VOLUME RENDERING OF HUMAN HAND ANATOMY

by

Jingtao Huang

A Thesis Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
MASTER OF SCIENCE
(COMPUTER SCIENCE)

August 2022

Copyright 2022 Jingtao Huang

Dedication

To my grandparents, in loving memory.

Acknowledgements

I would like to thank my advisor, Jernej Barbič, for his exemplary guidance, support, and encouragement during my master's degree. A special thanks to my friends Mianlun Zheng, Bohan Wang, Rongxin You, and Yuanhan Zhang for their inspiration and friendship. Finally, I thank my loving parents for their constant and unconditional support.

Table of Contents

List of Tables

List of Figures

Abstract

This thesis investigates volume rendering techniques and the design of transfer functions to improve the visualization of human hand anatomy using medical resonance imaging (MRI). Volume rendering of the human hand is a challenging but crucial problem for understanding hand anatomy in three dimensions. The hand is inherently complex, and it is difficult to design volume rendering transfer functions that are suitable for each specific application. We first introduce the fundamentals of volume rendering and hand anatomy, and discuss the related work on hand anatomy visualization. Then, we propose a ray-casting pipeline that uses both the hand MRI volume data, and the geometry of its inner structures and their known segmentation, permitting fine control over the appearance of each tissue. Additionally, we devise two sets of transfer functions to emphasize different hand tissues of interest, while preserving the visual context of the hand, inspired by the concept of "focus-plus-context." We also present pipeline extensions to tackle several artifacts of standard volume ray-casting. We evaluate our volumetric rendering on five challenging hand motion sequences. Finally, our experimental results show that our volumetric rendering improves hand anatomy visualization, compared to standard surface and volume rendering techniques.

Chapter 1

Introduction

The human hand is a vitally important part of the human body. Hands are capable of a wide range of precise motions due to their delicate and complex structures. Each hand consists of 27 bones, 34 muscles, over 100 ligaments and tendons, and many blood vessels and nerves, all confined into a small volumetric region. Visualizing the hand anatomy can help us understand its underlying structure and functionality, and visualize how anatomy inside the human hand moves under realistic hand motions.

Unlike surface rendering, volume rendering of medical image scans, such as magnetic resonance imaging (MRI), and computerized tomography (CT) data, can visualize the underlying interior structures. MRI scanning is preferable over CT since it is non-invasive and involves no ionizing radiation, and because it has a better contrast on soft tissues. However, volume rendering is challenging due to the following facts: (1) volume rendering is computationally more expensive than surface rendering; (2) the design of transfer functions, a crucial component of volume rendering, is a manual, tedious, and time-consuming process; (3) special attention needs to be paid to avoid rendering artifacts.

The MRI signal is a scalar-valued quantity available at points of a regular 3D grid with some small resolution (typically around 1mm). As such, the MRI signal alone has no color or opacity, it is a simple grayscale value, and there is no a prior way of mapping it to colors. By opacity we mean the standard transparency quantity in computer graphics, i.e., opacity=0 means a fully transparent object, and opacity=1 is a fully opaque object, and the in-between values indicate partial transparency. A transfer function maps the MRI intensity value to optical properties such as the (R, G, B) color and opacity. Essentially, a transfer function is a "colormap" that determines how the scalar-valued MRI values are converted into color and opacity. The transfer function is a critical ingredient of volume rendering, and its selection is highly non-trivial for biological tissues. A key contribution of this thesis is how to select the transfer function, potentially on a spatially-varying or per-tissue basis, to produce high-quality volume renderings of the human hand.

1.1 Objectives

This thesis aims to explore volumetric rendering of the human hand's internal anatomy, specifically to improve the visualization of hand MRI volume data. To do so, we define the following objectives:

- Implement the state-of-the-art volume rendering method, namely the volume ray casting [17] (referred to as "ray casting" in this thesis), to visualize MRI volume data directly. Ray casting evaluates the volume rendering integral by casting a ray into the volume, and integrating the optical properties along the ray at discrete sample positions.

- Propose a ray-casting-based rendering pipeline that takes not only the MRI volume data, but also the surface geometry of the internal structures as inputs, and produces high-quality volume renders that present interior structures with smooth boundaries.

- Study the human hand anatomy and design transfer functions that are suitable for hand tissues, including bones, muscles, tendons, joint ligaments and fat.

- Test our volumetric rendering method on multiple hand animations. Compare our results with renders produced by both standard surface-based rendering techniques and volume rendering done by related work.

1.2 Outline

The remainder of this thesis is organized as follows. Chapter 2 explains the theoretical background of volume rendering and human hand anatomy. Chapter 3 introduces the related work. Chapter 4 describes our volume rendering method and our choice of the transfer function. Chapter 5 presents the experimental results. Finally, Chapter 6 concludes this thesis with a summary and potential future work.

Chapter 2

Theoretical Background

2.1 Volume Rendering

The material in this section follows the ray casting algorithm and descriptions as given by [9] and [27]; we modify it as needed to accommodate our hand volumetric rendering.

2.1.1 Volume Data

Different from surface rendering, volume rendering visualizes information from a 3D volumetric data, which can be written as a mapping

$$\phi : \mathbb{R}^3 \to \mathbb{R},$$

from 3D space to a scalar value. The 3D scalar field, defined on a discretized grid, typically comes from measurements or simulations, and each volume element (called voxel) represents a unit of space, as illustrated in Figure 2.1.

2.1.2 Optical Models

Volume rendering treats volume as a distribution of light-emitting particles of certain densities. Thus, it can display the volume data by evaluating an optical model which describes how the volume emits,

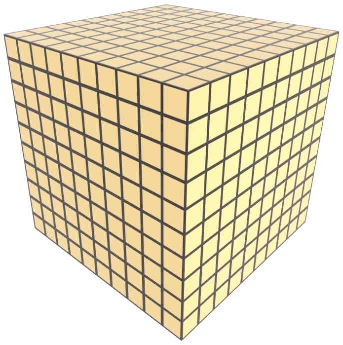
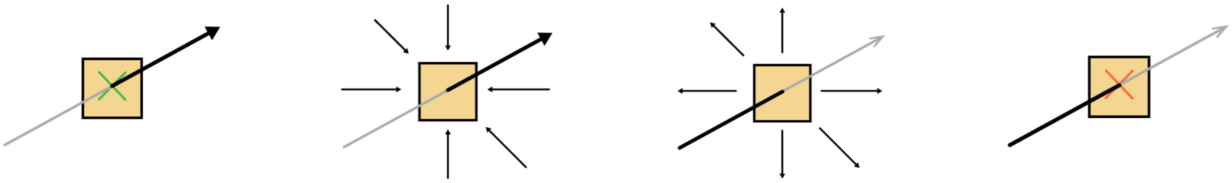


Figure 2.1: Volume data represented by a discrete uniform grid.



(a) emission

(b) in-scattering

(c) out-scattering

(d) absorption

Figure 2.2: Interactions between light and particles that affect the energy along the light ray.

scatters, and absorbs light. To achieve this, scalar value is mapped to physical quantities that describe the light interaction at the respective point in 3D space. This mapping performs the classification, and is usually considered to be a part of the transfer function. The physical quantities are then used to synthesize images. The following types of interaction are usually taken into account:

Emission. The particle actively emits light (Figure 2.2a), increasing the radiative energy.

Scattering. The particle scatters light, essentially changing the direction of light propagation, increasing or reducing radiative energy along a light ray. Through in-scattering (Figure 2.2b), additional energy is redirected into the direction of the ray; conversely, out-scattering (Figure 2.2c) removes energy from the current ray.

Absorption. The particle absorbs light (Figure 2.2d) and converts radiative energy into heat.

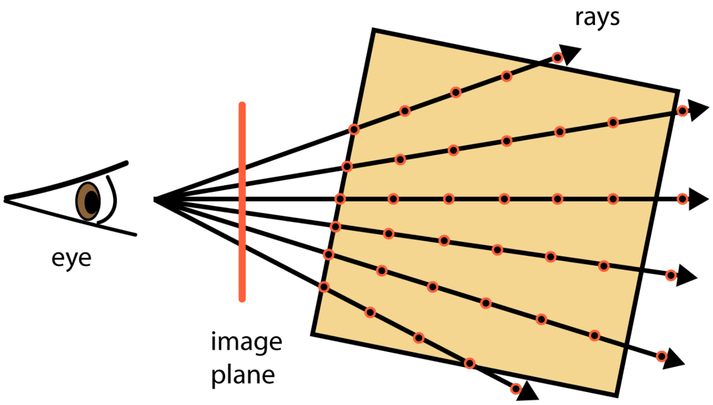


Figure 2.3: **Ray-casting principle.** For each pixel, one viewing ray is traced. The ray is sampled at discrete positions to evaluate the volume-rendering integral.

There are a few optical models [24] for volume rendering: emission only, absorption only, absorption plus emission, scattering and shading/shadowing, and multiple scattering. Among them, the emission plus absorption model is the most widely used, because it provides a good compromise between generality and efficiency of computation. In this model, the particle can emit light and absorb incident light while scattering and indirect illumination are neglected.

2.1.3 Volume Ray Casting

Volume ray casting [17] is the most basic, but also the most flexible volume rendering algorithm. The basic idea is to evaluate the volume-rendering integral (Section 2.1.4) along camera rays directly. As illustrated in Figure 2.3, for each pixel in the rendered image, a single ray is cast into the volume. At equispaced intervals along the ray, the discrete volume data is resampled, usually using tri-linear interpolation (more on Section 4.3) as the reconstruction filter. After resampling, the scalar data value is mapped to optical properties by means of a transfer function, yielding an RGBA quadruplet that includes the corresponding emission and absorption coefficients for this location. The volume rendering integral is then approximated via either front-to-back or back-to-front compositing (section 2.1.6), where usually the former is used in ray casting.

2.1.4 The Volume Rendering Integral

We denote a ray that has been cast into the volume by $\mathbf{x}(t)$, and parameterize it by the distance t from the camera. The scalar MRI value corresponding to a position along the ray is denoted by $s(\mathbf{x}(t))$. For the emission plus absorption model, absorption coefficients κ (accounting for absorption of light) and emissive color c (accounting for radiant energy actively emitted) can be simplified as function of the distance t instead of the scalar value s:

$$c(t) := c(s(\mathbf{x}(t))),$$

$$\kappa(t) := \kappa(s(\mathbf{x}(t))).$$

If this model are employed, the volume rendering integral integrates absorption coefficients $\kappa(t)$ and emissive colors $c(t)$ along a ray:

$$C = \int_0^\infty c(t)e^{-\tau(0,t)}\,\mathrm{d}t, \tag{2.1}$$

where the term

$$\tau(d_1, d_2) = \int_{d_1}^{d_2} \kappa(\,\hat{t}\,)\,\mathrm{d}\hat{t}, \tag{2.2}$$

is defined as the *optical depth* between distances d_1 and d_2. It measures how long the light travels before it is absorbed. Smaller values of optical depth mean that the participating medium is more transparent, and higher values mean that it is more opaque. The emissive color $c(t)$ has three components (R, G, B); each component is treated separately. Also note that $c(t)$ must already be pre-multiplied with opacity, i.e., it is the (R, G, B) returned by the transfer function, multiplied by the opacity given by the transfer function; this is called associated color [36].

2.1.5 Discretization

The main goal of volume rendering is to compute the volume-rendering integral (Equation 2.1). Typically, the integral cannot be evaluated analytically. Instead, numerical methods are applied to find an approximation as close to the exact value as possible.

The optical depth τ (Equation 2.2), which is the cumulative absorption up to a certain position $\mathbf{x}(t)$ along the ray, can be approximated by a Riemann sum

$$\tau(0,t) \approx \hat{\tau}(0,t) = \sum_{i=0}^{\lfloor T/\Delta t \rfloor} \kappa(i \cdot \Delta t)\Delta t, \tag{2.3}$$

with Δt denoting the distance between successive sampling locations. The summation in the exponent can immediately be substituted by a multiplication of exponentiation terms:

$$e^{-\hat{\tau}(0,t)} = \prod_{i=0}^{\lfloor T/\Delta t \rfloor} e^{-\kappa(i \cdot \Delta t)\Delta t}. \tag{2.4}$$

Now, we define the opacity as $\alpha_i = 1 - e^{-\kappa(i \cdot \Delta t)\Delta t}$, and rewrite Equation 2.4 as:

$$e^{-\hat{\tau}(0,t)} = \prod_{i=0}^{\lfloor T/\Delta t \rfloor} (1 - \alpha_i). \tag{2.5}$$

This allows the opacity α_i to be used as an approximation for the absorption of the i-th ray segment, instead of absorption at a single point.

Similarly, the emitted color of the i-th ray segment can be approximated by

$$C_i = c(i \cdot \Delta t)\Delta t. \tag{2.6}$$

Having approximated both the emissions and absorption along a ray, we now arrive at the approximate evaluation of the volume rendering integral as (denoting the number of samples by $n = \lfloor T/\Delta t \rfloor$)

$$\hat{C} = \sum_{i=0}^{n} C_i \prod_{j=0}^{i-1} (1 - \alpha_j), \tag{2.7}$$

which can be evaluated iteratively by performing either front-to-back or back-to-front compositing. We note that Equation 2.7 is evaluated separately for each color channel R, G, B.

2.1.6 Compositing Schemes

Equation 2.7 can be evaluated iteratively in front-to-back order by stepping i from 1 to n:

$$\begin{aligned} C'_i &= C'_{i-1} + (1 - \alpha'_{i-1}) C_i, \\ \alpha'_i &= \alpha'_{i-1} + (1 - \alpha'_{i-1}) \alpha_i. \end{aligned} \tag{2.8}$$

New values C'_i and α'_i are calculated from the color C_i and opacity α_i at the current location i, and from the composited color C'_{i-1} and opacity α'_{i-1} at the previous location $i-1$. The starting condition is $C'_0 = 0$ and $\alpha'_0 = 0$.

The alternative iterative formulation evaluates Equation 2.7 in back-to-front order by stepping i from $n - 1$ to 0:

$$C'_i = C_i + (1 - \alpha_i) C'_{i+1}, \tag{2.9}$$

A new value C'_i is calculated from the color C_i and opacity α_i at the current location i, and from the composite color C'_{i+1} at the previous location $i + 1$. The starting condition is $C'_n = 0$. Unlike front-to-back compositing, back-to-front compositing does not require tracking the alpha values.

2.2 Human Hand Anatomy

The structure of the human hand is extremely complex. It is composed of bones, joints, ligaments, muscles, tendons, blood vessels, nerves and skin. Bones are linked with joints. The hand has ligaments that hold the bones and cartilage together and provide flexibility. Muscles and tendons connect the bones, and through activation, create contractile forces and torques that bend the joints. The descriptions of the anatomy in this section were obtained using public medical websites ([25], [2], [7]), and were then condensed and paraphrased for consistency and clarity.

2.2.1 Skeletal System

Bones provide shape and support for the hand; they also serve as a storage site for minerals and provide the medium (marrow) for the development and storage of blood cells. They are made up of compact bone, spongy bone, and bone marrow, as illustrated in Figure 2.4. Compact bone makes up the outer layer of the bone. Spongy bone is found mostly at the ends of bones and contains red marrow. Bone marrow is located in the center of most bones and has many blood vessels. There are two types of bone marrow: red and yellow. Red marrow contains blood stem cells that can become red blood cells, white blood cells, or platelets. Yellow marrow is made mainly of fat.

There are three major types of bones in the hand (Figure 2.5), including:

Phalanges. The 14 bones that are found in the fingers of each hand. Each finger has 3 phalanges (the distal, intermediate or middle, and proximal); the thumb only has 2 (the distal and proximal).

Metacarpal bones. These 5 bones compose the middle part of the hand.

Carpal bones. These are the 8 bones located in the wrist; they are arranged in two rows. There are 4 bones in the distal row (hamate, capitate, trapezoid, and trapezium); and 4 in the proximal row (pisiform, lunate, triquetrum, and scaphoid).

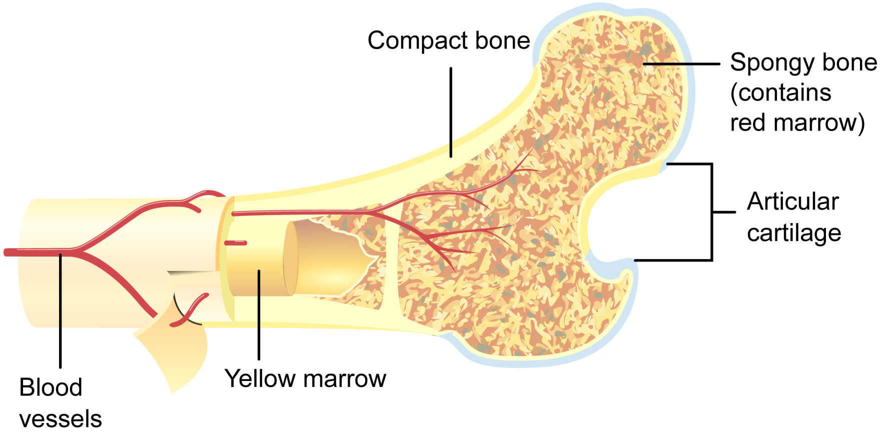


Figure 2.4: **Bones consist of compact bone, spongy bone, and bone marrow.** Compact bone makes up the outer layer. Spongy bone is found at the ends of the bone. Bone marrow is located in the center of the bone. Image downloaded and adapted from https://commons.wikimedia.org/wiki/File:Bone_Marrow_in_a_cross-section-FR.svg by Pbroks13 under licence CC-BY-3.0.

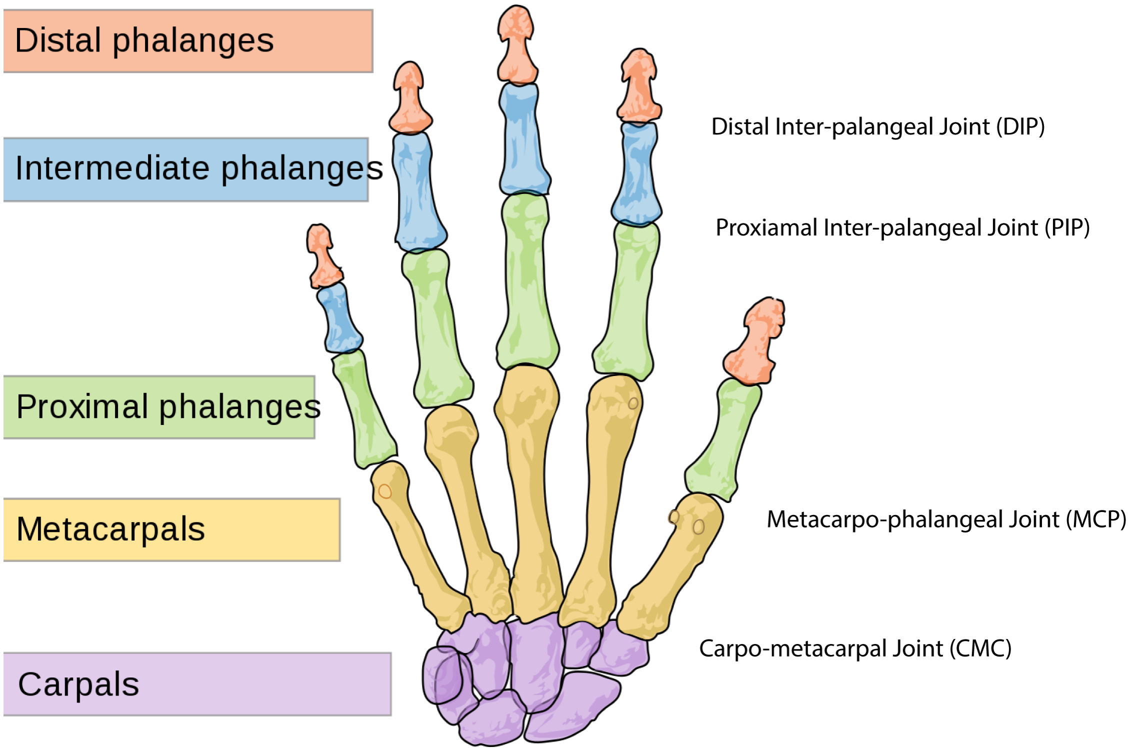

Distal phalanges
Intermediate phalanges
Proximal phalanges
Metacarpals
Carpals
Distal Inter-palangeal Joint (DIP)
Proxiamal Inter-palangeal Joint (PIP)
Metacarpo-phalangeal Joint (MCP)
Carpo-metacarpal Joint (CMC)

Figure 2.5: **The bones and joints of the human hand.** Source: Wikimedia Commons.

The joints in our hands are made up of cartilage surfaces that cap the bones, which allows the bones to glide smoothly against one another as they articulate. There are four types of joints in our hand (Figure 2.5):

Distal inter-phalangeal (DIP) joints. The 5 joints closest to the fingertips.

Proximal inter-phalangeal (PIP) joints. The 4 joints in the middle of fingers.

Metacarpo-phalangeal (MCP) joints. The 5 joints at the base of fingers, referred to as the "knuckles."

Carpal-metacarpal (CMC) joints. The 5 joints located at the bottom of the hand.

The ligaments are tough bands of connective tissue that connect the bones to support them and keep them in place. Important ligaments of the hand are:

Collateral ligaments. Strong ligaments on either side of the finger and thumb joints, which prevent sideways movement of the joint.

Volar plate. A ligament that connects the proximal phalanx to the middle phalanx on the palm side of the joint. As the joint in the finger is straightened, this ligament tightens to keep the PIP joint from bending backward.

Volar radiocarpal ligaments. A complex web of ligaments that support the palm side of the wrist.

Dorsal radiocarpal ligaments Ligaments that support the back of the wrist.

2.2.2 Muscular System

The skeletal muscles are responsible for the movement of the hand and fingers. The dexterity of the hand comes from the unique configuration of bones, joints, and muscles: *flexion*, bending in the anterior direction (e.g., forming a fist); *extension*, straightening or bending in the posterior direction; *abduction*, movement away from the center the the hand (i.e., spreading the fingers); and *adduction*, movement toward the center of the body (i.e., bring the fingers together). The hand muscles consist of five groups (Figure 2.6):

Thenar muscles. A collection of three muscles at the fleshy base of the thumb on the palmar side that act to exert movement of the thumb. The thenar muscles include abductor pollicis brevis (APB), flexor pollicis brevis (FPB) and opponens pollicis (OP).

Hypothenar muscles. A collection of three muscles at the fleshy base of the pinky finger on the palmar side that act to exert movement of the pinky finger. The hypothenar muscles include abductor digiti minimi (ADM), flexor digiti minimi brevis (FDMB) and opponens digiti minimi (ODM).

Lumbricals. The lumbrical muscles are composed of four muscles that act to flex their respective MCP joint and extend the PIP and DIP joints.

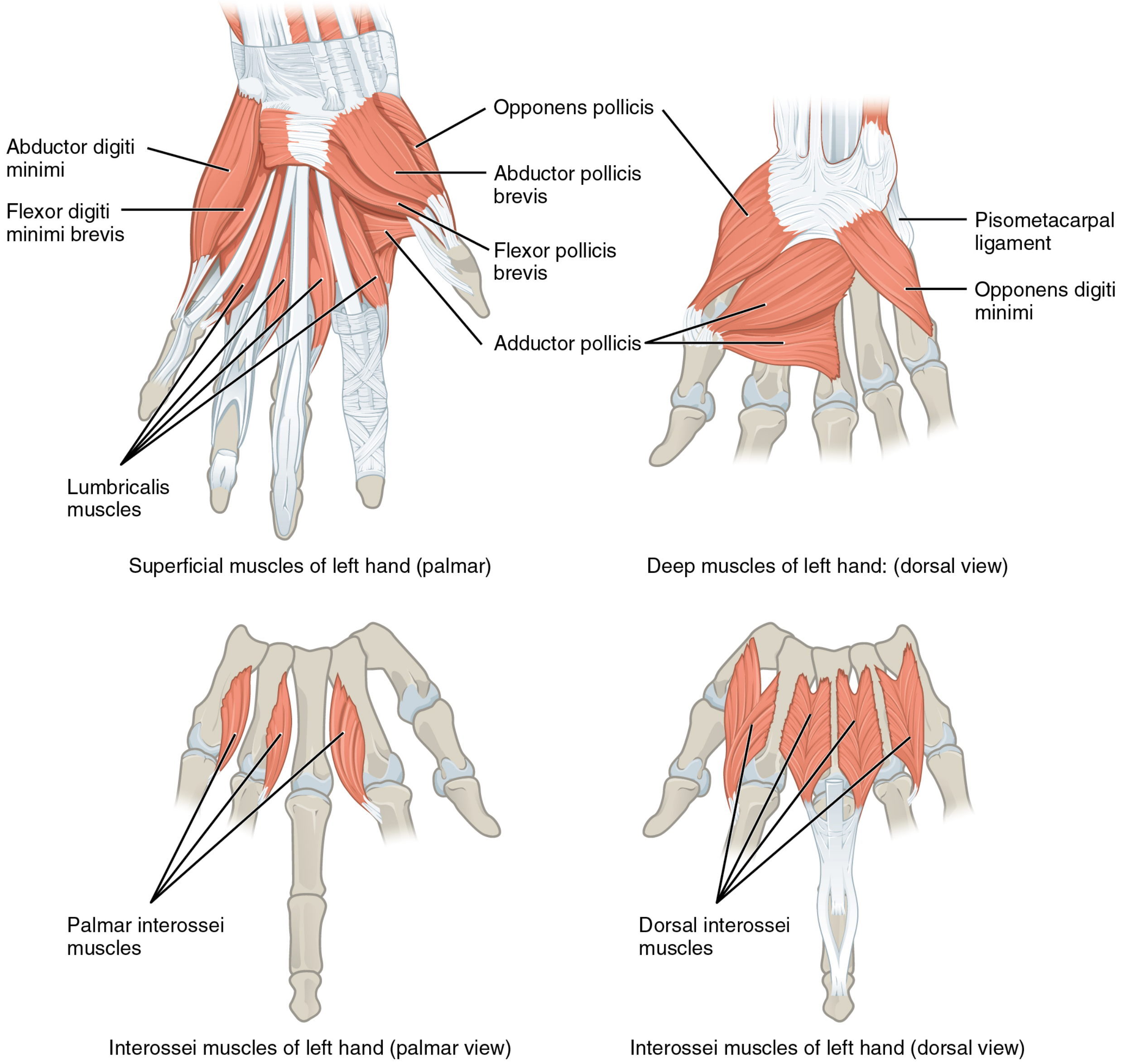
Interossei muscles. They are comprised of three palmar interossei (PI) muscles and four dorsal (DI) muscles.

Adductor pollicis. Adductor pollicis muscle adducts the thumb. It has two heads: transverse and oblique.

Tendons are bands of connective tissue that attach the muscles to the bone, enabling the muscles to move the bones. The main tendons of the hand (Figure 2.7) are:

Superficialis tendons. They pass through the palm side of the wrist and hand, and attach at the bases of the middle phalanges. They act with the profundus tendons to flex the wrist, MCP, and PIP joints.

Profundus tendons. Similar to superficialis tendons, except they attach at the bases of the distal phalanges. They act with the superficialis tendons to flex the wrist and MCP and PIP joints. They also flex the DIP joints.



Superficial muscles of left hand (palmar)

Deep muscles of left hand: (dorsal view)

Interossei muscles of left hand (palmar view)

Interossei muscles of left hand (dorsal view)

Figure 2.6: Muscles of the human hand. Image downloaded from https://commons.wikimedia.org/wiki/File:1121_Intrinsic_Muscles_of_the_Hand.jpg by OpenStax under license CC-BY-4.0.

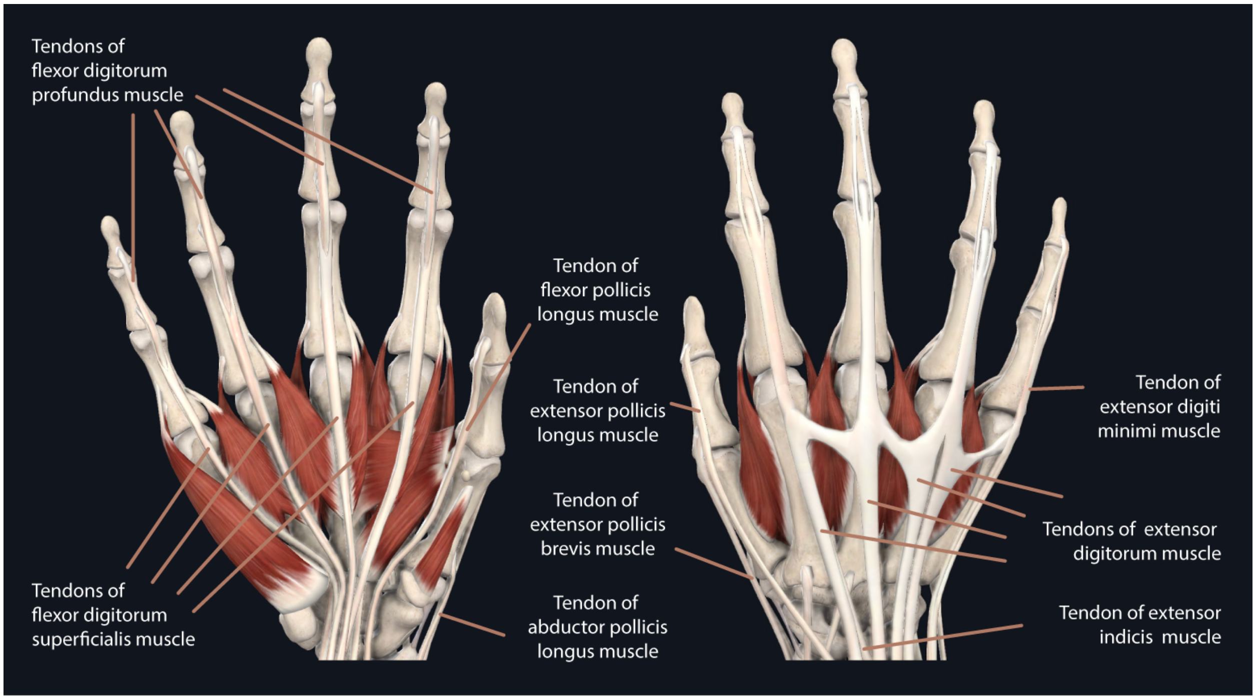

Tendons of flexor digitorum profundus muscle
Tendon of flexor pollicis longus muscle
Tendon of extensor pollicis longus muscle
Tendon of extensor pollicis brevis muscle
Tendon of abductor pollicis longus muscle
Tendons of flexor digitorum superficialis muscle
Tendon of extensor digiti minimi muscle
Tendons of extensor digitorum muscle
Tendon of extensor indicis muscle

Figure 2.7: **Tendons of the human hand.** © 2022 Elsevier. Image adapted, with permission, from https://3d4medical.com under student license.

Extensor tendons. They attach to the middle and distal phalanges and extend the wrist, MCP, PIP and DIP joints.

Flexor tendons. Nine long tendons which pass from the forearm through the carpal tunnel of the wrist. They diverge in the palm, where two go to each finger and one goes to the thumb.

Extensor pollicis brevis and abductor pollicis longus. They originate from the muscles in the top of the forearm and enable movement of the thumb.

2.2.3 Vascular and Nervous System

There are three main nerves which supply the hand and wrist, the median, radial, and ulnar nerves, as shown in Figure 2.8 in yellow. Nerves are like fiberoptic cables that carry signals. The brain sends signals

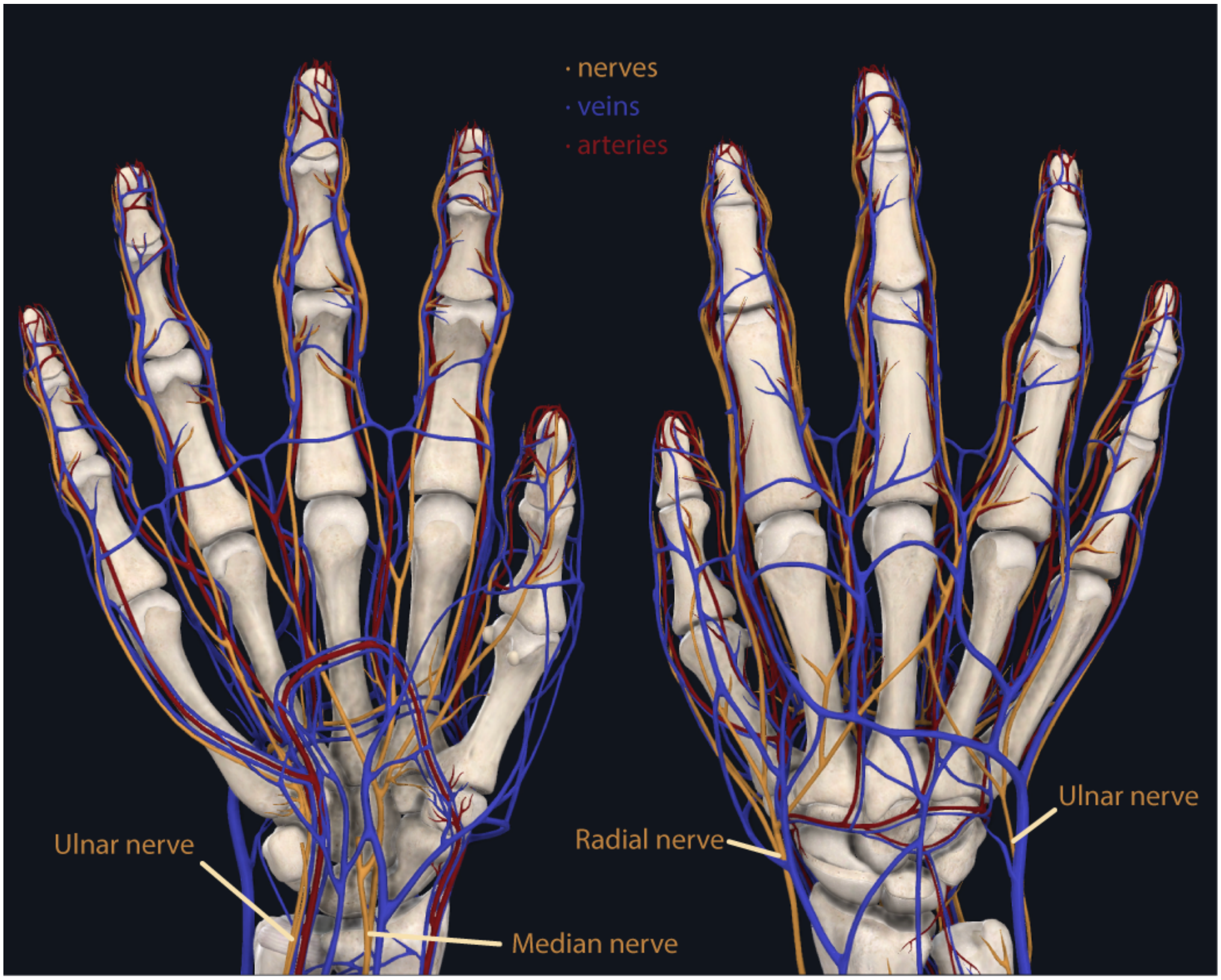


Figure 2.8: **Nerves (yellow), veins (blue) and arteries (red) of the hand.** © 2022 Elsevier. Image adapted, with permission, from https://3d4medical.com under student license.

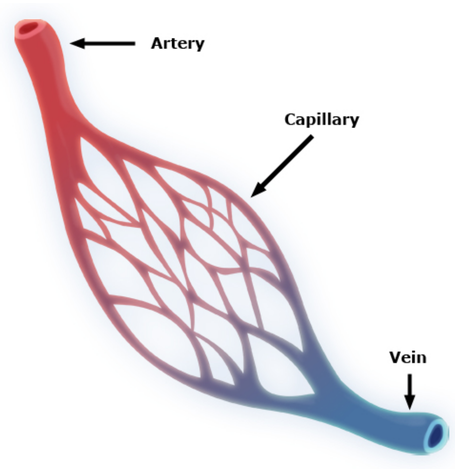


Figure 2.9: **Types of blood vessels.** Source: Wikimedia Commons.

down the spine and then through the nerves in the arm to control the muscles. Sensory signals like temperature, vibration, and pain travel from nerves in the hand up the arm, through the spine, and back to the brain.

Blood vessels are channels that carry blood throughout the body. They form a closed loop, like a circuit, that begins and ends at the heart. There are three types of blood vessels:

Arteries. Arteries (Figure 2.8 in red) are strong, muscular blood vessels that carry oxygen-rich blood from the heart to the body. They handle a large amount of force and pressure from the blood flow but do not carry a large volume of blood. At any given time, only about 10% to 15% of the body's blood is in the arteries.

Veins. Veins (Figure 2.8 in blue) do not have to carry highly pressurized blood, but they do have to carry large volumes of deoxygenated blood back to the heart. Thin, less elastic walls help them handle high volumes and low pressure. Most veins have valves that open and close. The valves control blood flow and keep the blood flowing in one direction. About 75% of the blood is in the veins.

Capillaries. Capillaries (Figure 2.9) are the smallest blood vessels that connect arteries and veins at the finger tip. Capillaries are where oxygen and nutrients are exchanged for carbon dioxide and waste.

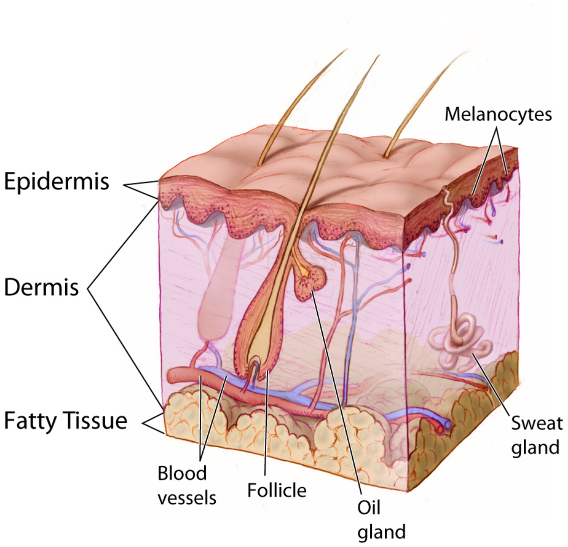


Figure 2.10: **Different layers of the skin.** Source: Wikimedia Commons.

2.2.4 Integumentary System

The skin functions as a barrier against the external environment. It prevents fluid loss, provides the sensation of touch, pressure, pain, and temperature. It also performs important functions like Vitamin D synthesis. The skin consists of three layers, including the superficial epidermis, the dermis in the middle, and the deep subcutaneous/fatty tissue, as illustrated in Figure 2.10.

Chapter 3

Related Work

3.1 Volume Rendering Algorithms

In general, volume rendering methods can be coarsely classified into two categories: indirect volume rendering (IVR), and direct volume rendering (DVR).

IVR or "surface rendering" fits geometric primitives from the volume data and then renders these primitives. Methods in this category include iso-surfacing [22] and frequency domain rendering [31].

DVR or simply "volume rendering" does not explicitly extract geometric structures from volume data, but rather renders the volumes based on fuzzy segmentation through the transfer function. Several volume rendering algorithms have been proposed over the years. Splatting [35] works by virtually "throwing" the voxels onto the image plane. In the process, every voxel in the object space projects/leaves a splat/footprint in the image plane that will represent the object. Shear-warp rendering [6][18] is based on a factorization of the viewing matrix into a 3D shear parallel to the slices of the volume data, a projection to form a distorted intermediate image, and a 2D warp to produce the final image. Texture slicing [5][32] takes advantage of GPU hardware to render the volume data represented by the 2D or 3D texture. In volume ray casting [20][17], which has been described in detail in Section 2.1.3, camera rays are cast into the volume, optical properties at sample positions are then integrated along these rays, and the results are the pixels that contribute to the final image. Of all volume rendering algorithms, volume ray casting has seen the largest body of publications and achieves images of highest quality [26].

3.2 Design of Transfer Functions

Suppose we have a volumetric dataset that contains scalar values that represent some spatially varying physical property. In general, there is no natural way to obtain emission and absorption coefficients from such data. Instead, the user must decide how the different structures in the data should look by assigning optical properties to the data values using an arbitrary (artificial) mapping. Such a mapping is called a transfer function.

Although a few approaches exist to automatically generate transfer functions by some image- or data-driven mechanisms [12], the process of finding an appropriate transfer function is in general a manual, tedious, and time-consuming procedure. It also requires detailed knowledge of the spatial structures that are represented by the dataset.

Arens and Domik (2010) [1] published a survey of transfer functions for volume rendering in which they subdivided transfer functions into the following six categories: 1D data-based, gradient 2D [20][15], curvature-based [13], size-based [8], texture-based [4], and distance-based [30]. Later, a state-of-the-art (STAR) report of transfer functions for volume rendering was published by Ljung et al. (2016) [21]. This STAR report classified transfer function research based on the following aspects: dimensionality, derived attributes, aggregated attributes, rendering aspects, automation, and user interfaces (Figure 3.1).

A transfer function is essentially a special, but important case of a *classification*. With such classification, certain region in the three-dimensional domain are identified to be the same material, such as bone, vessel, or soft tissue. However, a challenge presents itself when the values of these materials are not standardized and when they overlap with each other, which makes the material assignment ambiguous. This problem is particularly pertinent when dealing with MRI data.

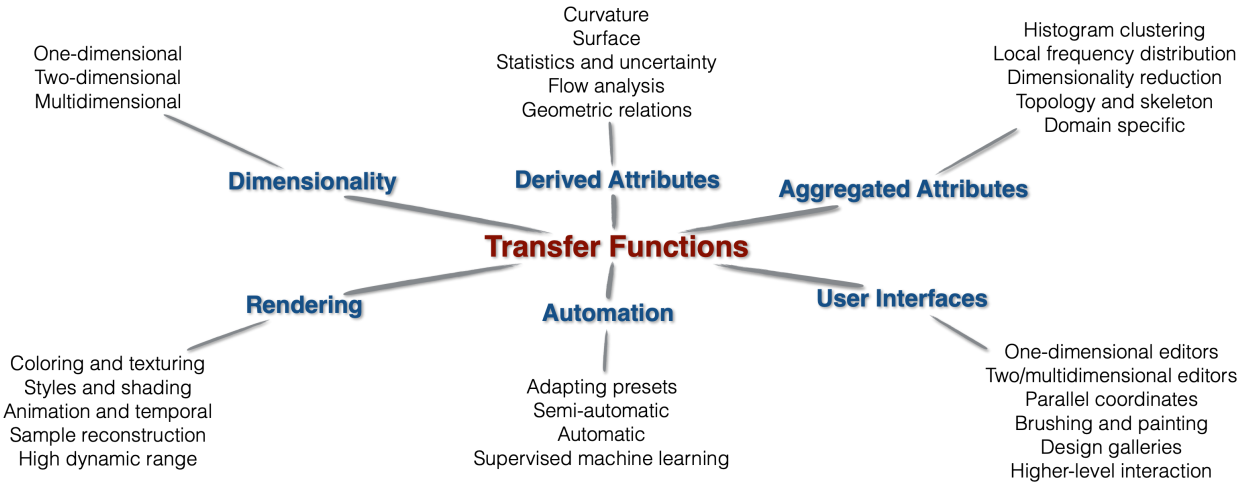


Figure 3.1: The classification of transfer functions. © 2022 John Wiley and Sons. Reproduced, with permission, from Ljung et al. [21].

Segmentation is the process of identifying or tagging individual voxels as belonging to one of several materials contained in a volume. Rendering with segmented volume data [28][11] can be seen as an extension of standard volume rendering, owing to the fact that different segmented regions can be rendered using different optical properties, via spatially varying classification and/or transfer functions.

3.3 Volume Rendering of the Human Hand

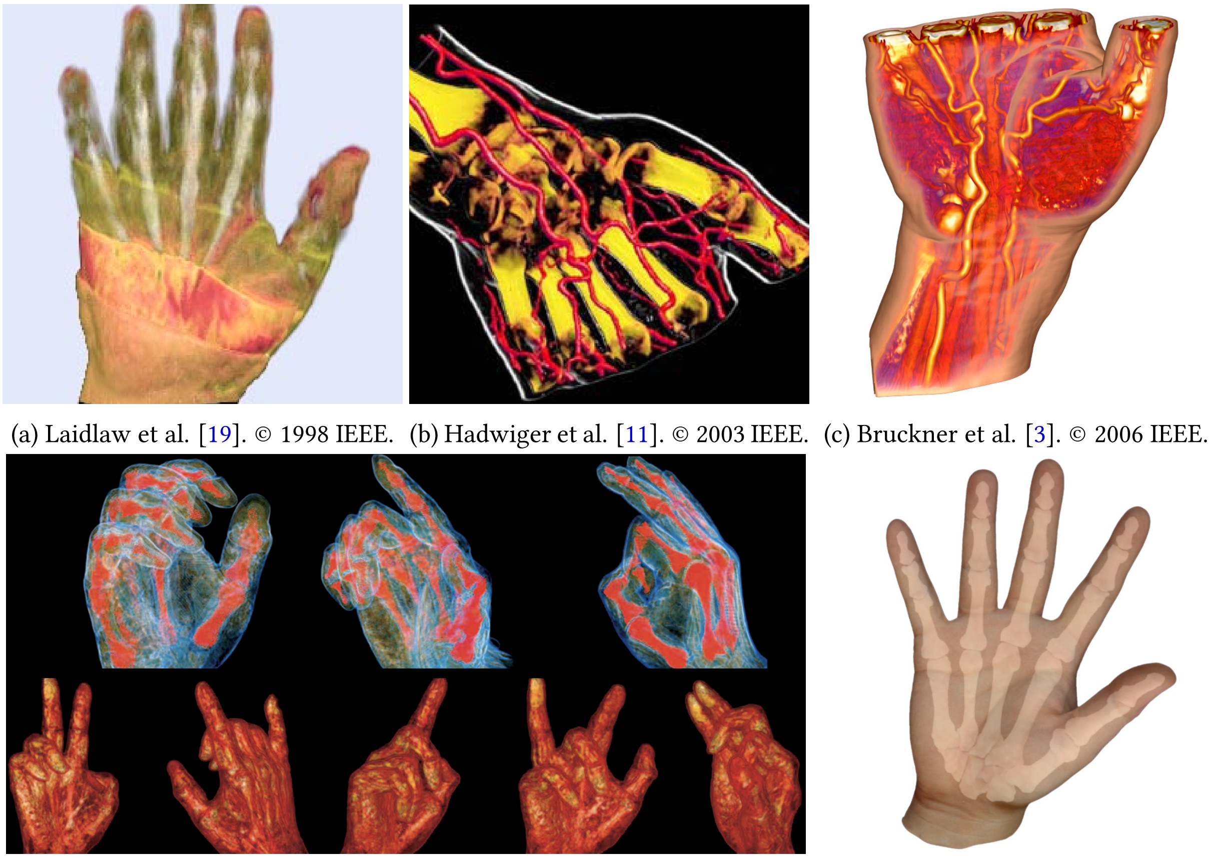
The goal of volume rendering of medical scans (CT or MRI) is to provide insight into human anatomy, such as the geometric shape and function of bones, muscles, tendons, skins, etc. This is a difficult problem because, when visualizing a three-dimensional dataset, it is in general not possible to concurrently show all the data. Instead, certain selective representations of the data must be used for visualization.

Laidlaw et al. (1998) [19] proposed an algorithm for identifying the distribution of different material types in the volume data without explicit segmentation. They allowed for mixture of materials inside a voxel; therefore stair-case artifacts caused by partial volume effects are reduced. These artifacts are usually introduced by segmentation algorithms and appear along the boundaries between materials. Different optical properties then can be assigned to different materials for volume rendering. Figure 3.2a demonstrates their result on a MRI hand data. The opacity of different materials is decreased above cutting planes to show details of the classification process within the hand.

The combination of different rendering styles is of particular interest, as it allows one to put emphasis on features of interest. Hadwiger et al. (2003) [11] presented a two-level volume rendering approach using explicit segmentation information, where different objects can have different transfer functions, different rendering modes that are not only limited to DVR (e.g., iso-surfacing, and non-photorealistic rendering), and different compositing modes (e.g., alpha blending, maximum intensity projection). The latter capability enables two-level volume rendering, which consists of one local compositing mode per object, and a second global compositing level that combines the contributions of different objects. Figure 3.2b shows their result on a CT segmented hand dataset with different rendering modes. In particular, the skin is rendered with non-photorealistic contour rendering and MIP compositing, the bones are rendered with DVR, and vessels are rendered with tone shading.

Bruckner et al. (2006) [3] proposed a context-preserving volume rendering model as a function of shading intensity, gradient magnitude, distance to the eye point, and previously accumulated opacity to selectively reduce the opacity in less important data regions. They took inspiration from illustration, where the *ghosting* is commonly used by artists when they want to visualize specific internal structures as well as the exterior. The name "context-preserving" is related to the concept of focus-plus-context, which is well-known in information visualization. Especially interesting subsets of the data, are considered to be "in focus," whereas the rest of the data merely provides context. Figure 3.2c shows their illustrative visualization of a CT hand dataset.

Rhee et al. (2010) [28] present the first anatomically accurate 3D volume rendering of the hand MRI data in motion, including bone animation and soft tissue deformation driven by a joint skeleton. The bones are segmented from the hand MRIs. In their work, two styles of renders were used. One uses a



(a) Laidlaw et al. [19]. © 1998 IEEE. (b) Hadwiger et al. [11]. © 2003 IEEE. (c) Bruckner et al. [3]. © 2006 IEEE.

(d) Rhee et al. [28]. © 2010 IEEE.

(e) Wang et al. 2019 [33].

Figure 3.2: Examples of volume rendering from related work. Reprinted with permission.

constant red color for bones, plus semi-transparent skin (Figure 3.2d top); and the other is employs the volume-rendering program developed by Kniss et al. [14] (Figure 3.2d bottom).

Wang et al. (2019) [33] proposed a system to model and simulate the hand using MRI. They first segmented hand bone anatomy (meshes) in multiple poses from MRI, interpolated and extrapolated them to the entire range of motion, essentially producing an accurate data-driven animation rig for bones. Then, they simulated soft tissues using finite element methods (FEM) driven by the bone animation. They visualized their results by compositing the surface renders of the bones and skin together with transparency, as illustrated in Figure 3.2e. They also published their dataset [34], which is the world's first dataset of MRI scans of the human hand in multiple poses.

Chapter 4

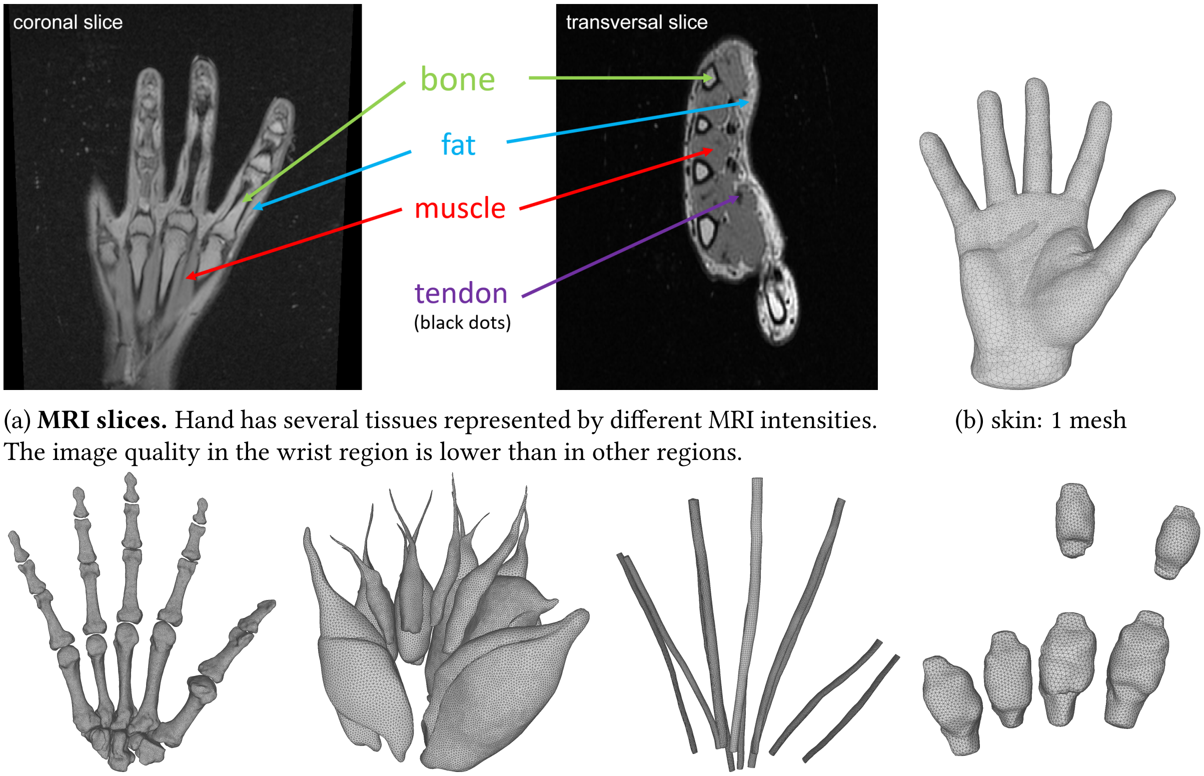
Volume Rendering Pipeline for Hand Anatomy

In this section, we give a structural overview of our anatomy volume-rendering pipeline based on the volume ray-casting algorithm. We propose an extension to this algorithm: given the geometry of the internal structures, the camera ray can be decomposed into several segments, depending on what objects are intersected by the ray; individual transfer functions can then be used for different segments, depending on what objects are traversed. The evaluation of the optical model for volume rendering can be separated into the following stages: data acquisition, volume traversal, interpolation, classification, and compositing.

4.1 Data Acquisition

Our hand MRI data in the neutral pose (Figure 4.1a) was acquired from the existing dataset created by Wang et al. (2019) [34]. They used lifecasting materials to generate hand molds to prevent hand movement during the scanning. The noise outside of the hand regions in their MRI images is actually the air bubbles of the mold. The neutral pose corresponds to "pose01" in the dataset. In addition, we use their geometry meshes corresponding to the internal structures. These meshes were segmented from the MRI using the method described in [33].

The geometry corresponds to the following structures of hand anatomy: skin, bones, muscles, tendons, and ligaments. All hand bones, including phalanges, metacarpals, and carpals, except the proximal row of



(a) **MRI slices.** Hand has several tissues represented by different MRI intensities. The image quality in the wrist region is lower than in other regions.

(b) skin: 1 mesh

(c) bones: 23 meshes

(d) muscles: 17 meshes

(e) tendons: 10 meshes

(f) ligaments: 6 meshes

Figure 4.1: MRI slices (a) and corresponding mesh geometry (b-f) in the neutral pose.

carpals were segmented, this is due to the low MRI quality at the wrist region (Figure 4.1a, Figure 4.1c). All muscles of the hand were segmented (Figure 4.1d). All tendons (except extensor pollicis brevis and abductor pollicis longus located at the medial side of the thumb) were paired and simplified into rod shapes (Figure 4.1e). Similarly, the ligaments (Figure 4.1f) at 4 MP joints of the index, middle, ring and pinky finger, and 2 PIP joints of the index and middle finger, were segmented and combined into six "capsules" that wrap around these joints. The numbers of segmented meshes are: 1 for the skin, 23 for bones, 17 for muscles, 10 for tendons, and 6 for ligaments. The extracted geometry closely matches the MRI data: Figure 4.1 shows the MRI slices and corresponding mesh geometry in the neutral pose.

4.2 Volume Traversal

Following the volume ray casting algorithm, for each pixel of the final image, a viewing ray is cast into the volume. Ray-object intersections are performed to find a list of all intersections between this ray and meshes in the scene. These intersections are then sorted by their distances to the camera. After the sorting is done, the positions of the first and last intersection define the starting and ending point of the sampling, respectively. If there is no intersections along the ray, no further computation is need and the value of the corresponding pixel will be the background color. Since empty space outside the hand does not contribute to the final image, sampling points between outermost intersections of viewing ray and mesh groups is a reasonable decision, and by doing so: (1) the workload for each pixel is reduced, and hence increasing the rendering speed; (2) any MRI noise outside the hand caused by the air bubbles in the hand mold (Figure 4.1a) are ignored, improving the image quality.

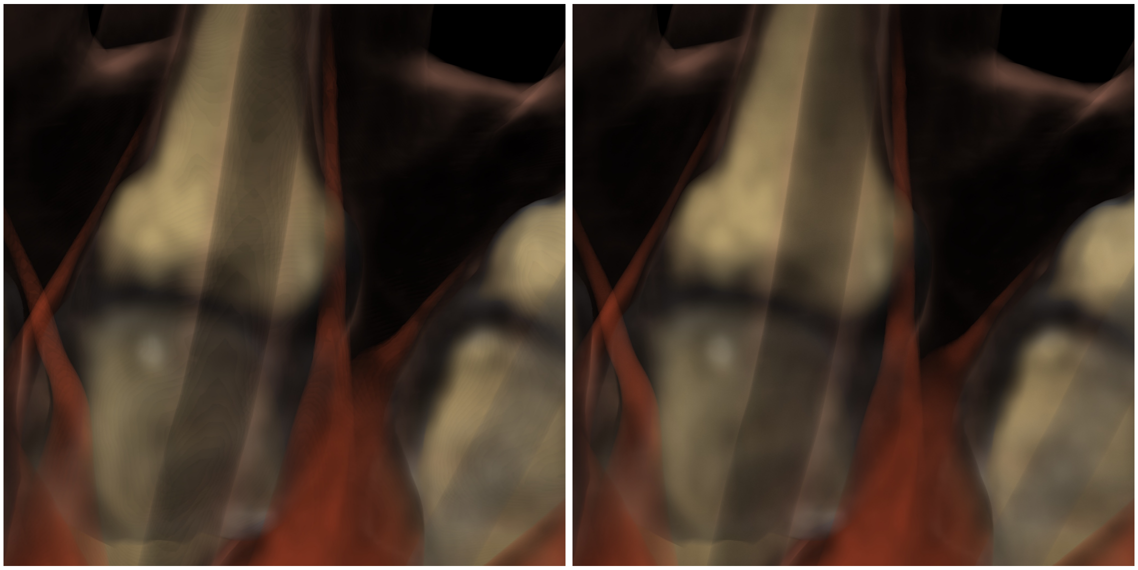


Figure 4.2: **Volume rendering without (left) and with (right) stochastic jittering of the sampling positions.** Wood-grain artifacts can be observed in the left image.

Along the path of the ray, equidistant sampling points can be selected. However, the approximate discretization of Equation 2.1 into Equation 2.7 may lead to sampling artifacts when the sampling rate is

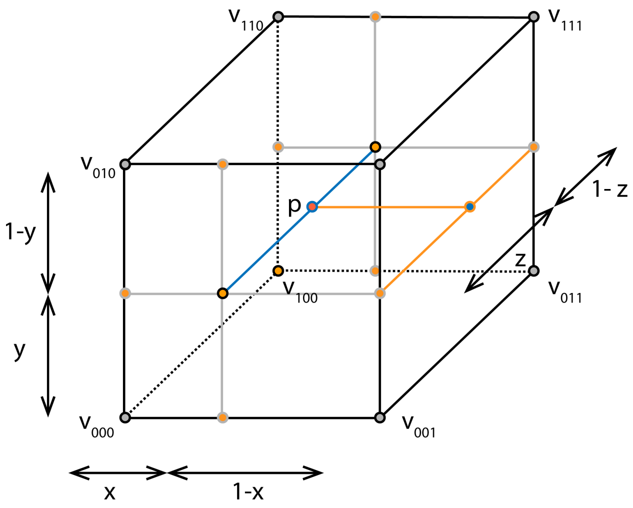


Figure 4.3: Tri-linear interpolation.

too low. This can manifest as "wood-grain artifacts" (Figure 4.2 left). This issue can be alleviated with two solutions: (1) decrease the distance between sample points, which will also decrease the rendering speed; (2) add random small offsets to the sampling positions of rays in the viewing direction using stochastic jittering. We adopted the second solution in our pipeline. Figure 4.2 shows a comparison of two renders without and with stochastic jittering.

4.3 Interpolation

Since the sampling positions are usually different from grid points, a continuous 3D field needs to be reconstructed from the discrete grid in order to obtain the data values at the sample positions.

We adopted tri-linear interpolation to interpolate grid MRI values to the entire volume (Figure 4.3). This is the most common reconstruction method for uniform grids, as it is fast to compute. To simplify the notation, the following discussion assumes normalized coordinate values x, y, and z that are in the interval

$[0, 1]$ for points within the voxel. Normalized coordinates can be obtained from arbitrary coordinates by scaling and translation. Tri-linear interpolation between eight points $\mathbf{v_{000}}$ to $\mathbf{v_{111}}$ is given by:

$$\begin{aligned} f(\mathbf{p}) = {} & (1-z)(1-y)(1-x)f(\mathbf{v_{000}}) \\ & + (1-z)(1-y)xf(\mathbf{v_{001}}) \\ & + (1-z)y(1-x)f(\mathbf{v_{010}}) \\ & + (1-z)yxf(\mathbf{v_{011}}) \\ & + z(1-y)(1-x)f(\mathbf{v_{100}}) \\ & + z(1-y)xf(\mathbf{v_{101}}) \\ & + zy(1-x)f(\mathbf{v_{110}}) \\ & + zyxf(\mathbf{v_{111}}), \end{aligned}$$

where $f(\mathbf{v_{000}})$ to $f(\mathbf{v_{111}})$ are the function values at the sample points $\mathbf{v_{000}}$ to $\mathbf{v_{111}}$, respectively. The result is the interpolated function value at point $\mathbf{p}$.

4.4 Classification

Classification maps scalar values of the dataset to optical properties for the volume-rendering integral (Equation 2.1). It allows us to distinguish different materials in a volume. In our method, a material or tissue type is assigned to each sample position. Each material stores a transfer function that maps from scalar value $s(\mathbf{x}(t))$ to color $C(t)$ and opacity $\alpha(t)$.

4.4.1 Material Assignment

We first classify each sample position based on its spatial relation to the following mesh groups: bones, muscles, tendons, ligaments, and skin. That is, if a sample point is inside a mesh group, it belongs to the category represented by the mesh group. The category of materials includes bone, muscle, tendon, ligament, and fat (inside the skin mesh). Since meshes overlap with each other, e.g., every internal mesh is inside the skin mesh, joint ligaments intersect with the end of the phalanges, etc., we assign the following priority to these categories:

$$bone > tendon > muscle > ligament > fat. \tag{4.1}$$

Specifically, our method works as follow: (1) for each pixel, we compute a sorted list of intersections between the viewing ray and all meshes in the scene; (2) create equidistant sampling points between the closest and furthest intersections, with stochastic jittering, in a front-to-back manner; (3) maintain a priority queue of (geometry id, material id) pairs that can be directly used for material assignment; the priority depends on the material id (Equation 4.1); (4) if we encounter a new mesh that is not in the queue, this implies that it is an entry point of an interval whereby all interior sampling points share the same material; a pair (geometry id, material id) for this mesh is inserted into the queue; (5) if we encounter an old mesh that is already in the queue, this implies an exit point of a material interval; the corresponding pair in the queue should be removed; (6) during volume traversal, the material with highest priority in the queue is assigned to the sample point. In Figure 4.1, we demonstrate an example of the material assignment.

It is evident that without priority, the material assignment of a sample point can be ambiguous because this sample point can belong to more than one category. With priority assignment, a sample point will be classified as the one that has the highest priority among all the categories that it could belong to. Figure 4.5 shows a comparison of the renders of bones, tendons, muscles, and ligaments without and with priority assignment using constant color and opacity as transfer functions (i.e., without MRI data).

Traditionally, segmentation is performed prior to the volume traversal. The segmentation information can be represented in an object ID volume, where each voxel contains the ID of the object it belongs to. Then the segmented volume data can be rendered using basic ray casting, with the knowledge of the object

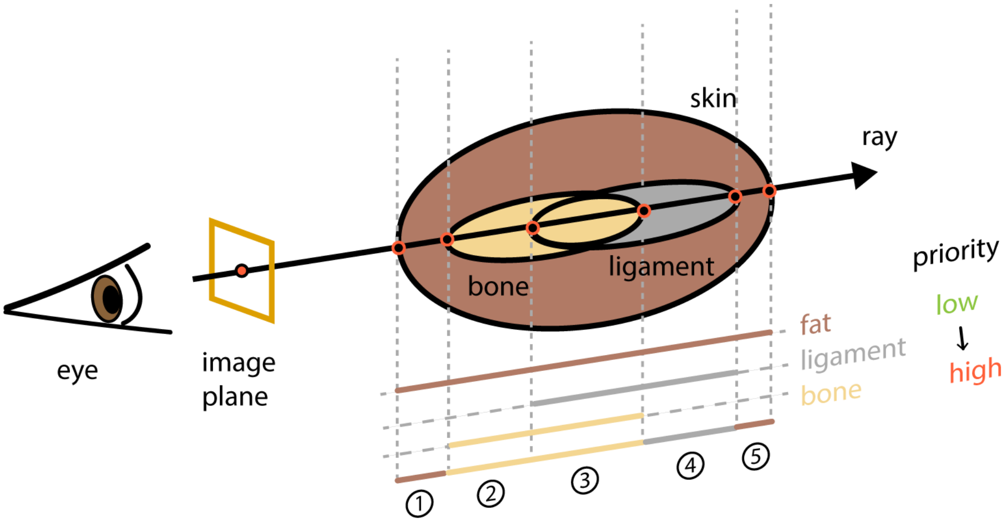


Figure 4.4: **An example of material assignment.** A single ray corresponding to a given image pixel is intersected with all meshes in the scene (skin, bone and ligament in this example). Sampling points that lie inside the interval between two neighboring intersections are identified as the same material. In the first and fifth intervals, only skin is present in the queue, and so every sampling point in these two segments is classified as fat tissue. In the second interval, both bone and skin are present in the queue. However, the bone has higher priority than the skin, so sample points in this region are classified as bone tissue. Similarly, sample points in the fourth interval are classified as ligament tissue. In the third interval, all tissues are present, hence the bone material is assigned because the bone has the highest priority.

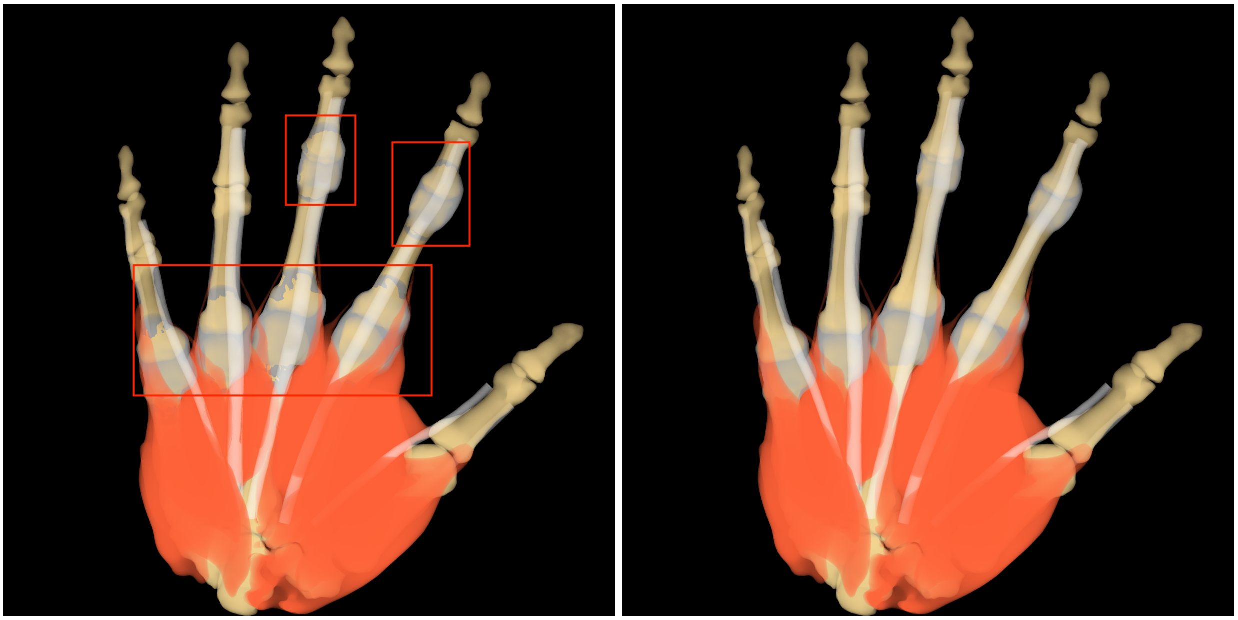


Figure 4.5: **Comparison of renders without (left) and with (right) priority assignment.** Artifacts marked with red squares are caused by wrong material assignments between bones and joint ligaments.

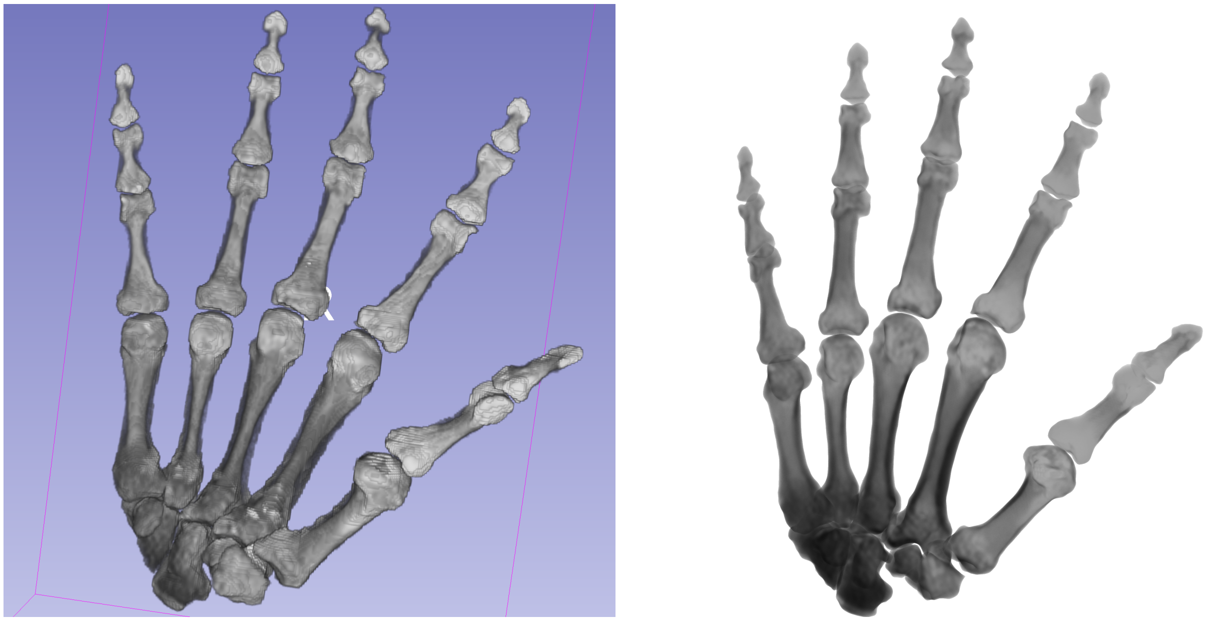


Figure 4.6: **Comparison of segmented bone renders between “3D slicer” (left) and our method (right).** Our result is free from staircase/voxelization artifacts that can be observed on the left.

ID of each voxel. However, staircase/voxelization artifacts caused by partial volume effects will appear at the boundaries of two objects. This approach of performing ray-object intersections and sampling within the segments between intersections produces higher image quality than the above-mentioned voxel approach. With our method, the boundary between two materials is the smooth surface of a mesh, instead of the irregular outer surface of a discretized grid. Therefore, our results are free of staircase artifacts. In Figure 4.6, we give the comparison of volume rendering of a segmented bone between “3D slicer” (a free and open-source software capable of segmentation and volume ray casting), and our renderer.

4.4.2 Transfer Functions

With material assignment, we can selectively use different transfer functions for different objects within the dataset. Previously, Hadwiger et al. (2003) [11] and Bruckner et al. (2006) [3] utilized focus-plus-context (F+C) to better visualize the 3D information: depending on whether objects are selected to be “in focus” or not, their visual appearance can be different. Whereas objects “in focus” may feature significant opacity, objects which are considered to be “context” act as a semitransparent reference. Inspired by this

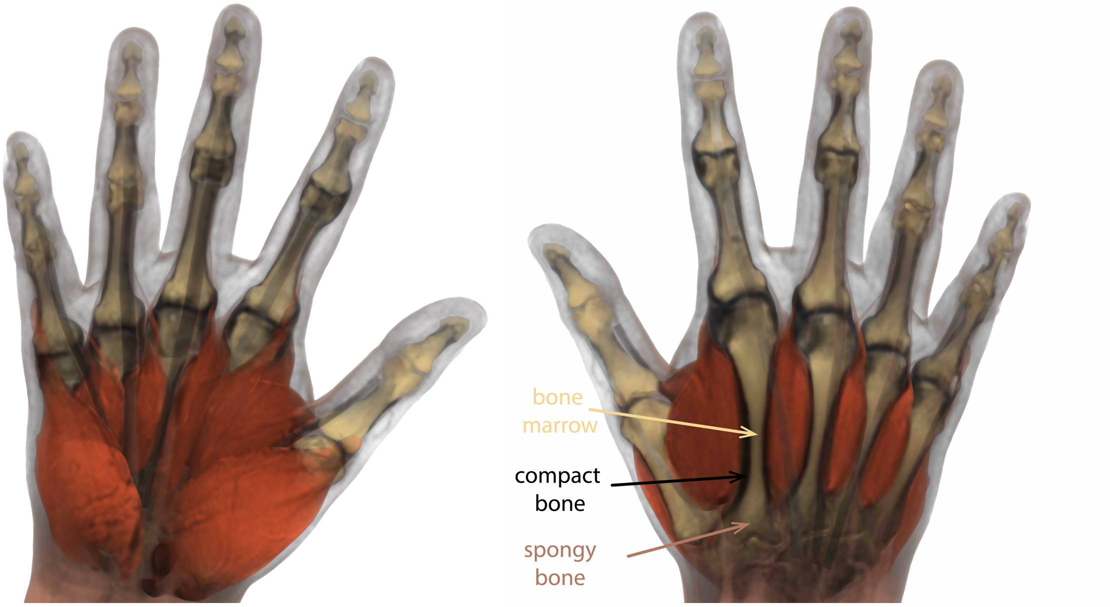


Figure 4.7: **Volume rendering with interior-emphasized style.** Internal organs are clearly shown (on both sides), boundaries between tissues are smooth and clear, and semi-transparent skin provides the context of the hand shape. In particular, on the palmar side (left), rich muscle textures can be observed; on the dorsal side (right), bone marrow, compact bone, and spongy bone of metacarpals can be recognized.

concept, we devised two styles/groups of transfer functions to individually emphasize the inner structures (bones, muscles, tendons, and ligaments) and outer subcutaneous fat tissue that lies beneath the skin. We named these two styles "interior-emphasized" and "fat-emphasized." A style is given by specifying a transfer function for each material. Hence, for each style, once a sample point's material is determined, the associated transfer function is also determined.

4.4.2.1 Interior-Emphasized Style

In the interior-emphasized style, the skin looks like a semi-transparent membrane, and internal organs (such as bones and muscles) under the skin are clearly shown with significant opacity and contrasting colors (Figure 4.7).

The interior-emphasized style is achieved as follows. (1) Divide objects into two major groups: non-fat (bones, muscles, ligaments, tendons) and fat tissues. (2) Assign each non-fat tissue color that is the product of a scale factor and a custom color $C_{material}$. The scale factor can be computed by first modulating the

Material	Transfer Function	$C_{material}$	$\alpha_{material}$
bone	$C(t) = \max(\min(a(s(\mathbf{x}(t))/s_{max})^b, 0.0), 1.0) * C_{material}$, $\alpha(t) = \alpha_{material}$.	(244, 214, 145)	1.0
muscle		(255, 98, 56)	1.0
ligament		(170, 170, 170)	1.0
tendon		(255, 255, 255)	1.0
fat	$C(t) = \rho_{fat}(s(\mathbf{x}(t)))/\rho_{fat\_max} * C_{material}$, $\alpha(t) = \rho_{fat}(s(\mathbf{x}(t)))/\rho_{fat\_max} * \alpha_{material}$.	(177, 122, 101)	0.6

Table 4.1: **Transfer functions for interior-emphasized style.** t denotes the distance to the camera; $\mathbf{x}(t)$ denotes the sampling position at distance t; $s(\mathbf{x})$ denotes the MRI scalar value at sample position $\mathbf{x}$; s_{max} denotes the maximum MRI value; $\rho_{fat}(s)$ denotes the frequency of MRI value s within fat region; ρ_{fat_max} denotes the maximum frequency of MRI values in the fat region. Column "Transfer Function" gives the formula to compute color $C(t)$ and opacity $\alpha(t)$ at sample position. Columns $C_{material}$ and $\alpha_{material}$ define the custom color and opacity for each material. All values are carefully chosen to best visualize the data.

normalized MRI value $s(\mathbf{x}(t))/s_{max}$ at sample point $\mathbf{x}(t)$ with a multiplier a and an exponent b, where s_{max} denotes the maximum MRI value in the dataset. Then, the scale factor is clamped to $[0, 1]$. (3) Assign non-fat tissue constant opacity $\alpha_{material}$. (4) Assign the fat tissue color that equals the product of the normalized frequency of the MRI values within the fat region $\rho_{fat}(s(\mathbf{x}(t)))/\rho_{fat\_max}$ and a custom color, where ρ_{fat_max} denotes the maximum frequency of MRI values in the fat region. (5) Assign the fat tissue opacity that equals the product of the normalized frequency of the MRI values within the fat region and a custom opacity. Table 4.1 listed the transfer function for each material; all values are carefully chosen to best visualize the data.

For internal organs, MRI values in these regions play an important part in the color transfer function, and hence internal organs are highlighted with MRI intensity. The reasons to use the frequency of MRI values are twofold. First, in the past, histogram analysis has been utilized to extract tissue features [19][23], and it is also intuitive to highlight the data in regions where the peak of the frequency histogram is reached. Second, in particular for fat tissue, we found that in the wrist region, the MRI intensity in the dataset has lower quality than in other regions (Figure 4.1a). This introduces a substantial amount of low MRI values in this region, and the MRI values of the skin happen to be in the same range. Therefore, by utilizing the

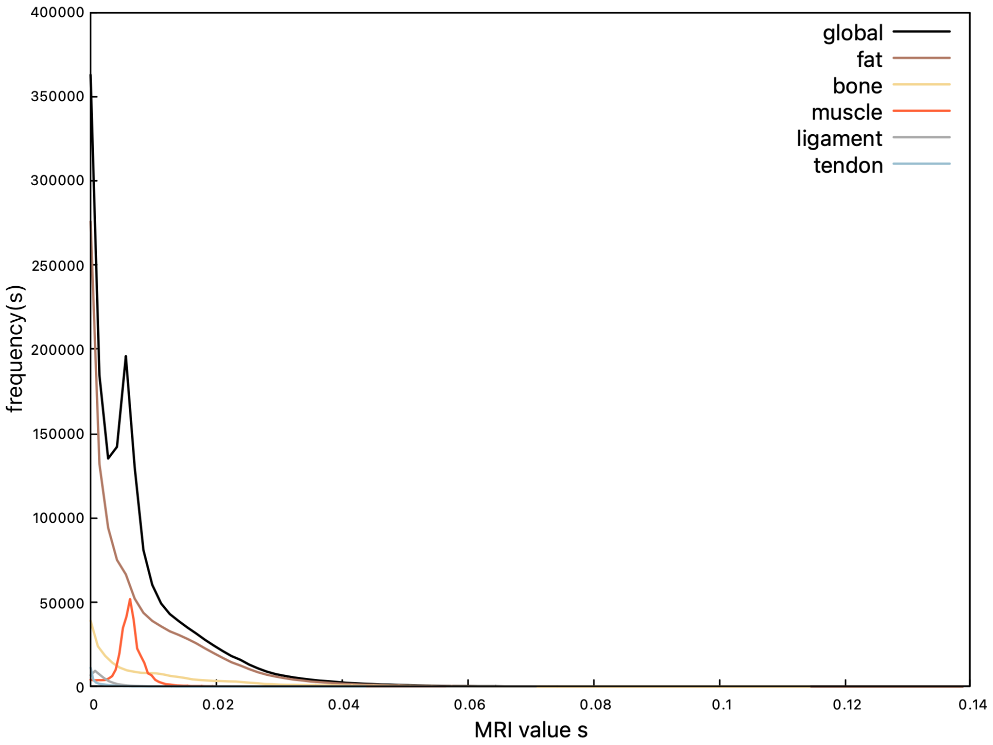


Figure 4.8: Frequency histogram of MRI values in the neutral pose.

frequency of the MRI values within fat to scale its color and opacity, we highlight the skin and partially hide the subcutaneous fat tissue, which has higher MRI intensity. Note that low intensity voxels in the wrist area are also emphasized, but they do not occlude the internal organs. Figure 4.8 shows the frequency histogram of the MRI value in all regions.

4.4.2.2 Fat-Emphasized Style

In the “fat-emphasized” style, the subcutaneous fat tissue is no longer translucent. Instead, it is “in focus” and features significant opacity. Other tissues have constant but contrasting colors, which provides the context of shape and spatial relation and highlights the black superficial veins presented in the fat layer, such as the traverse natatory veins, palmar digital veins, the superficial venous palmar arch, and dorsal metacarpal vein, as illustrated in Figure 4.9. These effects are more obvious on the dorsal side because the dorsal veins are closer to the skin. We highlight the corresponding veins in Figure 4.10 for reference.

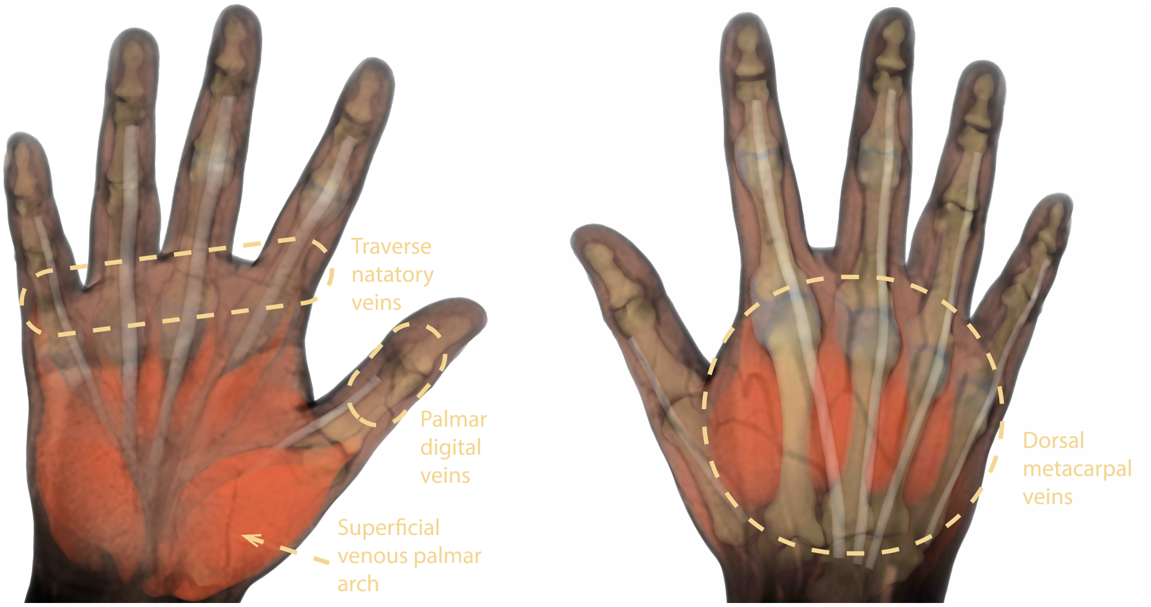


Figure 4.9: **Volume rendering with fat-emphasized style.** Fat layer is "in focus" and has significant opacity, while internal organs provide the "context" of shape and spatial relations. Several superficial veins underneath the skin are visible and annotated, such as the traverse natatory veins, palmar digital veins, superficial venous palmar arch, and dorsal metacarpal veins.

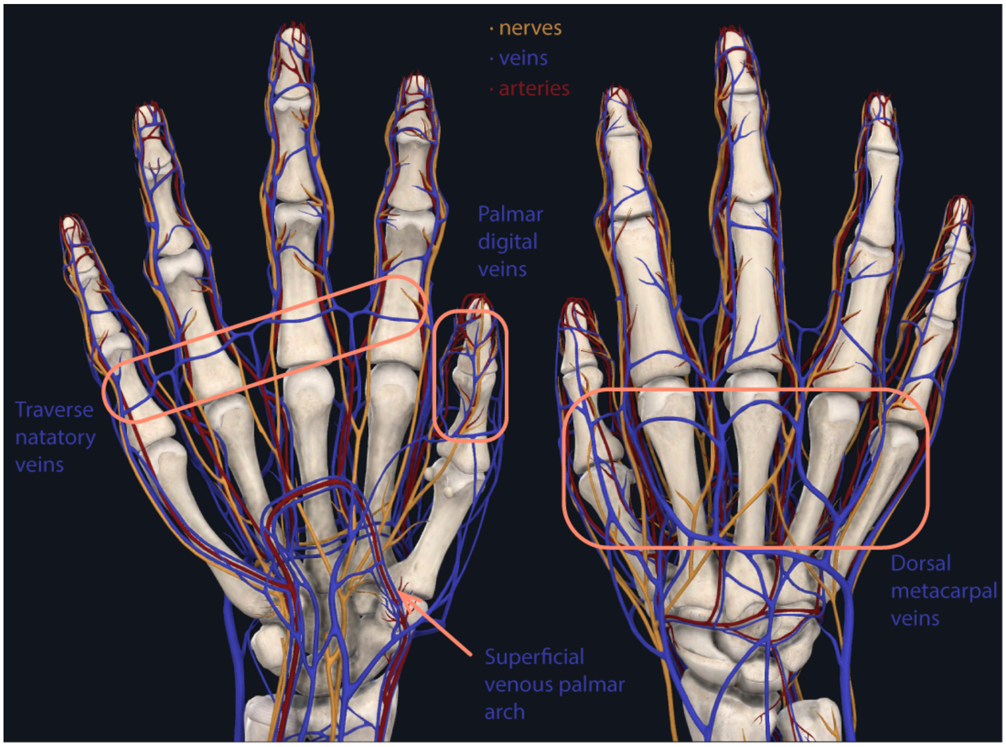


Figure 4.10: **Venous system of the hand with annotation.** © 2022 Elsevier. Image adapted, with permission, from https://3d4medical.com under student license.

Material	Transfer Function	$C_{material}$	$\alpha_{material}$
bone	$C(t) = C_{material},$ $\alpha(t) = \alpha_{material}.$	(244, 214, 145)	1.0
muscle		(255, 98, 56)	1.0
ligament		(170, 170, 170)	1.0
tendon		(255, 255, 255)	1.0
fat	$C(t) = \max(\min(a(s(\mathbf{x}(t))/s_{max})^b, 0.0), 1.0) * C_{material},$ $\alpha(t) = \alpha_{material}.$	(177, 122, 101)	0.6

Table 4.2: **Transfer functions for the fat-emphasized style.** t denotes the distance to the camera; $\mathbf{x}(t)$ denotes the sampling position at distance t; $s(\mathbf{x})$ denotes the MRI scalar value at sample position $\mathbf{x}$; and s_{max} denotes the maximum MRI value. The column "Transfer Function" gives the formula to compute color $C(t)$ and opacity $\alpha(t)$ at a sample position. Columns $C_{material}$ and $\alpha_{material}$ define the custom color and opacity for each material. All values are carefully chosen to best visualize the data.

The "fat-emphasized" appearance style is achieved as follows. (1) Divide objects into two groups: non-fat tissues and fat tissue. (2) Assign each non-fat tissue a constant color and a constant opacity. (3) Similar to the non-fat tissue in the "interior-emphasized" style, assign the fat tissue color that is the product of a scale factor and a custom color. The scale factor is the same as before, and can be computed by first modulating the normalized MRI value at the sample point with a multiplier and an exponent. Then, the scale factor is clamped to $[0, 1]$. (4) Assign the fat tissue a constant opacity. Table 4.2 gives the transfer function for each category.

It is important to note that no contrasting media such as those utilized in magnetic resonance angiography (MRA) are involved during the scanning, nor do we have segmented the veins. The veins are present because veins have lower MRI intensity compared to other tissues in the fat layer. For future work, more advanced transfer functions can be devised to utilize gradient magnitude (proposed by Leovy et al. [20]), in order to visualize the boundary surfaces of veins, possibly also arteries and nerves, and de-emphasize the homogeneous tissue in the fat layer.

4.5 Pre-Multiplication with Opacity, and Final Composition Equations

In Chapter 2, associated colors [36] were assumed, which consist of color components that are already weighted by their corresponding opacity. An alternative description uses color components that have not been pre-multiplied with opacity, i.e., non-associated colors. The colors given in Table 4.1 and Table 4.2 are non-associated colors. Hence, the previous equations have to be modified to allow for non-associated colors: original color terms have to be replaced by color terms that are explicitly weighted by opacity.

For example, C_i needs to be substituted by $\alpha_i C_i$ in the iterative compositing equations. With non-associated colors, the front-to-back compositing Equation 2.8 is replaced by

$$\begin{aligned} C'_i &= C'_{i-1} + (1 - \alpha'_{i-1})\alpha_i C_i, \\ \alpha'_i &= \alpha'_{i-1} + (1 - \alpha'_{i-1})\alpha_i. \end{aligned} \tag{4.2}$$

Similarly, the back-to-front compositing scheme is changed from Equation 2.9 to

$$C'_i = \alpha_i C_i + (1 - \alpha_i) C'_{i+1}. \tag{4.3}$$

We implemented front-to-back composition for non-associated colors using Equation 4.2. This yields a color and an opacity value for a single pixel of the final image. Multi-threading was utilized to calculate multiple pixels in parallel since the computations of different pixels are independent.

Chapter 5

Results

In this chapter, we evaluate our volume renderings method using animated sequences of MRIs. We compare the results with standard surface rendering, as well as volume rendering methods of related work.

5.1 Performance

All experiments were performed using the computing environment described in Table 5.1. Our CPU volume renderer was written in C++ and built and tested on Linux. We used Intel® Embree library for calculating ray-object intersections and oneTBB for computing multiple pixels in parallel. Python scripts were used to automate batch rendering of testing sequences.

Our animated hand MRI data and mesh geometry for testing was acquired from Mianlun Zheng, Bohan Wang, and Jernej Barbič [33][34]. These animated MRIs come from layered FEM simulation driven by five input joint animations: "close the fist" (keyframed, 132 frames), "opposition of the thumb" (keyframed, 996 frames), "performance animation" (motion-captured, 653 frames), "numbers 1-5" (motion-captured, 360 frames) and "American Sign Language" (keyframed, 732 frames).

The resolution of the input MRI volume data is $400 \times 400 \times 400$ and the voxel size is $0.64mm \times 0.64mm \times 0.64mm$. Table 5.2 shows the specifications of mesh geometry used for classification. We

Software Environment	
name	version
Operating system	Ubuntu 20.04
Linux kernel	5.13.0-44-generic
g++	12.0.1
Embree	3.4.0
oneTBB	2021.5.1
Python	3.8.10
CPU Specifications	
Number of CPUs	1
CPU model	Intel® Core™ i7-7700K
Number of cores	8
Processor base frequency	4.2 GHz
Max turbo frequency	4.5 GHz
Memory Specifications	
Memory size	40 GB
Memory speed	2400 MHz DDR4

Table 5.1: Software environment and hardware specification.

rendered these five motion sequences with two types of transfer functions (fat-emphasized, interior-emphasized) with three camera configurations (front, back, side), thus generating $5 \times 2 \times 3 = 30$ image sequences in total. On average, it takes 3.7 seconds and 4.7 seconds to render an image of 1024×1024 resolution with fat-emphasized and interior-emphasized style of transfer functions, respectively. Table 5.3 gives the time cost for each sequence. The maximum memory usage was 612.3 MB.

tissue type	# meshes	# vertices	# triangles
bones	23	788/8,291/3,353/77,111	1,572/17,838/6,764/15,576
muscles	17	1,089/10,924/3,247/55,205	2,174/21,844/6,442/109,506
tendons	10	3,000/9,032/7,613/76,129	5,996/18,060/15,222/152,218
ligaments	6	663/1,658/1,051/6,305	1,322/3,312/2,098/12,586
skin	1	18,105	36,206

Table 5.2: **Specifications of meshes of all hand tissues.** Column "# meshes" shows the number of meshes. Columns "# vertices" and "# triangles" give the minimum/maximum/average/total numbers of vertices and triangles for each mesh group, and numbers of vertices and triangles for the skin mesh.

sequences	fat-emphasized			interior-emphasized			n_f	t (hr)
	t_{ff} (s)	t_{fb} (s)	t_{fs} (s)	t_{if} (s)	t_{ib} (s)	t_{is} (s)		
"Close the fist"	3.60	3.54	3.45	4.68	5.24	4.05	132	0.9
"Opposition of the thumb"	3.87	3.82	3.76	4.81	5.04	4.49	996	7.1
"Performance animation"	3.65	3.66	3.50	4.52	5.14	4.14	653	4.5
"Numbers 1-5"	3.79	3.79	3.63	4.68	5.12	4.54	360	2.6
"American Sign Language"	3.67	3.64	3.50	4.49	4.85	4.44	732	5.0

Table 5.3: **The time cost for each animation sequence.** In the "fat-emphasized style" column, we give the average rendering time per frame for this style with front view (t_{ff}), back view (t_{fb}), and side view (t_{fs}). Similarly, in the column "interior-emphasized style", we show the same for this style with front view (t_{if}), back view (t_{ib}), and side view (t_{is}). Column n_f and t give the total number of frames and total render time for each sequence.

5.2 Results and Comparisons

Figure 5.1 gives the volume rendering results of the sequence “American Sign Language” with the interior-emphasized style. The hand was posed into letters A-E and rendered from both front and back views. In these renderings, color and opacity are consistent across the range of motion. The internal structures are clearly shown, the boundaries between tissues are smooth and clear, and the semitransparent skin provides the context of hand shape. On the palmar side, muscles are shown with rich texture due to the usage of MRI values in its transfer function; on the dorsal side, the anatomy of metacarpals can be easily identified for the same reason. The outer bone layer (compact bone) is shown in black, the bone center (bone marrow) is shown in yellow, and the two bone ends are shown in dark yellow.

Figure 5.2 shows the results of sequence “numbers 1-5” with the fat-emphasized style, in which the hand was posed into numbers 1 to 5 and rendered from the front and back views. The appearance of the hand and internal organs are consistent across the range of motion. Constant but contrasting colors of the internal tissues provide shape cues for all the inner structures. The superficial veins beneath the skin are clearly shown in black, and are nicely animated across the range of motion. One particular advantage we have gathered from our experiments is that using the "fat-emphasized" style, any FEM simulation instabilities in the fat layer are much easier to identify, compared to the surface rendering of the skin.

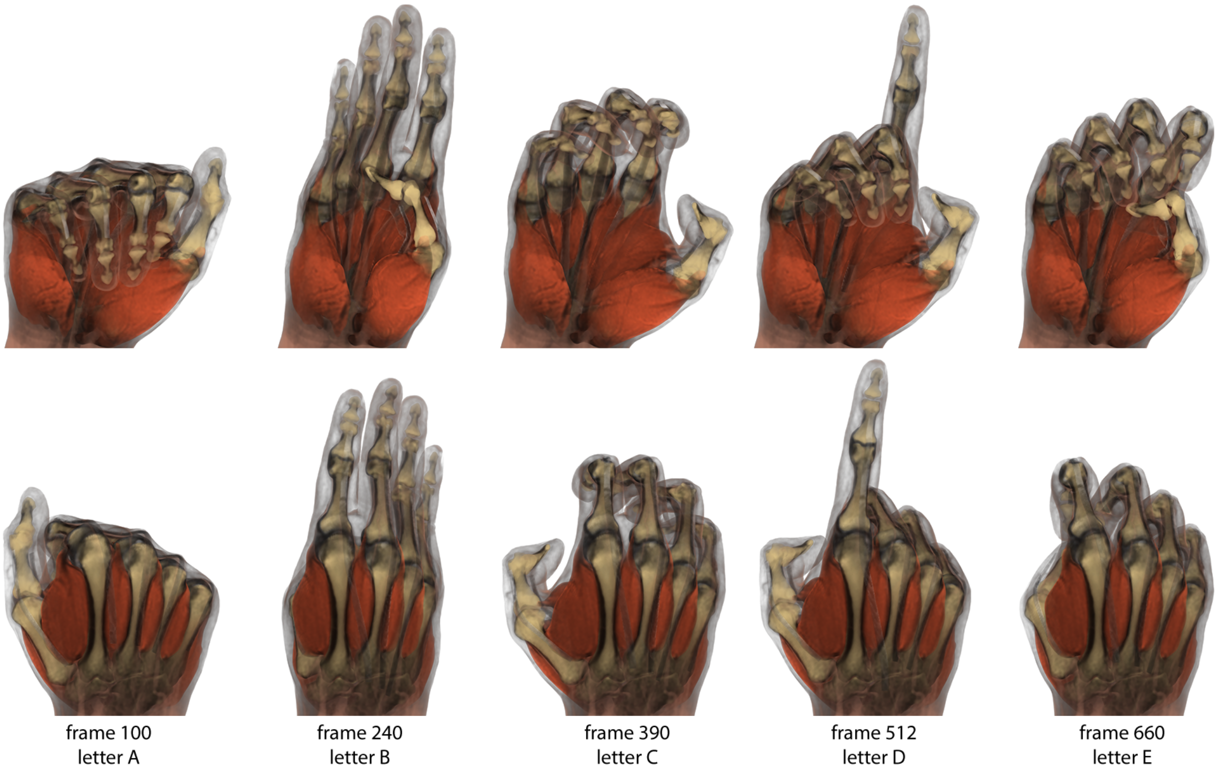


Figure 5.1: **"American Sign Language"**: Letters A to E, rendered with interior-emphasized style; front view (top row) and back view (bottom row). The internal structures are clearly shown, the boundaries between tissues are smooth and clear, and the semi-transparent skin provides the context of the hand shape.

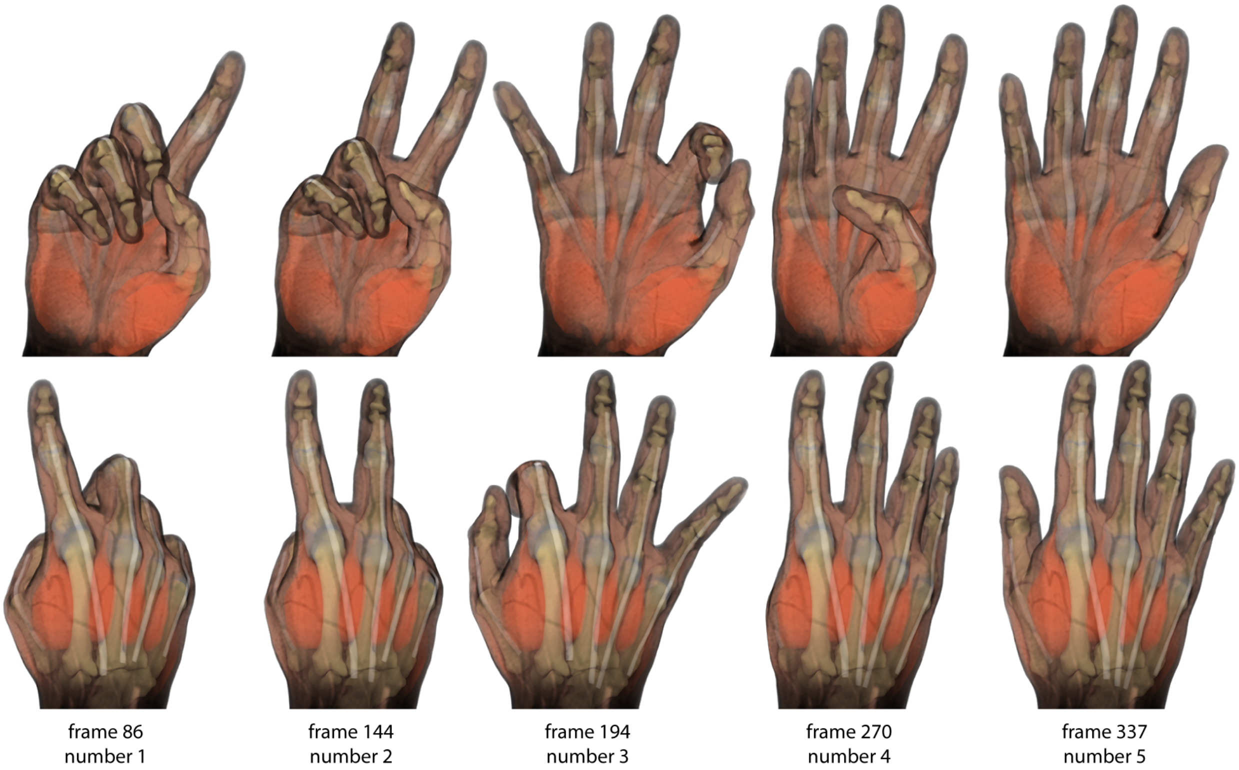


Figure 5.2: **"Numbers 1-5"**, rendered with fat-emphasized style from front view (top) and back view (bottom). The appearance of the hand and internal organs are consistent across the range of motion. Constant but contrasting colors of the internal tissues provide shape cues; the superficial veins beneath the skin are clearly shown in black and are nicely animated across the range of motion.

Figure 5.3 shows a comparison of fat-emphasized style and interior-emphasized styles, using the sequence "opposition of the thumb" (side view). It is important to note that these two styles are designed to emphasize different structures while preserving overall context, not to compete with each other. The visual difference between the two styles are mainly: (1) the anatomy of the bone is clearly shown in the interior-emphasized style, especially in middle and proximal phalanges and metacarpals, whereas the bone tissue in the fat-emphasized style merely provides the shape of its outer surface; (2) The shapes and spatial relation of tendons and ligaments can be recognized in the interior-emphasized style, but not as easily as in the fat-emphasized style, due to low MRI densities in these regions; (3) in interior-emphasized style, skin and wrist region have significant color and opacity, and the subcutaneous fat tissue is partially hidden, whereas in the fat-emphasized style, subcutaneous fat tissue has significant color and opacity, and skin and wrist region are close to black; (4) superficial veins are easier to be identified in the fat-emphasized style compared to the other style.

In terms of camera extrinsics (position and orientation), different views provide insight into different aspects of the hand. Specifically, in the interior-emphasized style, the front view emphasizes palmar muscles and phalanges due to the spatial relations between muscles and bones. In the back view, the metacarpals and carpals that were occluded in the front view, are now visible. In addition, the side view provides insights into the spatial relationships between tendons, ligaments, and interphalangeal joints. Similarly, the fat-emphasized style not only provides the shape cues for the above-mentioned structures, but also displays the variation of densities and superficial veins within fat tissue.

Figure 5.4 shows the comparison between surface rendering and our volume rendering in the “Close the fist” motion. The skin surface is rendered with Pixar RenderMan. Albedo and displacement textures are acquired from [33]. Although these images looks appealing due to the usage of textures and illumination, they do not show the internal anatomy. In addition, superficial veins that are visible in our fat-emphasized renders cannot be shown easily with surface rendering. Surface rendering of internal structures was done

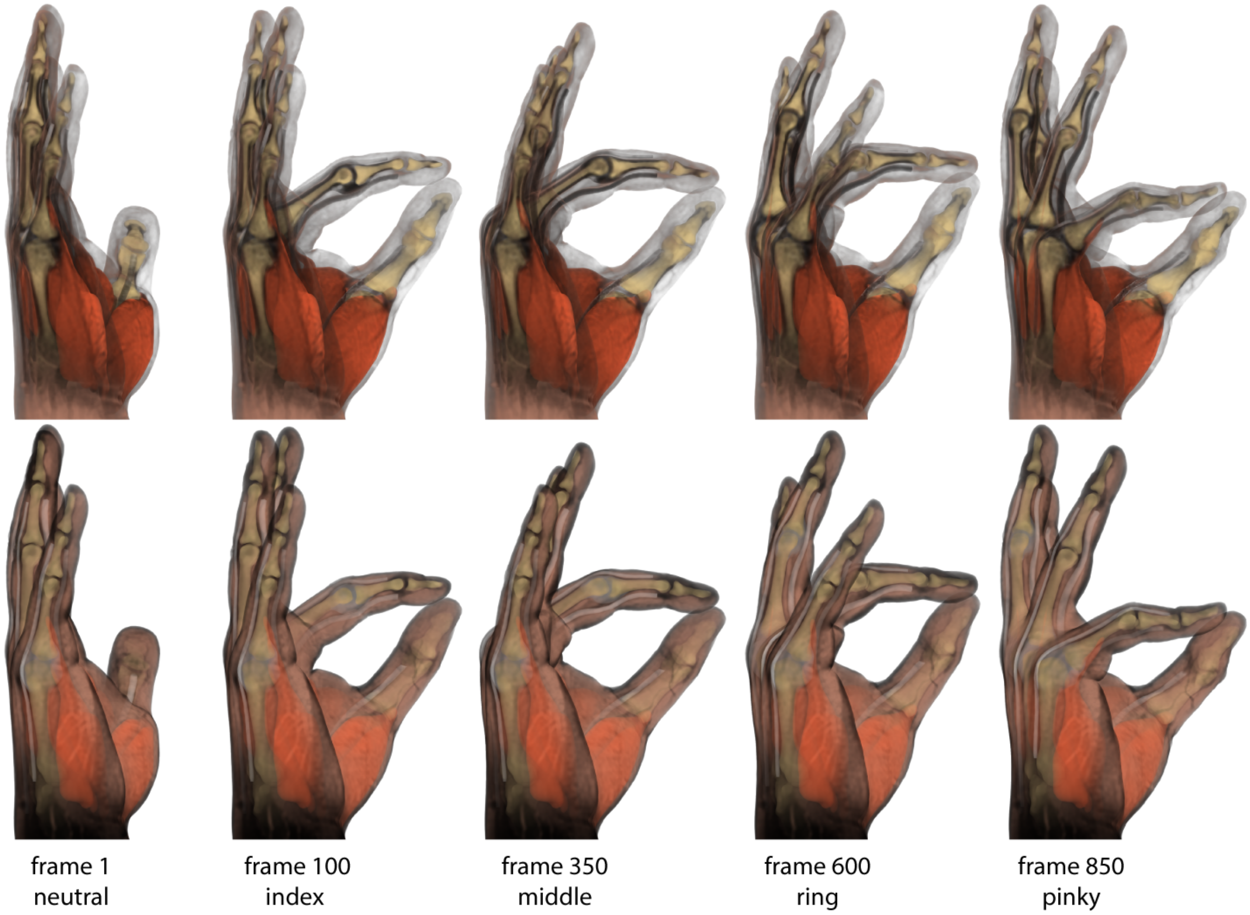


Figure 5.3: **"Opposition of the thumb"**, rendered from the side view in the interior-emphasized (top) and fat-emphasized (bottom) styles. The visual difference between the two styles are mainly: (1) the bone anatomies are clearly shown in the interior-emphasized style; (2) The shapes and spatial relations of tendons and ligaments are easier to identify in the fat-emphasized styles; (3) in the interior-emphasized style, skin and wrist regions have significant color and opacity, and the subcutaneous fat tissue is partially hidden; in contrast, in the fat-emphasized style, subcutaneous fat tissue has significant color and opacity, and the skin and wrist regions are close to black; (4) superficial veins are easier to identify in the fat-emphasized style.

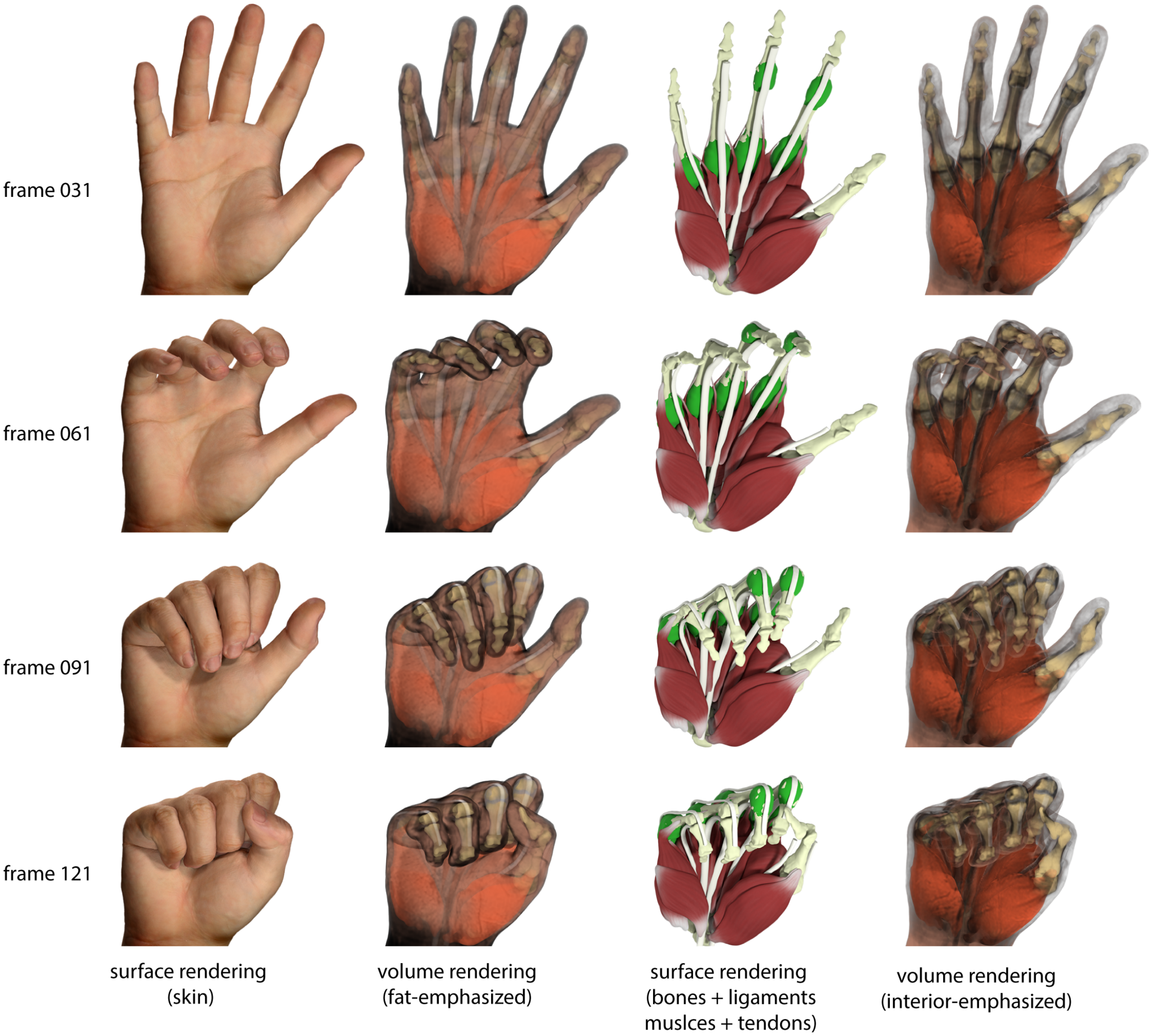

frame 031
frame 061
frame 091
frame 121
surface rendering (skin)
volume rendering (fat-emphasized)
surface rendering (bones + ligaments muslces + tendons)
volume rendering (interior-emphasized)

Figure 5.4: **Comparison between standard surface rendering (left) and our volume rendering (right) in the "close the fist" motion.** Surface renderings of the skin mesh and internal organs were created using Pixar RenderMan and Maya Arnold, respectively. Compared to surface rendering, our results are not photo-realistic due to the lack of textures, scattering, and global illumination. However, surface rendering cannot show the outer surface and internal structures simultaneously unless transparency is used, which will diminish the benefit of textures. Furthermore, surface rendering suffers from artifacts introduced by interpenetration between internal structures.

using Maya Arnold. Compared to our results, the surface rendering of the internal organs looks more photo-realistic, but does not show the outer surface of the hand. In addition, the textures of the muscles do not correspond to any real data. Moreover, surface rendering suffers from artifacts due to penetrations between different structures, e.g., ligaments and bones, tendons and muscles. These problems are solved in our MRI rendering method, thanks to our priority assignment of different tissues. It is important to note that not only single material assignment is possible, but a linear combination of different materials is also easily achievable in volume rendering.

We also compared our renders to related work. In Figure 5.5, we visually compared our results with [Rhee et al. 2010] [28], in similar poses. Similar to our method, they also have two rendering styles, one emphasizing the internal organs and the other emphasizing the fat tissue. Compared to them, in the first style, the bones and other important organs, such as tendons and muscles, are highlighted in our results with clear boundaries. In the second style, in their results, the bones and some superficial veins (above the thumb's metacarpal) are visible, but the boundaries between different tissues are not as clear as ours.

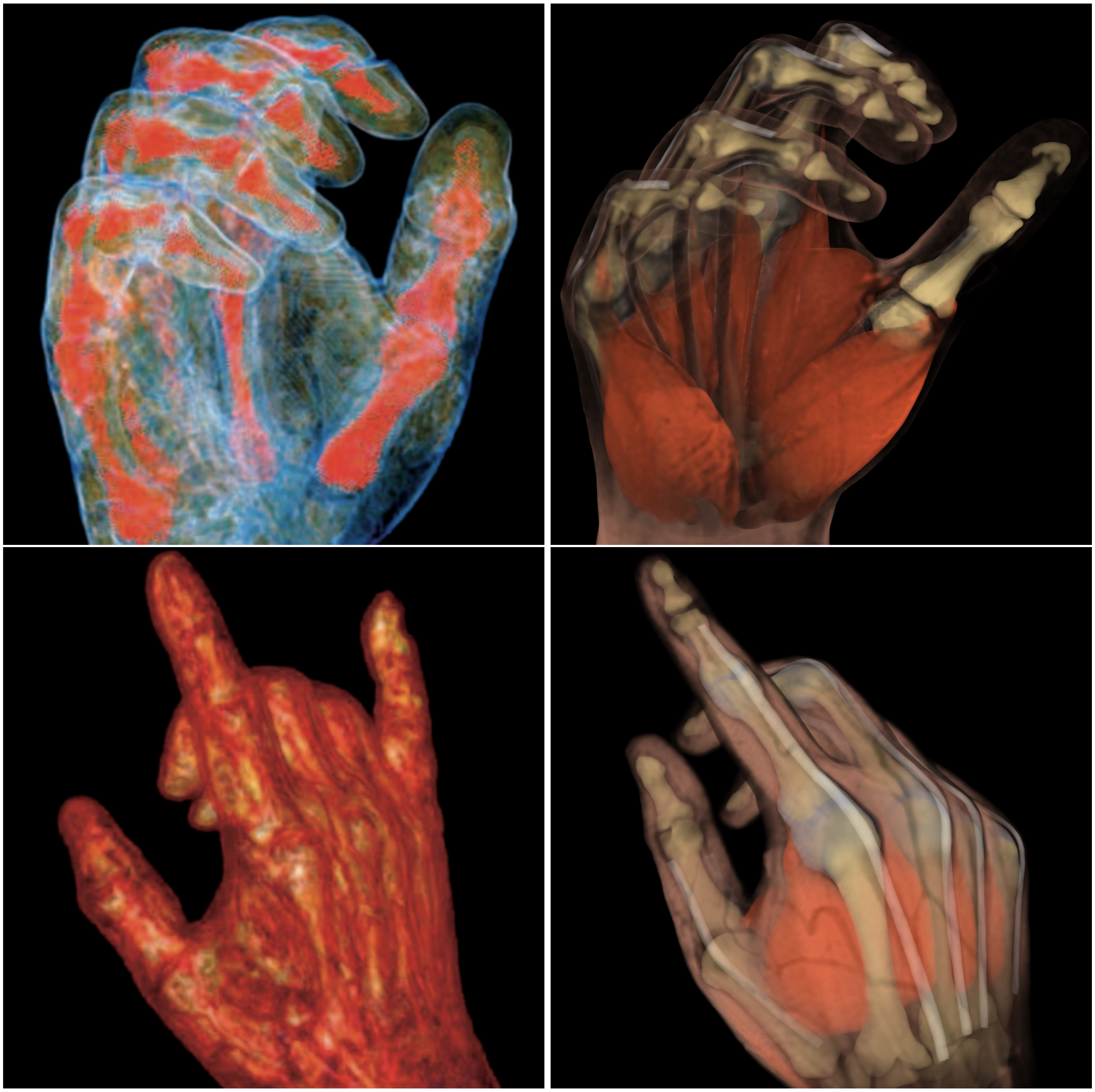


Figure 5.5: **Comparison of volume renders of [Rhee et al. 2010] (left) and our method (right).** © 2010 IEEE. Reprinted, with permission, from Rhee et al.[28].

Chapter 6

Conclusion

6.1 Summary

The main contributions of this master thesis are:

- The development of a volume rendering pipeline that implements the state-of-the-art volume rendering method, namely the volume ray casting algorithm.
- Demonstrating how the geometric shape of interior structures can be utilized to define the transfer function at each sample point, in a manner that resolves the ambiguities in areas of intersection, and removes staircasing artifacts.
- The design of two groups of transfer functions capable of emphasizing different aspects of the hand anatomy.
- The evaluation of the pipeline both in terms of rendering performance and image quality.

6.2 Future Work

We would like to improve the rendering performance of the pipeline and achieve interactive rendering by converting our CPU implementation to the GPU, so that the process of iterative design of transfer functions can be more efficient.

In addition, to improve volume rendering quality, it would be beneficial to include scattering into our optical model and implement photo-realistic Monte-Carlo ray casting, such as [29], [16], [10].

Finally, it would be interesting to explore more advanced transfer functions, and combine them with other rendering techniques, such as surface rendering, non-photorealistic rendering, or maximum intensity projection.

Bibliography

[1] Stephan Arens and Gitta Domik. “A Survey of Transfer Functions Suitable for Volume Rendering.” In: *VG@ Eurographics*. 2010, pp. 77–83.

[2] Arthritis. *Hand and Wrist Anatomy website*. https://www.arthritis.org/health-wellness/about-arthritis/where-it-hurts/hand-and-wrist-anatomy. 2022.

[3] Stefan Bruckner, Sören Grimm, Armin Kanitsar, and M Eduard Groller. “Illustrative context-preserving exploration of volume data”. In: *IEEE Transactions on Visualization and Computer Graphics* 12.6 (2006), pp. 1559–1569.

[4] Jesus J Caban and Penny Rheingans. “Texture-based transfer functions for direct volume rendering”. In: *IEEE Transactions on Visualization and Computer Graphics* 14.6 (2008), pp. 1364–1371.

[5] Brian Cabral, Nancy Cam, and Jim Foran. “Accelerated volume rendering and tomographic reconstruction using texture mapping hardware”. In: *Proceedings of the 1994 symposium on Volume visualization*. 1994, pp. 91–98.

[6] George G Cameron and PETER E Undrill. “Rendering volumetric medical image data on a SIMD-architecture computer”. In: *Proceedings of the Third Eurographics Workshop on Rendering*. 1992, pp. 135–145.

[7] Cleveland Clinic. *Blood Vessels website*. https://my.clevelandclinic.org/health/body/21640-blood-vessels. 2022.

[8] Carlos Correa and Kwan-Liu Ma. “Size-based transfer functions: A new volume exploration technique”. In: *IEEE transactions on visualization and computer graphics* 14.6 (2008), pp. 1380–1387.

[9] Klaus Engel, Markus Hadwiger, Joe M Kniss, Aaron E Lefohn, Christof Rezk Salama, and Daniel Weiskopf. “Real-time volume graphics”. In: *ACM Siggraph 2004 Course Notes*. 2004, 29–es.

[10] Franz A Fellner. “Introducing cinematic rendering: a novel technique for post-processing medical imaging data”. In: *Journal of Biomedical Science and Engineering* 9.3 (2016), pp. 170–175.

[11] Markus Hadwiger, Christoph Berger, and Helwig Hauser. "High-quality two-level volume rendering of segmented data sets on consumer graphics hardware". In: *IEEE Visualization, 2003. VIS 2003.* IEEE. 2003, pp. 301–308.

[12] Gordon Kindlmann and James W Durkin. "Semi-automatic generation of transfer functions for direct volume rendering". In: *IEEE Symposium on Volume Visualization (Cat. No. 989EX300).* IEEE. 1998, pp. 79–86.

[13] Gordon Kindlmann, Ross Whitaker, Tolga Tasdizen, and Torsten Moller. "Curvature-based transfer functions for direct volume rendering: Methods and applications". In: *IEEE Visualization, 2003. VIS 2003.* IEEE. 2003, pp. 513–520.

[14] Joe Kniss, Gordon Kindlmann, and Charles Hansen. "Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets". In: *Proceedings Visualization, 2001. VIS'01.* IEEE. 2001, pp. 255–562.

[15] Joe Kniss, Gordon Kindlmann, and Charles Hansen. "Multidimensional transfer functions for interactive volume rendering". In: *IEEE Transactions on visualization and computer graphics* 8.3 (2002), pp. 270–285.

[16] Thomas Kroes, Frits H Post, and Charl P Botha. "Exposure render: An interactive photo-realistic volume rendering framework". In: *PloS one* 7.7 (2012), e38586.

[17] Jens Krüger and Rüdiger Westermann. "Acceleration techniques for GPU-based volume rendering". In: *Visualization Conference, IEEE.* IEEE Computer Society. 2003, pp. 38–38.

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

[19] David H Laidlaw, Kurt W Fleischer, and Alan H Barr. "Partial-volume Bayesian classification of material mixtures in MR volume data using voxel histograms". In: *IEEE transactions on medical imaging* 17.1 (1998), pp. 74–86.

[20] Marc Levoy. "Display of surfaces from volume data". In: *IEEE Computer graphics and Applications* 8.3 (1988), pp. 29–37.

[21] Patric Ljung, Jens Krüger, Eduard Groller, Markus Hadwiger, Charles D Hansen, and Anders Ynnerman. "State of the art in transfer functions for direct volume rendering". In: *Computer Graphics Forum.* Vol. 35. 3. Wiley Online Library. 2016, pp. 669–691.

[22] William E Lorensen and Harvey E Cline. "Marching cubes: A high resolution 3D surface construction algorithm". In: *ACM siggraph computer graphics* 21.4 (1987), pp. 163–169.

[23] Claes Lundstrom, Patric Ljung, and Anders Ynnerman. "Local histograms for design of transfer functions in direct volume rendering". In: *IEEE Transactions on visualization and computer graphics* 12.6 (2006), pp. 1570–1579.

[24] Nelson Max. “Optical models for direct volume rendering”. In: *IEEE Transactions on Visualization and Computer Graphics* 1.2 (1995), pp. 99–108.

[25] Johns Hopkins Medicine. *Anatomy of the Hands website.* https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/anatomy-of-the-hand. 2022.

[26] Michael Meißner, Jian Huang, Dirk Bartz, Klaus Mueller, and Roger Crawfis. “A practical evaluation of popular volume rendering algorithms”. In: *Proceedings of the 2000 IEEE symposium on Volume visualization.* 2000, pp. 81–90.

[27] Christof Rezk-Salama, Markus Hadwiger, Timo Ropinski, and Patric Ljung. “Advanced illumination techniques for gpu-based volume raycasting”. In: *ACM SIGGRAPH Asia Course Notes* 10.1508044.1508045 (2008).

[28] Taehyun Rhee, John P Lewis, Ulrich Neumann, and Krishna Nayak. “Scan-based volume animation driven by locally adaptive articulated registrations”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.3 (2010), pp. 368–379.

[29] Christof Rezk Salama. “Gpu-based monte-carlo volume raycasting”. In: *15th Pacific Conference on Computer Graphics and Applications (PG’07).* IEEE. 2007, pp. 411–414.

[30] Andreas Tappenbeck, Bernhard Preim, and Volker Dicken. “Distance-based transfer function design: Specification methods and applications.” In: *SimVis.* 2006, pp. 259–274.

[31] Takashi Totsuka and Marc Levoy. “Frequency domain volume rendering”. In: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques.* 1993, pp. 271–278.

[32] Allen Van Gelder and Kwansik Kim. “Direct volume rendering with shading via three-dimensional textures”. In: *Proceedings of 1996 Symposium on Volume Visualization.* IEEE. 1996, pp. 23–30.

[33] Bohan Wang, George Matcuk, and Jernej Barbič. “Hand Modeling and Simulation Using Stabilized Magnetic Resonance Imaging”. In: *ACM Trans. on Graphics (SIGGRAPH 2019)* 38.4 (2019).

[34] Bohan Wang, George Matcuk, and Jernej Barbič. *Hand MRI dataset.* http://www.jernejbarbic.com/hand-mri-dataset. 2020.

[35] Lee Alan Westover. “Splatting: a parallel, feed-forward volume rendering algorithm”. PhD thesis. The University of North Carolina at Chapel Hill, 1991.

[36] Craig M Wittenbrink, Thomas Malzbender, and Michael E Goss. “Opacity-weighted color interpolation, for volume sampling”. In: *Proceedings of the 1998 IEEE symposium on Volume visualization.* 1998, pp. 135–142.