spatial distance, which refers to the separation between control points in the color palette,

3 Related Work

typically 1 pixel long. The velocity between two colors can then be mathematically described as shown in Equation 3.6.

$$V_{i,j} := \frac{D_{i,j}}{|t_j - t_i|} \quad (3.6)$$

In their original work [Buj+18], the authors define both local and global measures for distance and speed. Local measures focus on comparisons between neighboring colors, while global measures consider all possible combinations within the entire color palette. Since our primary interest lies in examining the role of perceptual uniformity in volume rendering, we will restrict our analysis to local measures only. This localized approach allows us to better capture subtle variations and transitions relevant to the visual continuity of transfer functions. Their mathematical expressions are displayed in Equation 3.7.

$$d_j := D_{j,j-1} \quad v_j := V_{j,j-1} \quad (3.7)$$

The first measure of interest is the local discriminative power, which intuitively reflects how distinguishable adjacent colors within a color palette are. In other words, it describes the extent to which color differences can be perceived by the human eye. This concept is closely related to the notion of just noticeable differences (JND). It is also worth noting that, when evaluated using unit intervals, this property is equivalent to the average local speed \bar{v} and it is mathematically expressed in Equation 3.8.

$$\bar{v} := \sum_{j=1}^n d_j \quad (3.8)$$

Now that the notion of local speed has been defined, we can describe a color space as uniform if the variation of local speed across all color palettes remains consistent. In perceptual terms, this means that two points placed at equal distances from a reference point in the color space should exhibit the same perceptual difference when compared to that reference color. As suggested in the original study [Buj+18], a suitable measure for evaluating local uniformity is the standard deviation of the local speed. This notion is mathematically described in Equation 3.9.

$$\sigma_v := \sqrt{\sum (v_j - \bar{v})^2} \quad (3.9)$$

4 Methodology

It always seems impossible until it's done.

Nelson Mandela

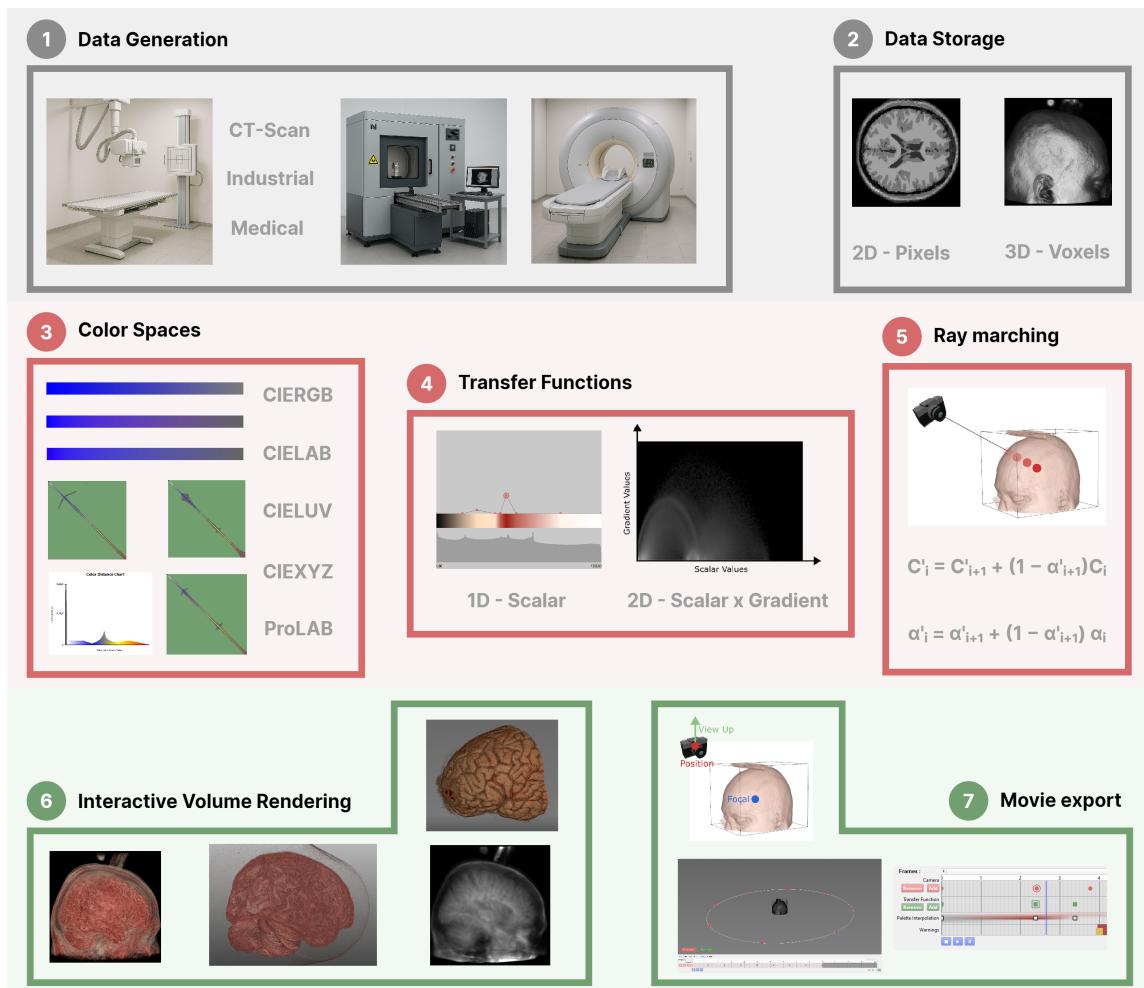


Figure 4.1: Diagram illustrating the volume visualization pipeline for the video export framework presented in this master thesis.

In this chapter, we begin by discussing the key elements required for camera trajectory interpolation in order to generate smooth and visually appealing motion. Figure 4.1 provides an overview of the data transformation pipeline that prepares the system for producing such motion. The challenges of interpolating both position and orientation are then investigated, with particular attention given to mismatches between these components. A modified version of Catmull-Rom interpolation is introduced to handle quaternion interpolation more effectively. We also address the problem of creating looped trajectories and explore the ability to pause spline motion at desired moments. Beyond trajectories, we examine how transfer functions can be interpolated, and a simple yet effective algorithm is presented to manage this task. Additionally, we incorporate the perceptually uniform color space ProLAB into our system to enhance color coherence. Finally, a new metaphor for representing transfer functions within a timeline-inspired interface is proposed, offering a more intuitive way to manage the temporal and visual aspects of volumetric rendering.

4.1 Camera Trajectory Interpolation

Generating insights into volumetric images has become crucial in the field of volume visualization for effective information transfer. CT scans are commonly used to create interpretable 3D images that aid in decision-making. In this thesis, we extend this concept by developing a framework that ensures smooth camera trajectory generation. When used for video export, this framework helps users maintain spatial awareness while analyzing volumetric renderings. For interpolating positional control points, we employ Catmull-Rom interpolation, previously discussed in Section 2.2, which offers two key advantages:

- An intuitive method for defining control points.
- Smooth curves that naturally follow the given trajectory.

An important challenge in trajectory interpolation is managing trajectory borders. To accurately approximate a circular path using Catmull-Rom interpolation, careful selection of control points is necessary. Specifically, the start and end points should correspond to either the first and second or the last and second-to-last points in the control point list. The directional control points should be selected accordingly to ensure smooth

4 Methodology

interpolation. A practical approach to achieve this is by treating the control point list as circular, allowing for seamless iteration over segments to construct a smooth and continuous trajectory. The figure 4.2 illustrates this construction.

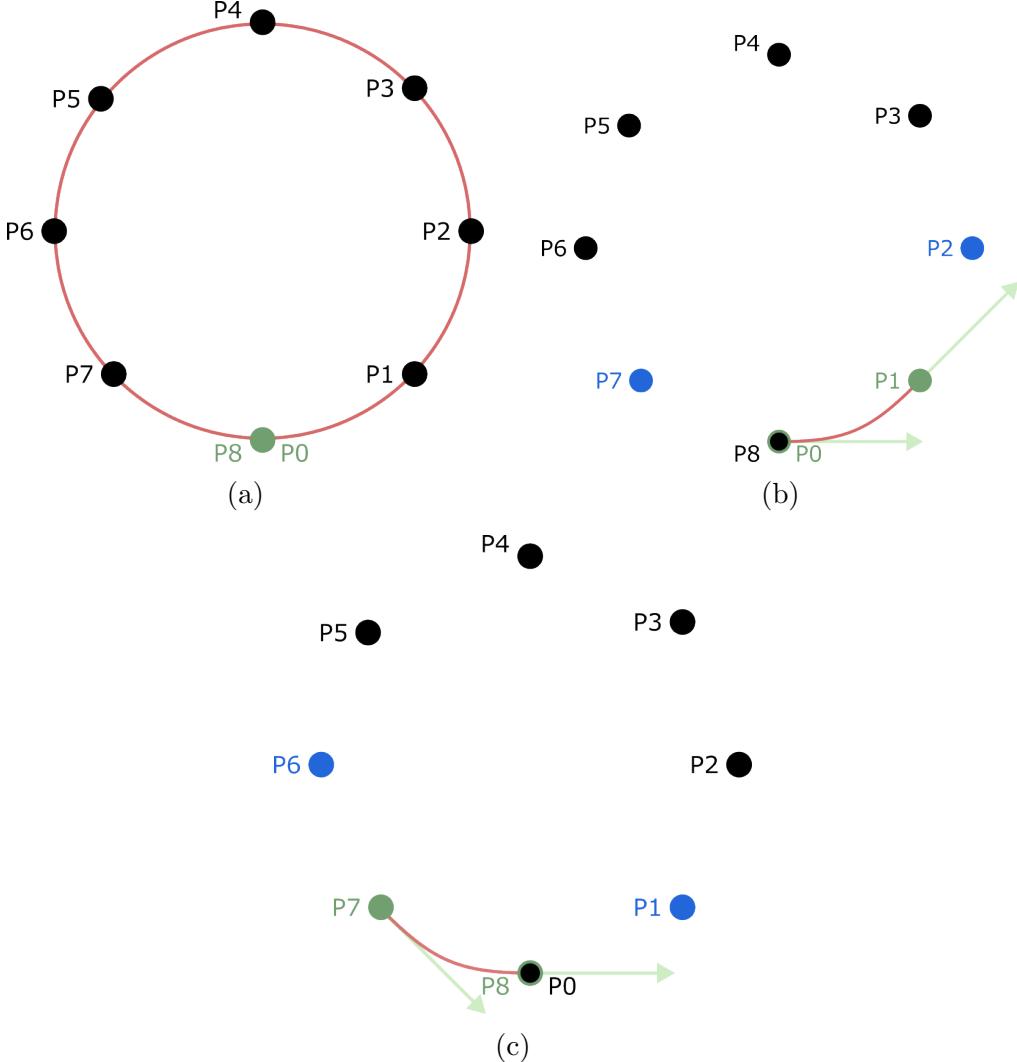


Figure 4.2: (a) Illustration of the construction of Hermite segments when two points (P_0 , P_8) coincide at the same position. (b) In the starting segment, P_2 and P_7 are the directional control points, while P_0 and P_1 are the start and end points, respectively. (c) In the end segment, P_6 and P_1 are the directional control points, while P_7 and P_8 are the start and end points, respectively.

The other scenario is illustrated in Figure 4.3, where the curve begins at one point and ends at another. In such cases, a good strategy is to ensure that the curve starts and finishes in a straight line. This approach eliminates unwanted curvature at the

4 Methodology

endpoints, as shown in Figure 4.3d, resulting in a smoother and more visually appealing interpolation. Algorithmically, this can be achieved by setting the directional control point equal to the starting or ending point. The red arrows in the figure illustrate the construction of the velocity vectors, as defined in Equation 2.8. The same logic is applied symmetrically at the end of the trajectory.

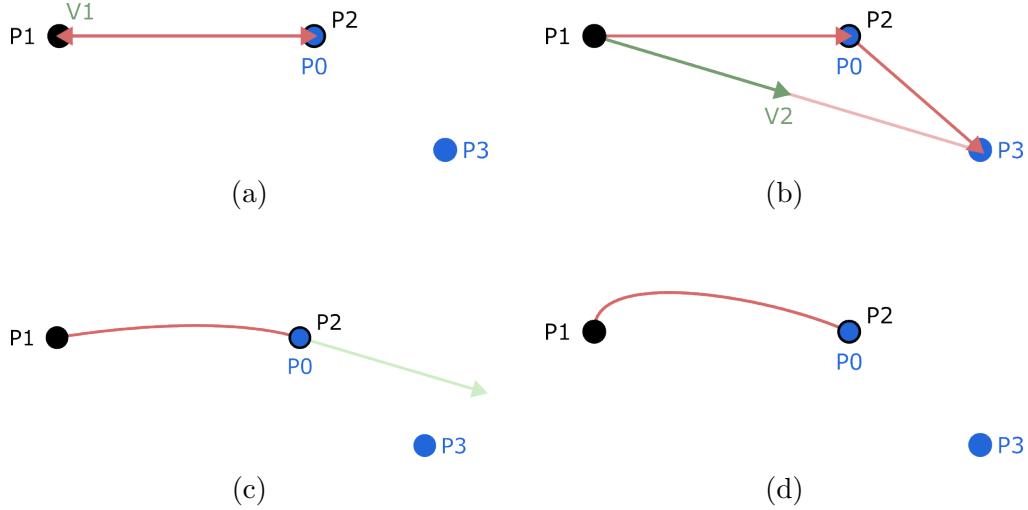


Figure 4.3: Illustration of the construction of a starting Hermite segment with a straight line, where P_1 and P_2 are the starting and ending points, and P_0 and P_3 are the directional control points. (a) The speed V_1 is chosen to be null by setting $P_2 = P_0$. (b) The speed V_2 is calculated following the standard rules. (c) The curve is constructed using Hermite interpolation. (d) An example of undesirable behavior when the Hermite interpolation begins without aligning the initial segment along a straight path.

For the orientation of the camera, we can follow two different approaches. The simpler approach applies Catmull-Rom interpolation to both the focal point and the view-up vector of the camera. In practice, these parameters are adjusted as the user modifies the camera orientation. The more advanced approach utilizes quaternions and an adapted version of Catmull-Rom interpolation specifically designed to work with quaternions. In the first approach, illustrated in Figure 4.4a, the focal point is typically defined either freely or at the center of the object. Setting the focal point at the center of the object, rather than the user's position at the time of defining the control points, simplifies the camera trajectory setup. This method ensures that the 3D volume remains perfectly centered on the screen, allowing for smooth and seamless cutscenes.

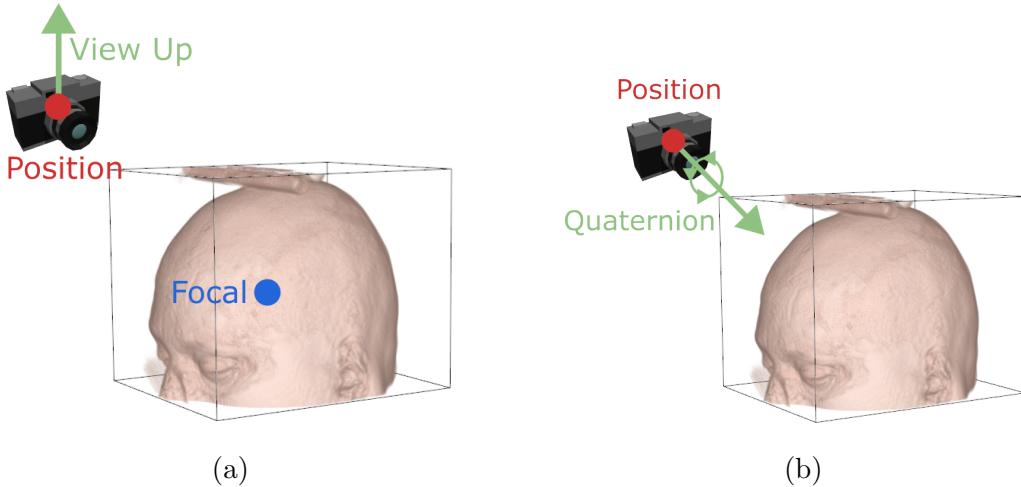


Figure 4.4: Illustration of camera orientation and position representation. (a) The camera’s state is defined using its position, view-up vector, and focal point. (b) The camera’s state is defined using a quaternion and its position.

When the focal point is no longer fixed at the center of the object and is instead interpolated, the process becomes more technical and challenging for generating smooth animations. In practice, users may unintentionally introduce inconsistencies when setting up control points, leading to undesirable wobbling effects due to the decentralization of the object in the trajectory curves. Another essential variable for interpolating camera properties is the view-up vector. Along with the position and focal point, it forms the orthogonal basis required to precisely define the camera’s orientation in 3D space. One approach to interpolating this vector is to treat it as a point and apply Catmull-Rom interpolation accordingly.

In the second approach, illustrated in Figure 4.4b, we utilize quaternions to interpolate the camera’s orientation across all control points. This method eliminates the need for a focal point or view-up vector, as quaternions fully define the camera’s orientation. However, quaternion interpolation is more complex and requires an adapted version of Catmull-Rom interpolation. As detailed in the insightful article by [Hol23], several key aspects must be considered. First, instead of using linear velocity, we work with angular velocity, which is more appropriate for rotational motion. Second, before mixing angular velocities, we convert the quaternions into axis-angle representation. Finally, after processing the interpolation, we convert the results back into quaternion form. More details on the implementation can be found in the original reference.

4 Methodology

After implementing both quaternion-based and view-up/focal-point-based representations for camera orientation, along with position interpolation, we were able to assess the strengths and limitations of each method in the context of camera path planning. For trajectory interpolation, the Catmull-Rom spline was chosen due to its ability to produce smooth and visually appealing paths. It was slightly adapted to support smooth pauses within animations. As previously noted, Catmull-Rom splines are a type of cubic spline that ensure continuity in both position and first derivative, which is crucial for natural camera motion. However, this continuity also makes it challenging to introduce intentional discontinuities or pauses by simply adding two control points at the same location. Figure 4.5 illustrates the issue: when control points are placed too close together, the resulting curve remains smooth but leads to unnatural or unintended motion in the animation path.

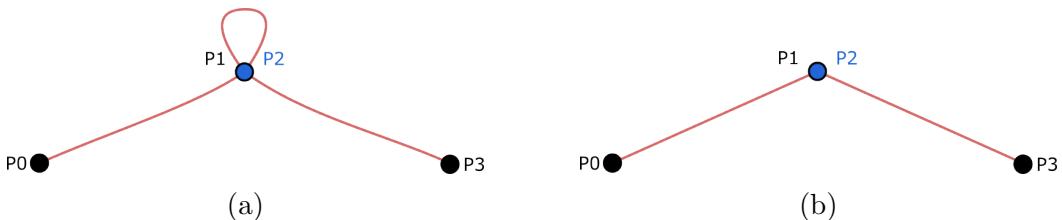


Figure 4.5: (a) Illustration of an unintended loop caused by placing two nearby control points in a Catmull-Rom spline, resulting in a smooth but undesired motion path. (b) Desired behavior when setting a pause by placing two control points at the exact same position, producing a stationary segment in the animation.

Now, let us revisit our sub-research question **(Q2.2) How can pause frames be effectively encoded as control points within a trajectory definition framework?**. To address this challenge, we delved deeper into the foundation of Catmull-Rom interpolation and further explored the literature surrounding spline-based curve modeling. This investigation led us to the concept of monotone cubic interpolation, a technique designed to preserve monotonicity and prevent unwanted oscillations or overshooting in interpolated curves. The approach is well-documented in the literature [Bir20] and effectively implemented in practical applications, such as those described in [Hol23]. What makes this solution particularly appealing is its simplicity and elegance. It involves introducing specific constraints on the derivative values used within the Hermite spline formulation. By using this monotonicity implementation appropriately, the interpolation becomes more controlled and predictable, allowing for smooth transitions while maintaining the

4 Methodology

option to include effective pause frames where desired. This technique is mathematically described in Equation 4.1.

$$\text{Catmull-rom-monotone}(x, P_0, P_1, P_2, P_3) = \text{Hermite}(P_1, P_2, LV_1, LV_2) \quad (4.1)$$

Where LV_1 and LV_2 represent the constrained speed values passed into the Hermite spline for interpolation. Specifically, all components of the speeds are calculated for each dimension $i \in x, y, z$.

$$LV_{1i} = \begin{cases} \text{clamp}(V_1, T_{\max}(P_0, P_1, P_2), T_{\min}(P_0, P_1, P_2)) & \text{if } \text{sign}(P_1 - P_0) = \text{sign}(P_2 - P_1) \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

where:

$$\begin{aligned} T_{\max}(P_0, P_1, P_2) &= \max(-3|P_1 - P_0| - 3|P_2 - P_1|) \\ T_{\min}(P_0, P_1, P_2) &= \min(3|P_1 - P_0|, 3|P_2 - P_1|) \end{aligned}$$

And analogously for LV_2 :

$$LV_{2i} = \begin{cases} \text{clamp}(V_2, T_{\max}(P_1, P_2, P_3), T_{\min}(P_1, P_2, P_3)) & \text{if } \text{sign}(P_2 - P_1) = \text{sign}(P_3 - P_2) \\ 0 & \text{otherwise} \end{cases} \quad (4.3)$$

where:

$$\begin{aligned} T_{\max}(P_1, P_2, P_3) &= \max(-3|P_2 - P_1| - 3|P_3 - P_2|) \\ T_{\min}(P_1, P_2, P_3) &= \min(3|P_2 - P_1|, 3|P_3 - P_2|) \end{aligned}$$

By applying Equation 4.1 , we can generalize the interpolation method to operate not only on camera positions but also on focal points. This extension allows to smoothly interpolate the point of interest that the camera is looking at, ensuring coherent transitions in viewpoint direction alongside the camera's spatial movement. Such an approach is particularly useful for maintaining visual continuity and enhancing user orientation in dynamic volume-rendered scenes.

4 Methodology

However, this method encounters a limitation when applied to the view-up vector. Unlike scalar points, vectors represent direction and orientation, which makes their interpolation less straightforward and intuitive. The challenge lies in the need to preserve directionality and avoid distortions during the transition. Nonetheless, if the directional change between the initial and final vectors is not too large, specifically if the angle between them does not exceed approximately 120 degrees, then the same interpolation method used for positional points can be applied effectively to vectors. This 120-degree threshold was determined empirically and serves as a practical guideline for ensuring smooth and consistent camera orientation.

It is important to note that this method may introduce abrupt changes in speed, particularly when transitioning between the points of different spline segments. These fluctuations can lead to undesirable oscillations along the camera path, potentially disrupting the perceived smoothness of motion and negatively impacting the user experience. As discussed in Chapter 2.2, such variations can be mitigated by weighting the spline interpolation based on the actual distances between control points. This adjustment ensures a more consistent traversal speed by aligning the timing of interpolation with the true spatial configuration of the path.

Another approach to interpolating camera orientation, as discussed previously, is to use quaternions instead of relying on the view-up and focal-point definitions of the camera orientation. In practice, these camera attributes are converted into quaternions, and then a modified version of Catmull-Rom interpolation is applied to the quaternions. The modification involves converting linear velocities into angular velocities within the algorithm. By using quaternions for orientation instead of positions, we can apply the correct rotational representation to mix angular velocities, taking full advantage of the properties of quaternions. A detailed implementation of this approach can be found in the original paper [Hol23].

Now, we can combine all the interpolation methods to create the final camera path planning framework. However, a new challenge emerges when combining position and orientation interpolation: desynchronization. This issue occurs when the interpolations of position and orientation are not aligned, resulting in a noticeable effect where the object becomes misaligned and starts to wobble. Figure 4.6 illustrates this misfocus caused by desynchronization. One solution, mentioned earlier in the chapters, is to lock the focal point to the center of the image or introduce additional control points to

4 Methodology

mitigate this effect. However, this approach reduces the degrees of freedom available for the user when setting up the camera path.

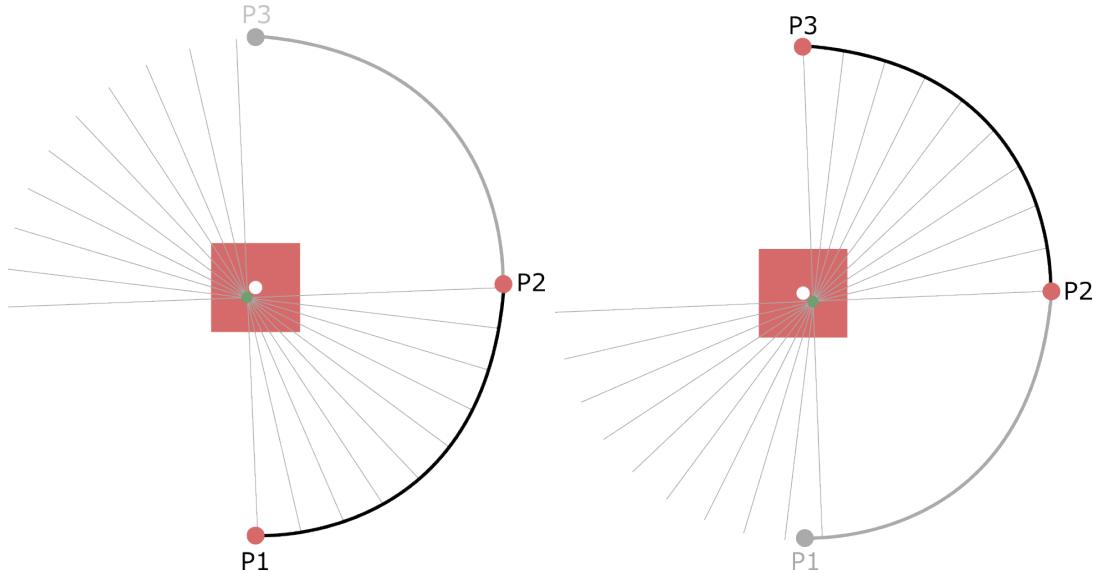


Figure 4.6: Illustration showing camera misalignment caused by separate interpolation of position and orientation. The point of convergence for all orientations is not centered, and it varies between spline segments.

To conclude this section, the methodology and design decisions behind the camera trajectory interpolation system can be effectively summarized using the "What, Why, and How" framework as presented in the book *Visualization of Time-Oriented Data* by Aigner et al. [Aig+23]. This structured approach helps clarify the objectives, motivations, and technical solutions that guided the development of the camera path and orientation interpolation system.

- **What?**

The main objective is to create a continuous and well-defined camera trajectory, accompanied by a corresponding sequence of orientation points. Together, these components enable the generation of smooth and visually appealing animations. The trajectory should be easy for users to define, requiring minimal effort while still allowing precise control over the camera's movement. At the same time, the motion should avoid unnatural or erratic behavior, such as awkward shifts, sudden changes, or misaligned focus on the object. The system should also support the inclusion of pause effects, which allow for more deliberate and controlled animations when necessary.

- **Why?**

This implementation offers valuable benefits across a range of application domains. By allowing users to efficiently generate coherent and attractive camera animations, the system increases the communicative power of volumetric visualizations. These animations can serve as effective tools for presenting complex information in educational, promotional, or scientific contexts. In more critical areas, such as surgical planning, medical training, or defect inspection in materials science, smooth and accurate camera trajectories are essential for proper interpretation and collaboration.

- **How?**

To solve the challenges of camera interpolation, the system introduces a multi-method framework. For positional interpolation, the Catmull-Rom spline algorithm is used to generate smooth paths through a set of user-defined keypoints. Orientation is handled using one of the three complementary methods. The first method interpolates both the view-up vector and the focal point using the same Catmull-Rom algorithm, resulting in a balanced orientation trajectory. The second method simplifies orientation control by fixing the focal point at the center of the object and interpolating only the view-up vector, which is particularly useful for orbital camera movements. The third method interpolates quaternion representations of the camera's orientation, using a modified Catmull-Rom algorithm adapted for this representation. This approach ensures smooth rotational transitions, even in complex motion scenarios. Additionally, a monotonic speed control mechanism is introduced to support camera pauses. This allows users to define segments of the trajectory with zero velocity, enabling precise control over timing without introducing visual artifacts or unstable behavior.

4.2 General Transfer Function Interpolation Design

1D transfer functions are defined by a set of control points, each specifying a particular color and opacity for a given data value. To represent the full range of values within the volumetric dataset, these points are typically interpolated using a simple RGB interpolation method to generate a continuous color palette. Our research has identified a gap in the existing literature, especially regarding the exploration of alternative interpolation

4 Methodology

techniques for computing intermediate points. Perceptually uniform color palettes, however, naturally enhance color discrimination and coherence, preventing the introduction of unwanted tonal distortions in the final rendered image. These tonalities, as discussed in [Buj23], can introduce undesirable shifts toward colors that are semantically unrelated to the data being represented, potentially misleading the viewer and impairing the interpretability of the visualization. By eliminating these inconsistencies and offering other benefits, perceptually uniform color palettes improve the representation of finer details, ultimately resulting in higher-quality output. This observation motivates the central research question of this thesis, formulated in **(Q1) How can perceptually uniform color palettes be used in Volume rendering?**.

The interpolation of transfer functions becomes essential when maintaining spatial awareness while ensuring smooth and visually appealing transformations of the rendered volumetric data. These interpolations are particularly useful when information needs to be effectively shared, such as in medical conferences, prototype presentations, or fiber analysis. In summary, for any application where classical transfer functions are utilized, transfer function interpolation can provide valuable enhancements. To properly understand transfer function blending, it is crucial to differentiate between two types of interpolation involved in the process.

- **Interpolation within a Single Transfer Function:** This occurs inherently within a single transfer function. Control points, which are mapped to scalar values of the volumetric data, define colors and opacities. The interpolation process determines the intermediate colors and opacities for values between these control points.
- **Interpolation Between Multiple Transfer Functions:** This involves blending different transfer functions rather than just interpolating within one. Here, control points define a mapping not from scalar values to colors and opacities, but from an index ratio to a specific transfer function. This index increases with the number of control points, and its decimal part determines the blend between the nearest defined transfer functions.

To address the second type of interpolation challenge identified in our study, we propose a simple yet effective Algorithm 1 that provides a practical solution for computing intermediate values in transfer functions. While the method is intentionally designed to be naive in its implementation for the sake of clarity and accessibility, it serves as a

4 Methodology

foundational step toward answering the research question outlined in **(Q3) How can transfer function interpolations be executed to produce smooth, semantically coherent, and visually appealing animations?**. By enabling efficient and perceptually aware interpolation within transfer functions, our approach contributes to improving the visual quality and interpretability of volume-rendered animations.

To better understand the interpolation algorithm, let us first consider a simple case: Interpolating between just two transfer function control points. Once this fundamental interpolation is established, we can extend the approach to multiple control points by processing them in consecutive pairs. First let us establish the necessary terminology. A transfer function maps scalar values from a volumetric dataset to colors and opacities to be used for the compositing step in direct volume visualization. The x-axis of a transfer function corresponds to the scalar values, which typically span the entire range of the dataset. When blending between two different transfer functions, an interpolation ratio, denoted as t , is introduced. This ratio varies from 0 to 1, where $t = 0$ corresponds to the first transfer function, $t = 1$ corresponds to the second, and intermediate values blend the two.

The interpolation algorithm follows three essential steps. First, a new list of scalars is constructed by combining all the unique x-axis scalar values from both input transfer functions. At this stage, only the scalar values are considered, and their corresponding color and opacity mappings are temporarily ignored. In the second step, for each scalar value in the new list, the corresponding colors and opacities from both input transfer functions are retrieved and paired for future combination. Finally, in the third step, an interpolation is performed between these values using the ratio t . The interpolated color and opacity are computed as an interpolation of the initial and final paired values, ensuring a smooth transition between the two transfer functions. Refer to the accompanying pseudo-code illustration for a clearer understanding of the algorithm.

It is important to highlight that the interpolation of colors can be performed using different methods. In this thesis, we adopted the straightforward CIERGB interpolation, applying it independently to each color component. A more refined, yet computationally more demanding, alternative is the use of the ProLAB color space. As previously discussed, ProLAB tries to preserve perceptual uniformity, which could enhance the visibility of details in the resulting animation.

By ensuring that color transitions follow a perceptually uniform path, we can achieve

Algorithm 1 Naive Interpolation Between Two Transfer Functions

Require: $t \in [0, 1]$, TransferFunction0, TransferFunction1
Ensure: TransferFunctionInterpolated

```

1: controlPointsX  $\leftarrow []$ 
2: for all control points  $C$  in TransferFunction0 do
3:   if  $C \notin controlPointsX$  then
4:     controlPointsX  $\leftarrow controlPointsX.push(C.x)$ 
5:   end if
6: end for
7: for all control points  $C$  in TransferFunction1 do
8:   if  $C \notin controlPointsX$  then
9:     controlPointsX  $\leftarrow controlPointsX.push(C.x)$ 
10:  end if
11: end for
12: for all scalar  $X$  in controlPointsX do
13:    $color_0 \leftarrow TransferFunction0.GetColor(X)$ 
14:    $color_1 \leftarrow TransferFunction1.GetColor(X)$ 
15:    $colorInterpolated \leftarrow InterpolateColors(t, color_0, color_1)$ 
16:   TransferFunctionInterpolated.AddControlPoint( $X, colorInterpolated$ )
17: end for
```

smoother and more intuitive transformations in our framework. In practical terms, this means that modifying transfer function keyframes results in more controlled and predictable changes, making it significantly easier to create visually appealing animations. Given that transfer functions are already complex to manipulate, any improvement that simplifies the process is a valuable contribution.

4.3 Transfer Function Interpolation in a Timeline

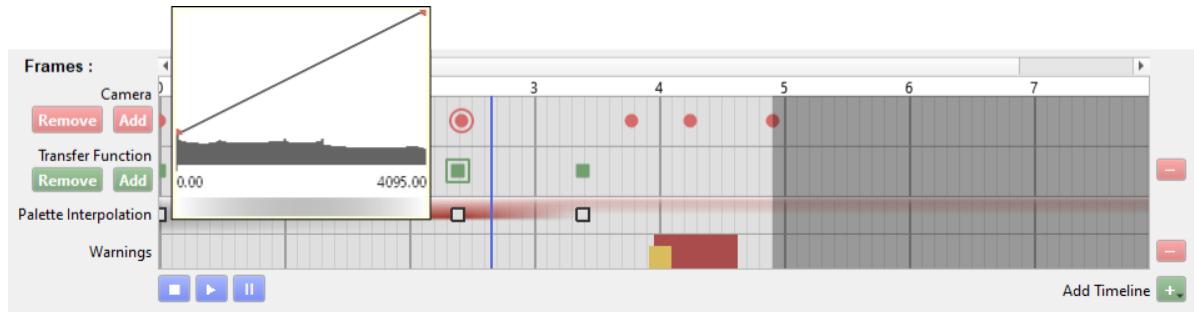


Figure 4.7: Widget displaying the timeline of our framework, encoding transfer function control points, color palette interpolation, and transfer function definition through hovering.

Now that we understand how to interpolate between transfer functions, it becomes essential to provide a conceptual metaphor within our framework that empowers users to intuitively manage and explore these interpolations. To support this need, we take again the inspiration from the "What, Why, and How" methodology presented in the book *Visualization of Time-Oriented Data* by Aigner et al. [Aig+23]. This methodology emphasizes the importance of building visualizations that are expressive, effective in supporting user goals, and efficient in interaction and performance. By aligning our framework with these principles, we aim to ensure that users are not only able to understand the purpose and structure of the interpolation process but can also navigate and control it with minimal cognitive effort. Our implementation adheres to the "overview first, zoom and filter, details-on-demand" philosophy, which has proven successful in a wide range of visualization systems. This approach ensures a smooth and insightful user experience, allowing users to first gain a broad understanding of the data, then interactively explore specific regions of interest, and finally retrieve detailed information as needed.

- **What?**

In our framework, we aim to visualize the evolution of transfer function definitions over a continuous time scale. More specifically, this visualization operates within a time-based interval domain, meaning that intermediate points are interpolated to ensure smooth transitions. The representation follows a linear arrangement, maintaining an ordered viewpoint that provides clarity and coherence in the interpolation process. It is important to note that interpolating transfer functions is

4 Methodology

not a straightforward task due to their structure, which consists of an ordered set of color and opacity control points, each mapped to a specific scalar value within the observed volume. Additionally, at the core of transfer functions lies an intrinsic interpolation process that fills in the gaps between these control points, creating a continuous mapping from scalar values to optical properties. This layered interpolation complexity makes the transition between transfer functions particularly challenging.

- **Why?**

Implementing this visualization technique would enhance the user's understanding of how the transfer function evolves over time. A practical benefit of this approach is that users could identify and correct unwanted colors or opacity levels that may arise from non-perceptual color palettes. By navigating to a specific frame in the timeline, they could make precise adjustments to the transfer function, ensuring a more visually coherent and effective representation.

- **How?**

We propose separating individual color palettes from each transfer function definition, rotating them 90 degrees counterclockwise for intuitive user interpretation, and interpolating across all intermediate control points. The resulting visualization can be observed at the color palette interpolation row of the Figure 4.7. Additionally, control points are represented as interactive elements that users can hover over to view detailed information about the transfer function at that specific position.

5 Results & Evaluations

Science is built up with facts, as a house is with stones. But a collection of facts is no more a science than a heap of stones is a house.

Henri Poincaré

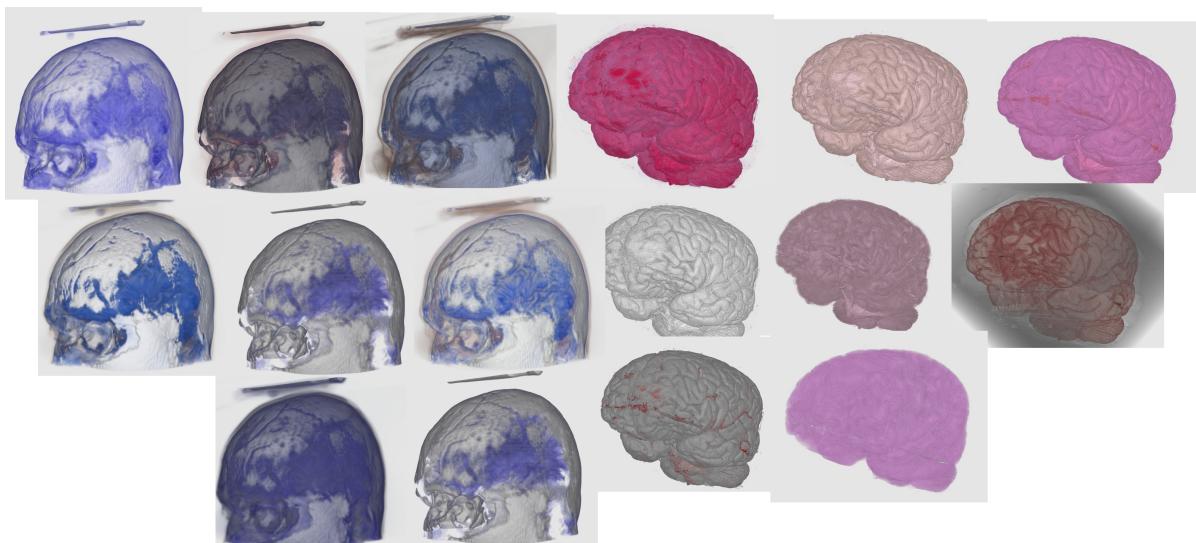


Figure 5.1: Volume-rendered images generated from user-defined transfer functions during the evaluation of perceptual uniformity in direct volumetric rendering.

In this chapter, we divide the evaluation into two main components: first, the assessment of our video export framework, and second, the examination of the role of perceptual uniformity in volume rendering. Each part is structured into two sections: one describing the design of the corresponding user study, and the other presenting the results obtained.

Our evaluation methodology combines both subjective feedback, gathered from participants who engaged with the system, and objective analysis using quantitative metrics defined in previous chapters.

5.1 Evaluation Design for Camera Trajectory Interpolation

With all the main concepts defined, including perceptual color blending and camera state management, we proceeded to evaluate our framework. A brief tutorial was provided to introduce participants to the available features, after which they were asked to create a video of their choice. This user study targeted a small group of specialized participants who carefully familiarized themselves with the application and completed the assigned tasks. The primary task assigned to the participants was to follow written instructions step by step to become familiar with the framework. The first dataset used was a human head extracted from the VTK samples database. Throughout the experiment, participants were continuously questioned about any difficulties they faced and their thoughts on the key functionalities of the framework. For the second task, users were asked to create a video of their choice and export it, allowing us to assess how effectively they were using the application autonomously while gathering their overall feedback. The datasets available for this second phase included samples from the VTK library and the [Wal+24]. Participants were free to choose whichever dataset they preferred.

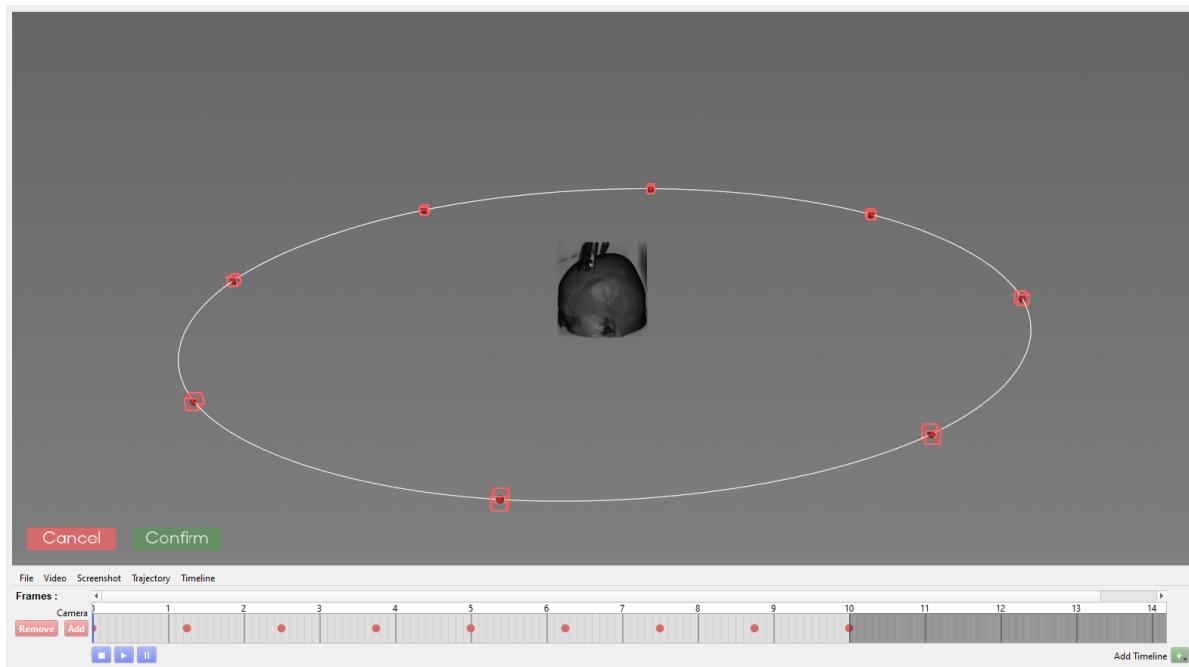
A practical note regarding dataset preparation: the organ database images were originally provided in the JPEG2000 (JP2) format. To maintain compatibility with the more commonly supported MHD/RAW formats in VTK, these images were batch-converted to TIFF format using GIMP. They were then loaded slice-by-slice into VTK, reconstructed into volumetric data, and finally saved again in MHD format to facilitate easier handling within the framework. The instructions for both tasks are outlined below:

Task 1:

- First load the head volume dataset. (File > Load Volume). You can find it in the provided directory resources within the compressed folder.

5 Results & Evaluations

- Now, create a circular trajectory using the default camera trajectory feature (Trajectory > Predefined > Circular). You can play the preview with the blue play button located at the bottom of the timeline.
- Then, modify the time spent in the middle section of the trajectory, i.e., the final video shall have a faster pace when moving from the back of the skull and a slower pace when traversing the front. You can adjust this by shortening and extending the keyframes' distances in the timeline.
- Next, the front of the skull should be more prominent on the screen. To achieve this, zoom in on the details of the front face. Enter the editing trajectory mode (Trajectory > Edit Trajectory). You will see different red or green spheres representing the control points. Move the control point closer to the central object. Using the X, Y, Z keys on the keyboard, lock the translation to only one axis. Confirm when you are done editing. The figure below shows the trajectory editing mode available in our video export framework.



- Next, create a pause effect when we are close to the skull, which is useful for visualizing detailed zones of interest. To do this, double-click the corresponding zoomed-in control point on the camera timeline. On the left side below the camera timeline, click "Add".

- Display the trajectory (Trajectory > Show Trajectory) to ensure the points coincide in the same location, without an undesirable trajectory. If the points are not aligned, enter the editing trajectory mode (Trajectory > Edit Trajectory) and adjust them so they overlap. Once the points coincide, adjust the duration of the pause by modifying the distances between keyframes in the timeline.
- Now that we have set a pause in front of the head, modify the transfer function for that section. In the bottom-right side of the application, add a new timeline: the transfer function timeline. Adding this timeline lets you control the evolution of the transfer function used to render the volume at a specific time.
- Double-click the first keyframe in the transfer function timeline and set the volume to display skin, skull, and muscle colors.
- After completing the aforementioned tasks, add two new transfer function keyframes: one at the beginning of the pause and another at the end. To add keyframes, position the cursor at the desired point on the timeline, double-click, and then click "Add" below the Transfer Function timeline.
- Next, modify the appearance of the head after the pause. Change the transfer function at the end of the pause by double-clicking it. Ensure that only the skull is visible in this section. Preview the visualization.
- Now, export the final video in HD. Go to Video > Export, select the path where you want to save the video, and specify the desired video duration. Finally, click "Export," and the video will be generated in the selected folder. Note that the keyframe distances will be proportionally preserved in the exported video.
- Once the video export is complete, you can save your project (File > Save Project) if you want to reopen it later and reset the camera and transfer function keyframes (Timeline > Reset Camera Frames / Timeline > Reset Transfer Functions).

Task 2:

- Load a volume of your choice (File > Load Volume). The datasets are available in the resources folder of the compressed directory.
- Add camera control points at positions where you would like the camera to pass. You can view the trajectory (Trajectory > Show Trajectory). To check if any

object is out of view (in red) or if the visualization speed is too fast (in yellow), activate the Warning timeline (Add timeline > Add Warning timeline).

- You can activate or deactivate interpolation methods for the camera orientation. You can choose to switch the view to object-centered or use quaternion interpolation instead of the view-up and focal-point interpolation.
- Finally, freely use the application to create a visualization of your choice. You can also use the transfer function timeline to experiment with the visualization.

During the execution of this user study, participants were also asked a series of questions to gather information about their prior experience and familiarity with functionalities similar to those offered by our video export framework. Table 5.1 summarizes all the questions presented to the participants during the study.

5.2 Evaluation Results regarding Camera Trajectory Interpolation

During our second user study, which was specifically aimed at evaluating the camera trajectory interpolation and, as a result, the video export framework, we worked with a smaller but more technically experienced group of participants. This focused evaluation provided several valuable insights into the current capabilities and limitations of the implementations. Some of the feedback addressed general software improvements, such as the need for a more intuitive user interface and better overall user experience. Other comments pointed to deeper issues related to the core functionality of the framework. One clear observation was that configuring high-quality 3D rendering videos using our system generally required some familiarity with video editing concepts. Participants with prior experience in similar tools were able to understand and navigate the available features more easily. For example, one participant initially found the interface and workflow confusing and expressed some frustration. However, after receiving only minimal guidance during the session, they quickly adapted to the system, produced an impressive final result, and reported enjoying the creative process of using the application.

Despite these positive outcomes, a critical issue was identified related to camera motion. Several participants noted a misalignment between the camera orientation and the

Task 1:	
	Have you previously used a video editing tool or any software with timelines and keyframes?
	What are your thoughts on the automatically generated camera path?
	How do you feel about the continuity of the video preview after adjusting the speeds?
	What are your impressions when setting control point positions in the preview to zoom into an area of interest?
	How did you find modifying the transfer function and viewing its effects in the video preview?
	How was your experience while exporting the video?
Task 2:	
	What method do you prefer for setting the orientation of the camera?
	How challenging was it to create an interesting camera trajectory for the video export?
Final Thoughts:	
	Do you see this application fitting into your workflow or professional activities? If so, how?
	Which domains or fields could benefit from the use of this framework?
	In what ways could this framework improve or support day-to-day tasks or workflows?

Table 5.1: List of questions used in the second user study to gather feedback from participants on the video export framework.

rendered object during animation playback. This misalignment resulted in a noticeable wobbling effect, which reduced the visual coherence of the output and disrupted the viewer’s sense of immersion. Even when the position and speed of the camera were interpolated smoothly, inconsistencies in orientation made the object appear unstable, which negatively impacted the quality of the final exported video. Addressing this issue would likely require a more advanced approach to camera interpolation, especially one that ensures a consistent relationship between the camera and the object across different segments of the spline. Although solving this problem was beyond the scope of the current work, it represents a meaningful direction for future improvement and further research.

One aspect of the application that received particularly positive feedback during the

5 Results & Evaluations

study was the simplicity and effectiveness of creating and editing predefined camera paths. Participants reported that with only a few clicks, they were able to produce smooth and coherent camera movements that offered comprehensive visual coverage of the volumetric data. Users appreciated the direct manipulation of control points using intuitive visual aids such as axis handles, as well as the real-time synchronization between the 3D view and the timeline interface. This integration provided immediate feedback and allowed users to fine-tune their animations efficiently. Despite these strengths, the lack of a user-friendly mechanism to adjust camera orientation independently of position was noted as a limitation. Several users expressed a desire for more direct control over the direction the camera was facing at each point along the path.

Beyond usability, participants also identified compelling potential use cases for the application across a range of professional domains. In the medical field, for example, users saw opportunities to use the tool for surgical planning and patient education. The ability to produce precise and immersive visual presentations was considered valuable for improving decision-making processes and enhancing communication between medical professionals. In the context of advertising and visual storytelling, participants noted that the application could be used to highlight the complexity and sophistication of 3D models. One specific example involved dynamically visualizing an airplane model to showcase its internal components. By modifying transfer functions and camera perspectives, users could create animations that gradually reveal key structures, helping to convey technical details in a visually appealing way.

In terms of desired functionality, a frequently mentioned request was the inclusion of cutting planes or X-ray-style views. Participants explained that while 1D transfer functions provide a level of control over what is visible, they are not always sufficient for isolating internal regions of interest. Being able to slice through or reveal specific cross-sections of a volume would enable more targeted exploration, especially when certain structures are obstructed by surrounding material.

Several of these suggestions were later taken into account and partially integrated into the framework. The feedback gathered during this phase played a critical role in refining the tool, helping it evolve into a more complete and user-centered system that better addresses the practical needs and expectations of its users.

5.3 Evaluation Design for Perceptually Uniform Transfer Functions

Regarding the subjective evaluation, we organized a group study with approximately 20 participants, including both individuals experienced with 1D transfer functions and those without prior knowledge. Participants were invited to explore our framework and then respond to a series of questions regarding their experience. The study utilized two volumetric datasets: one featuring a human head and a mysterious backpack from the VTK (Visualization Tool Kit) sample database, and another realistic dataset sourced from the volumetric organ database [Wal+24].

In the first phase, designed to familiarize participants with 1D transfer functions, subjects were provided with the first volumetric dataset and instructed to isolate the skull, assigning it a bone-like color, and to color the internal organs of the head in blue. Why blue? Experimental observations revealed that blue offers strong discrimination between different color spaces. Following this task, participants were asked to create a transparent effect on the outer skin. After completing their initial version, they were then invited to adjust their results to more closely match the reference shown in Figure 5.2a, with time allocated for this adjustment. By the end of the first phase, users had gained a superficial understanding of how the transfer function interacts with the volumetric data. It is important to note that no in-depth knowledge of the technical details or the implementation of the volumetric pipeline was required to achieve sufficient results.

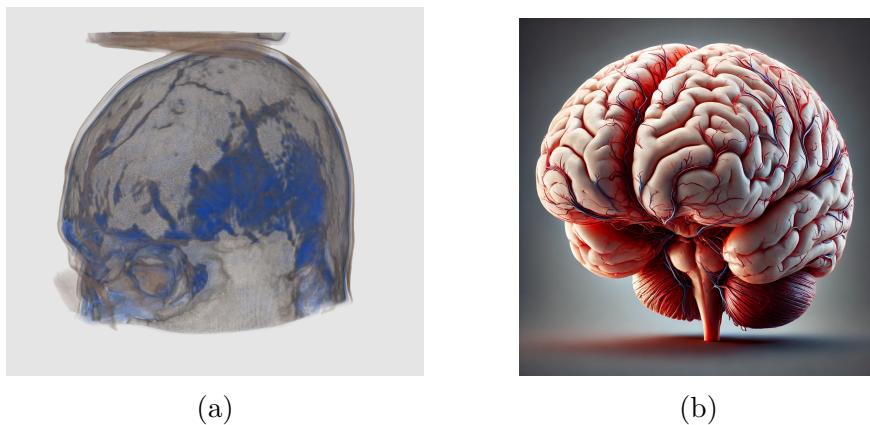


Figure 5.2: (a) Image used in Phase 1 to familiarize participants with 1D transfer functions, guiding them to fine-tune the image and arrive at a desired outcome. (b) Reference brain image that participants aim to replicate by adjusting the transfer function [AI25].

5 Results & Evaluations

For the second phase, we present the user with an image of a brain, identical to the one shown in Figure 5.2b, and ask them to adjust the transfer function to make the image resemble the reference. The user is given the freedom to take as much time as needed to refine the image until they are satisfied with the result or no longer wish to make changes. The time taken to produce the image is recorded. Participants were also informed that the reference image represented an ideal that could not be perfectly achieved. If they felt stuck or found the task too difficult to continue, they were free to stop their progress and move on to the next step. The resulting images from this experiment are displayed in Figure 5.1.

The key aspect of these two phases was that users constructed their own rendered images, which were later used in the third phase to evaluate the impact of perceptually uniform color spaces. Specifically, at the end of each phase, screenshots of the rendered images were captured using the respective transfer functions mapped to the CIERGB, CIELAB, PROLAB, and "CIEDE2000" color spaces. This approach ensured that the users themselves defined the transfer functions, removing potential biases from the authors of this thesis. However, it also introduced a double-edged outcome: while eliminating author bias, it allowed the possibility that some transfer functions might not reveal significant differences across color spaces. Additionally, one could argue that since the transfer functions were originally constructed using the CIERGB color space, the resulting rendered images were, in a sense, naturally optimized for that space. In other words, direct comparisons between screenshots might not clearly reveal whether one color space offers a better perceptual quality than another. However, since the editing process of transfer functions is limited to adjusting color and opacity control points, we argue that if users consistently preferred the visualizations produced with a different color space, this would still suggest that the alternative space provides a superior perceptual outcome overall.

Before, during, and after the experiment, several questions are posed to the subjects to gather insights for future analysis. These questions can be found in the table 5.2 and aim to better understand their background, experience, challenges faced, and preferences regarding the different color interpolation techniques and their overall use of the transfer function framework.

Before Phase 1		Answer format
	How familiar are you with volumetric data and its visualization techniques?	From very unfamiliar to very familiar
	How would you rate your familiarity with transfer functions?	From very unfamiliar to very familiar
Before Phase 2		
	How comfortable are you freely using transfer functions?	From not comfortable to very comfortable
	How challenging was it to achieve the reference image while adjusting the transfer function?	Very difficult to very easy
Before Phase 3		
	Using this real dataset, how challenging was it to adjust the transfer function in order to achieve the desired result?	Very difficult to very easy
	Do you have any feedback or comments regarding the experiment?	Open answer

Table 5.2: Table of questions asked to participants during the experiment.

It is also important to emphasize that all participants received a tutorial before engaging with the transfer function editor. The purpose of this tutorial was to ensure that users were familiar with the available tools and interface interactions without influencing their creative decisions. To maintain objectivity, the tutorial was carefully designed to be neutral in tone, focusing exclusively on the functionalities of the system rather than prescribing specific workflows or preferred outcomes. During the tutorial, participants were guided through essential operations such as zooming in and out of the line chart for precise adjustments, modifying the position of control points to influence color and opacity mappings, and utilizing the histogram to identify regions of interest within the dataset. Additionally, users learned how to add and remove control points as needed, as well as how to navigate the preview window freely by adjusting the camera position. This interactive preview allowed them to observe the immediate visual effects of their

5 Results & Evaluations

changes in real time. The interface presented during this instructional phase is depicted in Figure 5.3, which illustrates the components of the tutorial that supported users in becoming comfortable with the editor before beginning their tasks.

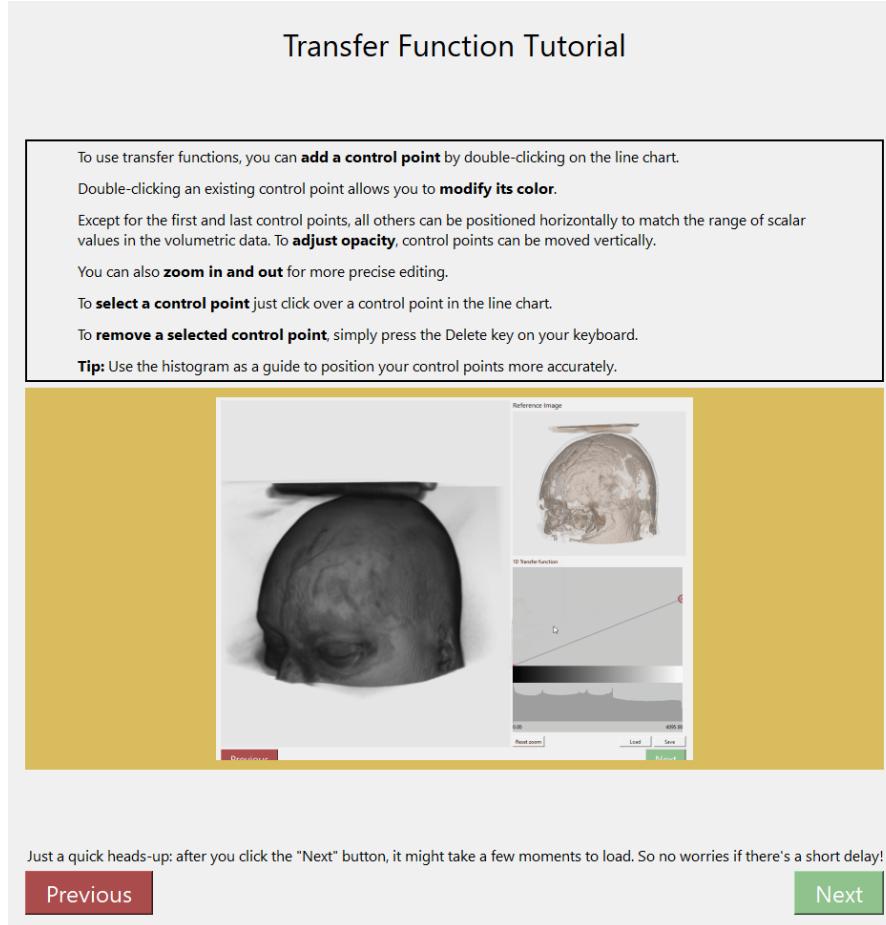


Figure 5.3: Tutorial page used in the perceptual uniformity user study for volume rendering transfer function evaluation.

To objectively evaluate uniformity, we rely on the set of metrics introduced in the paper by Bujack et al. [Buj+18]. This work consolidates terminology and methods from various sources that deal with perceptual aspects of color. Among the most relevant properties for our study are discriminative power, uniformity, and classical statistical measures based on local color distance calculations. These metrics were previously outlined in Chapter 3. For both phases of the study, we analyzed the participants' transfer function designs using these objective criteria. A summary and discussion of the metric results will follow, providing insights and proposing hypotheses for future research.

It is important to emphasize that the conclusions drawn from this analysis are limited by the small sample size of the study. As such, they should be interpreted as preliminary observations rather than statistically significant findings. A larger and more diverse participant group would be necessary to draw robust, generalizable conclusions. The participants in this study included university students, medical professionals, practitioners, PhD candidates, individuals of different genders, and some with color vision deficiencies such as color blindness.

As previously mentioned, one strategy we adopted to assess perceptual differences between color spaces was to incorporate user-generated results into the study. This idea was central to the design of the third phase of our user study. In this phase, participants were presented with pairs of images and asked to judge the quality of each rendering for later analysis. But what defines the "quality" of a 3D render in this context? For our purposes, it was based on factors such as good contrast between colors selected for the transfer function, color coherence that produced visually appealing results, or simply an intuitive sense that one image looked better. While this last criterion is subjective and difficult to describe, it plays an important role in human perception and decision-making.

Participants compared images, two at a time, each generated using different color spaces. For each pair, they were asked whether the images appeared identical or if one was noticeably better than the other. Simply placing images side by side, however, proved insufficient, as the differences were often subtle. To address this, we employed a method where images were superimposed, allowing participants to switch between them using a button. This approach made even minor differences much easier to detect and evaluate. Finally, participants were asked to comment on the difficulty of distinguishing between images and to describe the main differences they noticed during the comparisons. These observations will be discussed in detail in the results section.

5.4 Evaluation Results regarding Perceptually Uniform Transfer Functions

As a result of exploring the influence of perceptual color spaces on volume rendering, we are able to analyze the collected data to both support our hypothesis and identify

5 Results & Evaluations

opportunities for improving the video export framework. The user study offered valuable insights into the role of color appearance models in shaping user experience and visualization quality. The gathered data can be grouped into three primary categories:

- Personal background data: This includes physiological and academic information about the participants. Understanding this background helps contextualize the results and informs the interpretation of user behavior and performance.
- Qualitative feedback: This consists of open-ended responses and general impressions of the tool. Such feedback is essential for identifying usability issues, gathering suggestions for improvement, and evaluating the overall user experience.
- Technical performance data: Collected during the three phases of the transfer function design study, this data includes screen recordings, user preferences, and task completion times. This category is critical for evaluating the effectiveness of perceptual color models in volume rendering. It also enables a practical assessment of how traditional 1D transfer function design performs across varying levels of user expertise.

The participants in our user study were predominantly individuals with engineering and academic backgrounds, with the exception of one medical student. The age range of respondents varied between 18 and 55 years. Notably, one participant with color blindness took part in the study; their results did not significantly differ from those of other participants. Among those who consented to share device information, most used relatively modern monitors equipped with dynamic contrast adjustment features, which may have influenced their visual assessments.

In general, participants were initially unfamiliar with transfer functions. However, following the first experimental phase, they demonstrated increased confidence in manipulating transfer functions to explore volumetric datasets. This improvement is subtly reflected in Figure 5.4, which presents a boxplot of the time spent on each phase of the user study. Phase 1 involved familiarization with transfer functions. Phase 2 required their application on a real-world dataset. Phase 3 focused on image differentiation and was not directly related to the previous phases. It is worth highlighting that although the second dataset was considerably more complex, participants completed the task more quickly, suggesting a learning effect and growing comfort with the interface and tools.

5 Results & Evaluations

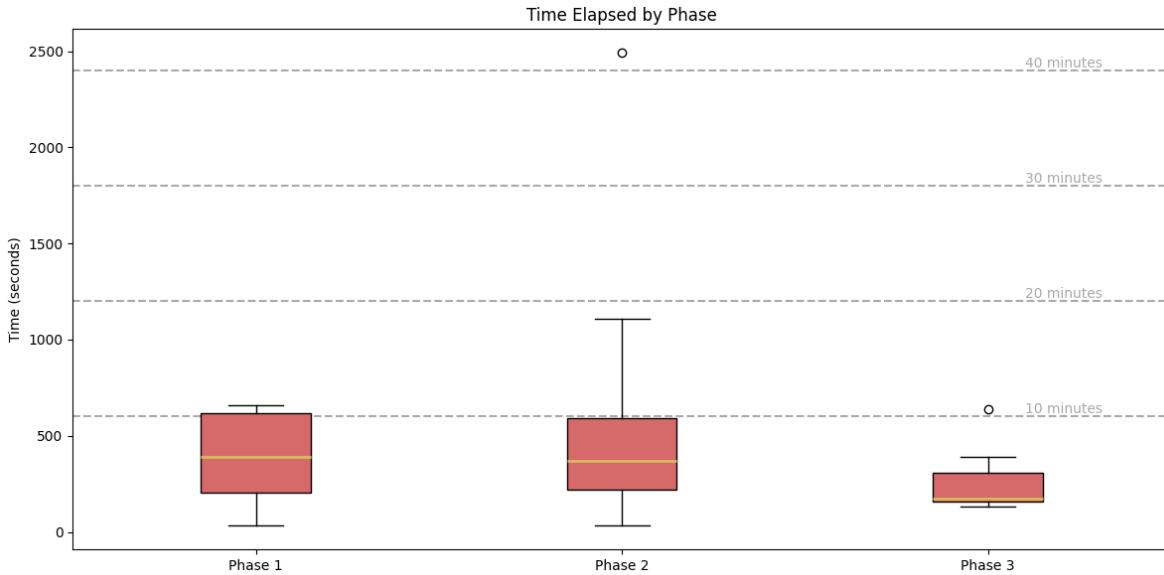


Figure 5.4: Time spent by participants across the three phases of the user study on perceptually uniform color spaces in transfer function design.

Using a Bradley-Terry model [BT52], we can estimate the relative strength or preference of each color space based on pairwise user comparisons. The Bradley-Terry model is a probabilistic framework commonly used to model outcomes of pairwise comparisons, where each item (in this case, a rendered image based on a specific color space) is assigned a score that reflects its perceived quality or preference level. The probability that one item is preferred over another is then determined by the relative scores of the two items. This method is particularly effective for subjective evaluation scenarios, as it allows us to convert qualitative user judgments into a quantitative ranking system, even with a limited number of comparisons. By applying this model, we can rank the rendered images according to user preferences within the scope of our study, providing a statistically grounded way to evaluate the perceptual effectiveness of different color spaces. Furthermore, by saving and analyzing the transfer functions created by participants, we uncover trends and common strategies used in designing color mappings. These insights offer valuable guidance for the development of future machine learning models that aim to assist or automate transfer function design, potentially improving usability and visual quality in volume rendering applications.

During the user study, we observed that a significant proportion of the comparisons made by participants resulted in ties, where the rendered images were judged to be of equal quality. Approximately one-third of all collected comparison data were marked

5 Results & Evaluations

as equal, indicating that participants often found it difficult to distinguish a clear preference between two renderings. This high incidence of ties underscores the limitations of conventional pairwise comparison models, such as the standard Bradley-Terry model, which assumes that one option is always preferred over another. To interpret the results more accurately, it becomes necessary to consider an extended or alternative model that can account for ties. Consider a simplified scenario involving three rendered images. Suppose we collect the following comparison outcomes:

- A is preferred over B
- B is preferred over C
- B is marked as equal to C twice

If we ignore the ties, the model would infer that the difference in quality between A and B is the same as between B and C. However, the tie data clearly suggests that B and C are quite similar, a nuance lost when ties are not accounted for. Some extended models address this by assigning a partial value to ties. For example, a tie may be treated as half a win, which is a common approach in sports statistics. While this method captures tie information more effectively, it raises the question of how much weight a tie should carry. The choice of weighting can influence the conclusions drawn from the data. The values obtained using Bradley-Terry with ties model are displayed in the table 5.3.

	CIERGB	CIELAB	PROLAB	CIEDE2000
Bradley Terry	1.900	0.704	1.141	0.655
Bradley Terry with Ties	0.777	0.472	1.131	0.480

Table 5.3: Strength scores computed using the Bradley-Terry model and its variant that accounts for ties, showing the relative preference levels for each color space based on user study comparisons.

To demonstrate the differences between using the standard Bradley-Terry model, ignoring ties, and the enhanced version that incorporates them, we present below the conclusions based on the computed strength scores of the evaluated color spaces. The conclusions of both studies are summarized below.

- Bradley-Terry model:

$$\text{CIERGB} > \text{PROLAB} > \text{CIELAB} > \text{CIEDE2000}$$

- Bradley-Terry with ties model:

PROLAB > CIERGB > CIEDE2000 > CIELAB

The Bradley-Terry model with ties, as introduced by Torsney et al. [Tor04] was implemented using the bratt function from the VGAM (Vector Generalized Linear and Additive Models) package in R [Doc25]. The data analyzed from Phase 3 of the user study is explicitly enumerated in Tables 5.4 and 5.5.

	CIERGB	CIELAB	PROLAB	CIEDE2000
CIERGB	-	6	6	6
CIELAB	11	-	13	10
PROLAB	14	4	-	9
CIEDE2000	17	13	13	-

Table 5.4: Image preference matrix showing the number of times each row color space was preferred over each column color space, based on results from Phase 3 of the transfer function color perception user study.

	CIERGB	CIELAB	PROLAB	CIEDE2000
CIERGB	-	12	9	6
CIELAB	12	-	12	6
PROLAB	9	12	-	9
CIEDE2000	6	6	7	-

Table 5.5: Matrix of ties indicating how often participants selected both images as equal during Phase 3 of the transfer function color perception user study.

With the results from the perceptually uniform color spaces user study now available, we can compare them against our objective metrics to address the research question outlined in **(Q1.3) How do color interpolations in perceptually uniform spaces compare to traditional linear RGB interpolation when used in animations?**. By focusing on the top-performing color space model, ProLAB, and comparing it with CIERGB, we assess their effectiveness using the previously defined objective metrics. For the distance difference chart, we apply a slight modification that aggregates all user-defined distances from the Phase 1 transfer functions, producing the chart presented in Figure 5.5. The details of how this chart was constructed are explained below.

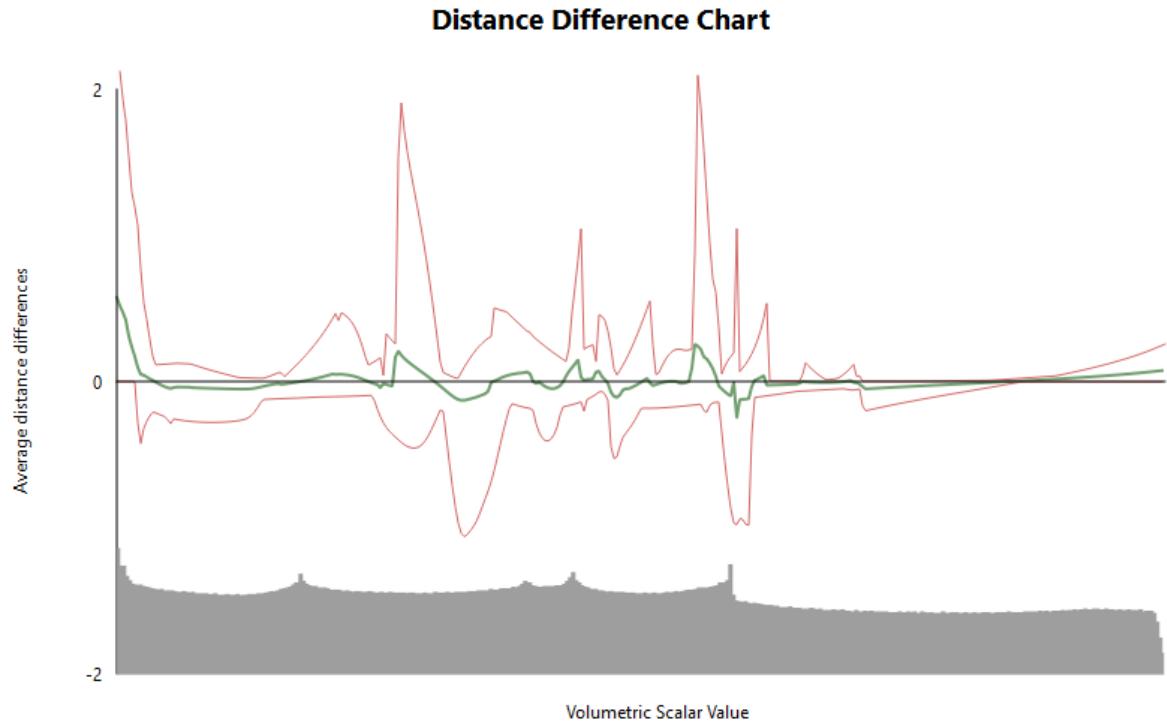


Figure 5.5: Aggregated color distance difference chart between ProLAB and CIERGB. The y-axis shows the difference in color distance (ΔE_{2000}) between two consecutive nodes, computed as ProLAB minus CIERGB.

This chart was constructed iterating over all volumetric scalar values and their successive neighboring values across the entire domain of the volumetric data. For each scalar value and its neighbor, we map it to a color using the user-defined transfer functions, which were collected during the user study. Then, for each neighboring pair, we compute the color difference using the ΔE_{2000} metric. This calculation is performed separately for both color spaces, ProLAB and CIERGB. Then, we compute the difference between the color distances obtained from the two color spaces, as shown in Equation 5.1. As ΔE_{2000} is directly related to the ability to distinguish between colors, differences in this metric across color spaces can be associated with the capacity of one space to reveal more visual detail than another.

$$CDD = \Delta E_{2000}^{ProLAB} - \Delta E_{2000}^{CIERGB} \quad (5.1)$$

Using this color distance difference, we aggregate the results across all user-defined transfer functions. The green line in the chart represents the average of these differences, while the red lines indicate the minimum and maximum difference values observed across

5 Results & Evaluations

the different transfer functions. A simple yet effective interpretation of this chart is the following: if a point lies above the x-axis, it indicates better color distinguishability for ProLAB when compared to CIERGB. To support our visual analysis, we deduced 2 values: the average local discriminative power and the integral of the green line in the aggregated color distance difference chart. The integral quantifies the area under the green curve, indicating how much of the chart lies in the positive or negative region. These values are presented in Table 5.6.

Average Discriminative Power	1.448
Distance Color Difference Integral	2.128

Table 5.6: Average discriminative power and integral of the distance difference for all user-defined transfer functions in the study comparing ProLAB and CIERGB.

These results align with our expectations; however, to ensure consistency with real-world conditions, the model requires slight adjustments. Without these modifications, discrepancies may arise. For example, by selecting the best and worst performing color space models, namely ProLAB and CIELAB, we can compare their performance using the previously defined objective metrics. This comparison is illustrated in Table 5.7 and Figure 5.6.

Average Discriminative Power	-1.399
Distance Color Difference Integral	-1.252

Table 5.7: Average discriminative power and integral of the distance difference for all user-defined transfer functions in the study.

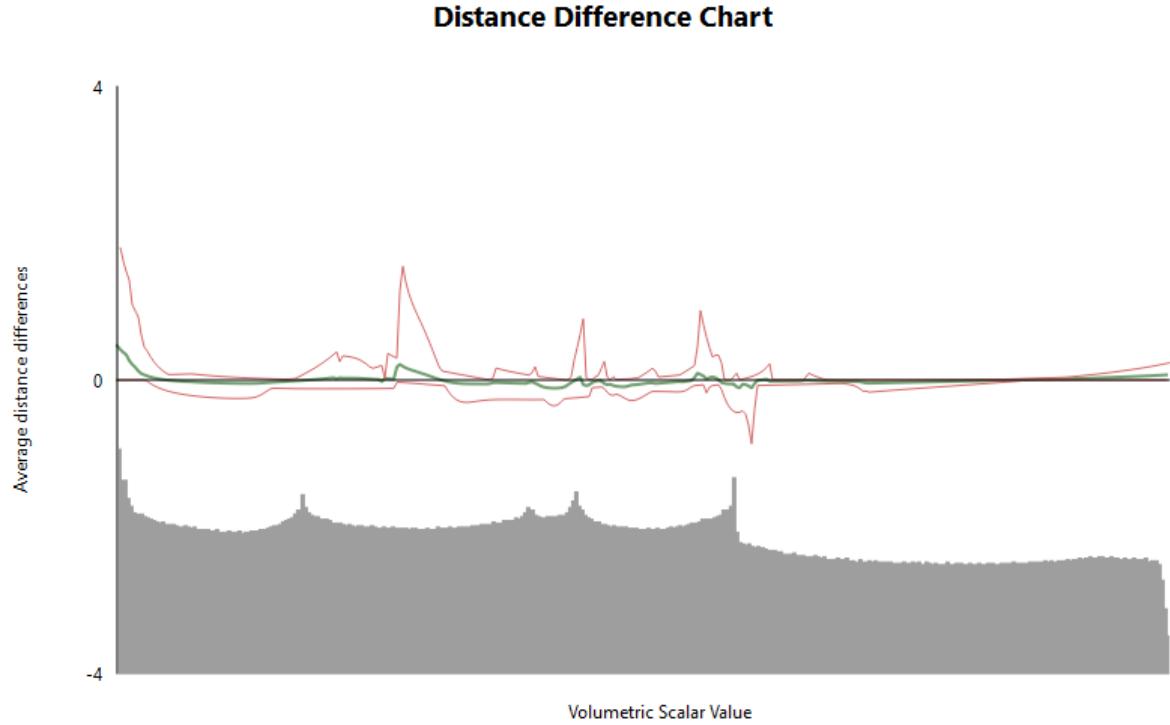


Figure 5.6: Aggregated color distance difference chart between ProLAB and CIELAB. The y-axis shows the difference in color distance (ΔE_{2000}) between two consecutive nodes, computed as ProLAB minus CIELAB.

Surprisingly, both the discriminative power and the integral of the green line favor the worst color space (CIELAB). But why is this the case? Should not the opposite be true? To understand the source of this discrepancy, we need to recognize that the color perception formulas currently take into account the entire spectrum of the volumetric data, which is incorrect. Not all parts of the transfer function domain contribute to the final image. In fact, certain areas of the color palette are completely invisible in the final render. To address this, we can incorporate the opacity by multiplying the color distances by the opacity values. This will allow us to weigh the results according to how much each part of the color spectrum is visible in the final image. This modification is formally represented in Equation 5.2.

$$\bar{v} := \sum_{j=1}^n d_j o_j \quad (5.2)$$

where o_j represents the opacity associated with the color at position j in the transfer

5 Results & Evaluations

function mapping.

To summarize, by incorporating the concept of opacity into the previously defined objective metrics, Table 5.8 presents the corresponding values with the appropriate weights applied during their calculation. Additionally, Figure 5.7 illustrates the impact of accounting for opacity when measuring color distances, highlighting how it influences the evaluation of discriminative power.

Average Discriminative Power	0.0722043
Distance Color Difference Integral	0.0120405

Table 5.8: Opacity-weighted average discriminative power and integral of the color distance difference for all user-defined transfer functions in the study.

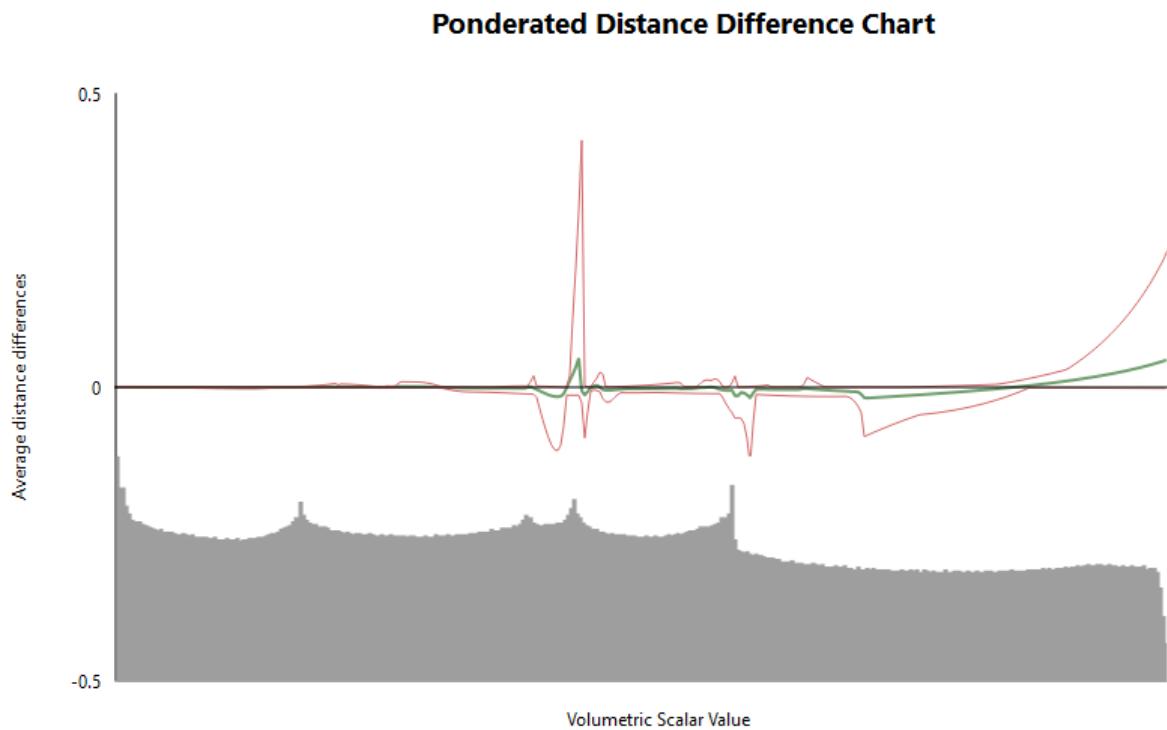


Figure 5.7: Illustration of the color distance difference chart, incorporating the opacity defined by the transfer functions.

As we can see, the values are now significantly smaller, which is expected since the opacity ranges from 0 to 1. However, the key observation is that the values are positive. This provides objective evidence of the improvement in perceptual clarity when using ProLAB as the color space instead of CIELAB in our transfer function.

6 Discussion

Storytelling reveals meaning without committing the error of defining it.

Hannah Arendt, *Men in Dark Times*

Based on our results, we could show that our video export framework effectively addresses the initial problems identified and provides concrete solutions that can support future research and applications. Participants expressed overall satisfaction with the functionalities offered and even requested additional features that could further benefit their respective communities. These ranged from human-centered applications like surgical planning to more strategic uses in funding presentations and commercial communication.

Regarding path planning methods, quaternion-based interpolation was preferred for exploring large objects where the path involved frequent orientation changes but minimal translation. However, this technique was less effective when combining translation and rotation, leading to discouraging results. In contrast, the method using view-up vectors and focal points produced smoother and more visually appealing animations, especially when orientation changes were moderate. In cases of sharp directional shifts, however, users found the results less convincing. For orbit-like animations, fixing the focal point at the origin significantly simplified the problem, resulting in near-perfect interpolation videos as long as the orientation remained relatively stable.

Concerning the use of perceptually uniform color spaces and objective metrics to improve image quality, our study provides subtle but promising evidence that these spaces can enhance visual clarity in volume rendering. Surprisingly, models like CIELAB and CIEDE2000 did not perform as well as anticipated. In the case of CIELAB, this aligns with prior criticisms in the literature, which highlight its failure to consistently uphold

6 Discussion

perceptual uniformity. Although CIELAB is still widely regarded as a perceptually uniform model by standard definitions and general acceptance, our findings, along with previous research, demonstrate that it is outperformed in comparison to more recent and advanced color appearance models. The underperformance of CIEDE2000 "color space", however, was more unexpected. Although this model was specifically designed to align with perceptual uniformity, participants in our study often preferred visuals generated using the more traditional CIERGB space. This discrepancy may arise from external influences such as background color, lighting conditions, or display hardware, all factors that can affect perceptual judgment. Another plausible explanation is the small sample size in our user study. With only a few participants, subtle perceptual differences may not have been captured accurately. A larger number of subjects in our study, combined with a more controlled environment and better-calibrated equipment, could explain our results. It is also possible that issues in our implementation of the algorithm used to construct the CIEDE2000 "color space" contributed to the unexpected results. Although the method may have effectively reduced perceptual distances within the color palette, its impact could have been diminished during interpolation of mid-range volumetric values or when combined with varying opacity levels.

7 Conclusion

I would rather have questions that can't be answered than answers that can't be questioned.

Richard Feynman

The primary goal of this thesis was to develop a robust research framework to support various domains in the context of video export from volumetric rendering. In particular, it aimed to assist research activities associated with the Fraunhofer group, spanning fields from medicine to materials science and mechanical applications. To achieve this, we investigated several techniques for camera path planning and introduced a novel connection between perceptually uniform color spaces and volumetric rendering. This work addressed four main research questions:

(Q1) How can perceptually uniform color palettes be used in Volume rendering?

With (Q1), we aimed to address a significant gap in the existing literature concerning the application of perceptually uniform color spaces in volume rendering. These color spaces enhance perceptual consistency by ensuring that equal distances in the color space correspond to roughly equal perceptual differences, which is crucial for producing visually coherent interpolations. In particular, the use of the ProLAB color space demonstrated notable improvements in the perception of fine details and smoother, more consistent color transitions throughout volumetric data. The effectiveness of this approach was first confirmed through objective metrics and subsequently supported by a subjective user study, providing strong evidence for its practical benefits. This research underscored the visual enhancements that perceptually uniform color spaces offer, especially by reducing undesirable color shifts and tones that often occur during color interpolations. When compared directly to the traditional CIERGB color space, ProLAB consistently

outperformed it in user preference tests involving 3D rendered images, highlighting its potential to improve the quality and clarity of volume rendering applications.

(Q2) How can interpolation between control points for camera position and orientation ensure smooth and visually appealing animation?

Question (Q2) addressed methods for achieving smooth interpolation of camera paths, covering both position and orientation aspects. The chosen strategy for defining control points in 3D space involved allowing the user to freely control the camera and save its state whenever desired for subsequent interpolation. This approach balanced ease of user interaction with the quality of the resulting camera trajectory. Catmull-Rom interpolation was selected due to its simplicity and ability to generate smooth curves from only this specific user-defined control points. For interpolating orientation, two techniques were evaluated: one based on quaternions and another utilizing view-up and focal point vectors. Each method presented specific strengths and weaknesses depending on the scenario. Furthermore, to support the highly requested feature of pausing the camera, the interpolation method was refined using monotonic cubic interpolation, which effectively prevented unwanted oscillations when control points were placed closely together to simulate the pause effect.

(Q3) How can transfer function interpolations be executed to produce smooth, semantically coherent, and visually appealing animations?

Question (Q3) inspired the creation and assessment of a new algorithm designed specifically for interpolating transfer functions. This approach facilitates smooth transitions between one-dimensional transfer functions, ensuring that the visual context is preserved throughout the process. By maintaining this continuity, the algorithm enhances immersion during animated volume visualizations, making the experience more engaging for users. Additionally, it supports clearer communication and more effective storytelling when presenting volume-rendered sequences, allowing viewers to better understand the underlying data and narrative.

Each research question was carefully crafted with the goal of providing fresh and meaningful insights into the integration of perceptually uniform color spaces within the context of volumetric rendering. This integration is crucial for improving the visual fidelity and clarity of rendered volumetric data, which in turn enhances the interpretability and usability of these visualizations across various scientific fields. At the same time, these questions also served to establish a strong and versatile foundation for the development

of a video export framework that is not only flexible but also powerful enough to meet diverse user needs. Throughout this thesis, a broad range of techniques related to color space manipulation, transfer function interpolation, and camera trajectory generation were systematically explored, rigorously evaluated, and thoughtfully compared. These efforts collectively contribute to an advancement in the state of volumetric visualization, pushing the boundaries of how volume-rendered data can be presented, analyzed, and communicated effectively. The results not only deepen our understanding of perceptual color integration but also pave the way for future innovations in volumetric data visualization and interactive media production.

7.1 Limitations and future work

Throughout the development of this thesis, several key limitations and challenges emerged, shaping the scope and direction of the research. This section highlights these obstacles, reflecting on areas where the work could be improved or refined. By acknowledging these constraints, we identify opportunities for future exploration. The discussion covers technical limitations in algorithms, constraints in user study design such as sample size and diversity, and issues with generalizing the results. Additionally, we propose potential directions for future research to address these challenges through improved methodologies, advanced techniques, and broader interdisciplinary collaboration.

One of the main issues encountered during camera path interpolation using both position and quaternions for orientation was wobbling effects, especially when the camera's focus changes frequently. A possible solution lies in linking orientation and position by leveraging the curvature of the path. Specifically, correlating the curvature with both the trajectory's speed and the angular velocity of the camera orientation could reduce this undesired wobble and produce smoother motion.

Another area for exploration involves expanding the use of transfer functions beyond the 1D models employed in this thesis. Future work could build upon this foundation to support 2D transfer functions, allowing for more complex and expressive visual mappings by including both intensity and gradient information, for example.

Finally, this thesis only examined a small subset of perceptually uniform color spaces. Many emerging color appearance models, which account for factors like background,

7 Conclusion

luminance, and ambient lighting, could offer more perceptually consistent results. Future research could evaluate the effectiveness of these models under various viewing conditions, perhaps even adapting them dynamically based on environmental inputs.

A Color Conversions

A.1 CIERGB and CIEXYZ

The conversion from RGB to XYZ is given by the following matrix transformation:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.2404542 & -1.5371385 & -0.4985314 \\ -0.9692660 & 1.8760108 & 0.0415560 \\ 0.0556434 & -0.2040259 & 1.0572252 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (\text{A.1})$$

The conversion from XYZ to RGB is given by the following matrix transformation:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124564 & 0.3575761 & 0.1804375 \\ 0.2126729 & 0.7151522 & 0.0721750 \\ 0.0193339 & 0.1191920 & 0.9503041 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{A.2})$$

The equations for converting between CIERGB and CIEXYZ color spaces (A.1 and A.2) should be used with each RGB and XYZ channel ranging from 0 to 1.

A.2 CIEXYZ and CIELUV

The conversion from CIEXYZ to CIELUV is expressed as follows:

$$L = \begin{cases} 116\sqrt[3]{\frac{Y}{100}} - 16, & \text{if } \frac{Y}{100} > 0.008856 \\ 9.033Y & \text{otherwise} \end{cases} \quad (\text{A.3})$$

$$u = 13L \left(\frac{4X}{X + 15Y + 3Z} - 0.1978 \right) \quad (\text{A.4})$$

A Color Conversions

$$v = 13L \left(\frac{9Y}{X + 15Y + 3Z} - 0.4683 \right) \quad (\text{A.5})$$

The conversion from CIELUV to CIEXYZ can be expressed as follows:

$$X = \frac{d + 5Y}{a - \frac{1}{3}} \quad (\text{A.6})$$

$$Y = \begin{cases} \left(\frac{L+16}{116}\right)^3 & \text{if } L > 7.9996 \\ \frac{L}{903.3} & \text{otherwise} \end{cases} \quad (\text{A.7})$$

$$Z = aX - 5Y \quad (\text{A.8})$$

where :

$$a = \frac{1}{3} \left(\frac{52L}{u + 2.5714L} - 1 \right) \quad d = Y \left(\frac{39L}{v + 6.0879L} - 5 \right) \quad (\text{A.9})$$

A.3 CIEXYZ and CIELAB

The conversion from CIEXYZ to CIELAB is given by the following formula [Lin17]:

$$L = 116f_y - 16 \quad (\text{A.10})$$

$$a = 500(f_x - f_y) \quad (\text{A.11})$$

$$b = 200(f_y - f_z) \quad (\text{A.12})$$

where :

$$f_x = \begin{cases} \sqrt[3]{x_r} & \text{if } x_r > 0.008856 \\ \frac{903.3x_r+16}{116} & \text{otherwise} \end{cases} \quad (\text{A.13})$$

$$f_y = \begin{cases} \sqrt[3]{y_r} & \text{if } y_r > 0.008856 \\ \frac{903.3y_r+16}{116} & \text{otherwise} \end{cases} \quad (\text{A.14})$$

$$f_z = \begin{cases} \sqrt[3]{z_r} & \text{if } z_r > 0.008856 \\ \frac{903.3z_r+16}{116} & \text{otherwise} \end{cases} \quad (\text{A.15})$$

A Color Conversions

And the corresponding transformations of X , Y , and Z under the D65 illuminant are given by:

$$x_r = \frac{X}{95.047} \quad y_r = \frac{Y}{100} \quad z_r = \frac{Z}{108.883} \quad (\text{A.16})$$

The inverse conversion from CIELAB to CIEXYZ is given by the following formula:

$$X = 95.047x_r \quad Y = 100y_r \quad Z = 108.883z_r \quad (\text{A.17})$$

where :

$$x_r = \begin{cases} f_x^3 & \text{if } f_x^3 > 0.008856 \\ \frac{116f_x - 16}{903.3} & \text{otherwise} \end{cases} \quad (\text{A.18})$$

$$y_r = \begin{cases} f_y^3 & \text{if } L > 7.9996 \\ \frac{L}{903.3} & \text{otherwise} \end{cases} \quad (\text{A.19})$$

$$z_r = \begin{cases} f_z^3 & \text{if } f_z^3 > 0.008856 \\ \frac{116f_z - 16}{903.3} & \text{otherwise} \end{cases} \quad (\text{A.20})$$

$$f_x = \frac{a}{500} + f_y \quad f_z = f_y - \frac{b}{200} \quad f_y = \frac{L + 16}{116} \quad (\text{A.21})$$

B Color Distances

B.1 $\Delta E_{CIE1994}$

$$\Delta E_{CIE1994} = \sqrt{(\Delta L)^2 + \left(\frac{\Delta C}{1 + 0.045C_1}\right)^2 + \left(\frac{\Delta H}{1 + 0.015C_1}\right)^2} \quad (\text{B.1})$$

where :

$$\Delta L = L_1 - L_2 \quad (\text{B.2})$$

$$\Delta C = C_1 - C_2 \quad C_1 = \sqrt{a_1^2 + b_1^2} \quad C_2 = \sqrt{a_2^2 + b_2^2} \quad (\text{B.3})$$

$$\Delta H = \sqrt{\Delta a^2 + \Delta b^2 + \Delta C^2} \quad \Delta a = a_1 - a_2 \quad \Delta b = b_1 - b_2 \quad (\text{B.4})$$

B.2 $\Delta E_{CIE2000}$

$$\Delta E_{CIE2000} = \sqrt{\left(\frac{\Delta L'}{S_L}\right)^2 + \left(\frac{\Delta C'}{S_C}\right)^2 + \left(\frac{\Delta H'}{S_H}\right)^2 + R_T \left(\frac{\Delta C'}{S_C}\right) \left(\frac{\Delta H'}{S_H}\right)} \quad (\text{B.5})$$

where :

$$\Delta L' = L_2 - L_1 \quad (\text{B.6})$$

$$S_L = 1 + \frac{0.015(\bar{L}' - 50)^2}{\sqrt{20 + (\bar{L}' - 50)^2}} \quad \bar{L}' = \frac{(L_1 + L_2)}{2} \quad (\text{B.7})$$

$$\Delta C' = C'_2 - C'_1 \quad C'_1 = \sqrt{a'^2_1 + b'^2_1} \quad C'_2 = \sqrt{a'^2_2 + b'^2_2} \quad (\text{B.8})$$

B Color Distances

$$a'_1 = a_1(1 + G) \quad a'_2 = a_2(1 + G) \quad (\text{B.9})$$

$$G = \frac{1}{2}(1 - \sqrt{\frac{\bar{C}^7}{\bar{C}^7 + 25^7}}) \quad (\text{B.10})$$

$$\bar{C} = \frac{C_1 + C_2}{2} \quad C_1 = \sqrt{a_1^2 + b_1^2} \quad C_2 = \sqrt{a_2^2 + b_2^2} \quad (\text{B.11})$$

$$S_C = 1 + 0.045\bar{C}' \quad \bar{C}' = \frac{C'_1 + C'_2}{2} \quad (\text{B.12})$$

$$\Delta H' = 2\sqrt{C'_1 C'_2} \sin(\frac{\Delta h'}{2}) \quad (\text{B.13})$$

$$\Delta h' = \begin{cases} h'_2 - h'_1 & \text{if } |h'_2 - h'_1| \leq 180^\circ \\ h'_2 - h'_1 + 360^\circ & \text{if } |h'_2 - h'_1| > 180^\circ \text{ and } h'_2 \leq h'_1 \\ h'_2 - h'_1 - 360^\circ & \text{otherwise} \end{cases} \quad (\text{B.14})$$

$$h'_1 = \begin{cases} \arctan(\frac{b_1}{a'_1}) & \text{if } \arctan(\frac{b_1}{a'_1}) \geq 0 \\ \arctan(\frac{b_1}{a'_1}) + 360^\circ & \text{otherwise} \end{cases} \quad (\text{B.15})$$

$$h'_2 = \begin{cases} \arctan(\frac{b_2}{a'_2}) & \text{if } \arctan(\frac{b_2}{a'_2}) \geq 0 \\ \arctan(\frac{b_2}{a'_2}) + 360^\circ & \text{otherwise} \end{cases} \quad (\text{B.16})$$

$$S_H = 1 + 0.015\bar{C}'T \quad (\text{B.17})$$

$$T = 1 - 0.17 \cos(\bar{H}' - 30^\circ) + 0.24 \cos(2\bar{H}') + 0.32 \cos(3\bar{H}' + 6^\circ) - 0.20 \cos(4\bar{H}' - 63^\circ) \quad (\text{B.18})$$

$$\bar{H}' = \begin{cases} \frac{h'_1 + h'_2 + 360^\circ}{2} & \text{if } |h'_1 + h'_2| > 180^\circ \\ \frac{h'_1 + h'_2}{2} & \text{otherwise} \end{cases} \quad (\text{B.19})$$

B Color Distances

$$R_T = -R_C \sin(2\Delta\theta) \quad R_C = 2\sqrt{\frac{\bar{C}'^7}{\bar{C}'^7 + 25^7}} \quad (\text{B.20})$$

$$\Delta\theta = 30 \exp\left(-\left(\frac{\bar{H}' - 275^\circ}{25}\right)^2\right) \quad (\text{B.21})$$

Bibliography

- [Ada43] EQ Adams. “Chromatic valence as a correlate of Munsell chroma”. In: *Proceedings of the twenty-eighth annual meeting of the optical Society of America*. Vol. 33. 12. 1943, p. 683 (cit. on p. 14).
- [AI25] OpenAI ChatGPT DALL·E AI. *Prompt: Realistic 3D rendering of a human brain from front-to-back angle, with detailed folds, visible red and blue blood vessels, cerebellum and brainstem. Soft lighting, medical illustration style, high detail.* 2025 (cit. on p. 55).
- [Aig+23] Wolfgang Aigner et al. *Visualization of Time-Oriented Data*. en. Human Computer Interaction Series. London: Springer London, 2023. ISBN: 978-1-4471-7526-1. DOI: 10.1007/978-1-4471-7527-8. URL: <https://link.springer.com/10.1007/978-1-4471-7527-8> (cit. on pp. 41, 46).
- [Bir20] James Bird. *Monotone Cubic Interpolation*. 2020. URL: <https://www.jb101.co.uk/2020/12/27/monotone-cubic-interpolation.html> (cit. on p. 38).
- [BT52] Ralph Allan Bradley and Milton E. Terry. “Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons”. In: *Biometrika* 39.3/4 (1952), pp. 324–345. ISSN: 0006-3444. DOI: 10.2307/2334029 (cit. on p. 61).
- [Bui73] Tuong-Phong Bui. “Illumination of computer generated images”. In: *Department of Computer Science, University of Utah* (1973) (cit. on p. 19).
- [Buj23] Roxana Bujack. “*The Mathematics of Perceiving Color Differences*” by Roxana Bujack. Nov. 2023. URL: <https://www.youtube.com/watch?v=elueUkw02yg> (cit. on pp. 3, 43).

Bibliography

- [Buj+18] Roxana Bujack et al. “The Good, the Bad, and the Ugly: A Theoretical Framework for the Assessment of Continuous Colormaps”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (Jan. 2018), pp. 923–933. ISSN: 1941-0506. DOI: 10.1109/TVCG.2017.2743978 (cit. on pp. 27, 32, 58).
- [Buj+22] Roxana Bujack et al. “The non-Riemannian nature of perceptual color space”. In: *Proceedings of the National Academy of Sciences* 119.18 (2022), e2119753119. DOI: 10.1073/pnas.2119753119. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.2119753119>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.2119753119> (cit. on p. 28).
- [Buj+24] Roxana Bujack et al. “Efficient Computation of Geodesics in Color Space”. In: *IEEE Transactions on Visualization and Computer Graphics* 30.9 (Sept. 2024), pp. 6507–6519. ISSN: 1941-0506. DOI: 10.1109/TVCG.2023.3346673 (cit. on p. 29).
- [CR74] Edwin Catmull and Raphael Rom. “A CLASS OF LOCAL INTERPOLATING SPLINES”. In: *Computer Aided Geometric Design*. Ed. by ROBERT E. Barnhill and RICHARD F. Riesenfeld. Academic Press, Jan. 1974, pp. 317–326. ISBN: 978-0-12-079050-0. DOI: 10.1016/B978-0-12-079050-0.50020-5. URL: <https://www.sciencedirect.com/science/article/pii/B9780120790500500205> (cit. on p. 10).
- [Doc25] R Documentation. *R: Bradley terry model with ties*. 2025. URL: <https://search.r-project.org/CRAN/refmans/VGAM/html/bratt.html> (cit. on p. 63).
- [Gau17] Sharma Gaurav. *Digital color imaging handbook*. CRC press, 2017 (cit. on p. 17).
- [Hol23] Daniel Holden. *Cubic Interpolation of Quaternions*. 2023. URL: <https://theorangeduck.com/page/cubic-interpolation-quaternions> (cit. on pp. 37, 38, 40).
- [HMO06] Rafael Huertas, Manuel Melgosa, and Claudio Oleari. “Performance of a color-difference formula based on OSA-UCS space using small–medium color differences”. In: *Journal of the Optical Society of America A* 23.9 (2006), pp. 2077–2084 (cit. on p. 28).

Bibliography

- [Kon+21] Ivan A. Konovalenko et al. “ProLab: A Perceptually Uniform Projective Color Coordinate System”. In: *IEEE Access* 9 (2021), pp. 133023–133042. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3115425 (cit. on pp. 24, 25, 27).
- [KK24] Olga Kosheleva and Vladik Kreinovich. “Why Bernstein Polynomials: Yet Another Explanation”. en. In: *ScholarWorks@UTEP* (2024) (cit. on p. 10).
- [Li+02] Changjun Li et al. “The Performance of CIECAM02”. en. In: *Color and Imaging Conference* 10 (2002), pp. 28–32. ISSN: 2166-9635. DOI: 10.2352/CIC.2002.10.1.art00007 (cit. on p. 27).
- [Li+17] Changjun Li et al. “Comprehensive color solutions: CAM16, CAT16, and CAM16-UCS”. en. In: *Color Research & Application* 42.6 (2017), pp. 703–718. ISSN: 1520-6378. DOI: 10.1002/col.22131 (cit. on p. 24).
- [Lin17] Bruce Lindbloom. *Useful Color Equations*. 2017. URL: www.brucelindbloom.com/index.html (cit. on pp. 15, 75).
- [Lju+16] Patric Ljung et al. “State of the Art in Transfer Functions for Direct Volume Rendering”. en. In: *Computer Graphics Forum* 35.3 (2016), pp. 669–691. ISSN: 1467-8659. DOI: 10.1111/cgf.12934 (cit. on pp. 19, 20, 22, 28).
- [LCR01] M. R. Luo, G. Cui, and B. Rigg. “The development of the CIE 2000 colour-difference formula: CIEDE2000”. en. In: *Color Research & Application* 26.5 (2001), pp. 340–350. ISSN: 1520-6378. DOI: 10.1002/col.1049 (cit. on p. 16).
- [Mor03] Nathan Moroney. “A hypothesis regarding the poor blue constancy of CIELAB”. In: *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur* 28.5 (2003), pp. 371–378 (cit. on p. 25).
- [NAS] NASA/MSFC/GHRC. DOI: 10.5067/SEAFLUX/DATA101 (cit. on p. 3).
- [RF12] Dibakar Raj Pant and Ivar Farup. “Riemannian formulation and comparison of color difference formulas”. In: *Color Research & Application* 37.6 (2012), pp. 429–440 (cit. on p. 28).
- [Sey22a] John Seymour. “Color inconstancy in CIELAB: A red herring?” en. In: *Color Research & Application* 47.4 (2022), pp. 900–919. ISSN: 1520-6378. DOI: 10.1002/col.22782 (cit. on p. 26).

Bibliography

- [Sey22b] John Seymour. “Six Things You Should Not Do With CIELAB”. In: *Technical Association of the Graphic Arts* (Aug. 2022) (cit. on p. 26).
- [Sey24] John Seymour. “CIELAB is Icky! Do we have any other choice? Nov. 2024. URL: <https://www.youtube.com/watch?v=DnnFjPMc1oQ> (cit. on p. 26).
- [SG31] T. Smith and J. Guild. “The C.I.E. colorimetric standards and their use”. en. In: *Transactions of the Optical Society* 33.3 (Jan. 1931), p. 73. ISSN: 1475-4878. DOI: 10.1088/1475-4878/33/3/301 (cit. on p. 13).
- [Tor04] Ben Torsney. “Fitting Bradley Terry Models Using a Multiplicative Algorithm”. en. In: *COMPSTAT 2004 — Proceedings in Computational Statistics*. Ed. by Jaromir Antoch. Heidelberg: Physica-Verlag HD, 2004, pp. 513–526. ISBN: 978-3-7908-2656-2. DOI: 10.1007/978-3-7908-2656-2_42 (cit. on p. 63).
- [VTK25] VTK. *Examples Data*. Dataset. Version 1. 2025 (cit. on pp. 18, 21).
- [Wal+24] C. L. Walsh et al. *Complete scan at 19.57 um of the brain of the body donor S-20-29*. Dataset. Version 1. 2024. DOI: 10.15151/ESRF-DC-1773692859 (cit. on pp. 49, 55).
- [Zey+18] Max Zeyen et al. “Color Interpolation for Non-Euclidean Color Spaces”. In: *2018 IEEE Scientific Visualization Conference (SciVis)*. Oct. 2018, pp. 11–15. DOI: 10.1109/SciVis.2018.8823597. URL: <https://ieeexplore.ieee.org/document/8823597/?arnumber=8823597> (cit. on p. 29).

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich diese Masterarbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe und alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, als solche gekennzeichnet sind, sowie, dass ich die Masterarbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt habe.

Passau, June 11, 2025



Jorge Korgut Junior