Facilitating the Design of Multidimensional and Local Transfer Functions for Volume Visualization

Advanced School for Computing and Imaging

This work was carried out in the ASCI graduate school.
ASCI dissertation series number 144.

A catalogue record is available from the Library Eindhoven University of Technology
ISBN: 978-90-386-1029-0

Printed by PrintPartners Ipskamp, Enschede, The Netherlands

Financial support for the publication of this thesis was kindly provided by Philips Medical systems Nederland B.V. (Healthcare Informatics – Research and Advanced Development), TU/e and ASCI.

Facilitating the Design of Multidimensional and Local Transfer Functions for Volume Visualization

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr.ir. C.J. van Duin, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op woensdag 20 juni 2007 om 13.00 uur

door

Petr Šereda

geboren te Plzeň, Tsjechië

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. F.A. Gerritsen
en
prof.dr.ir. B.M. ter Haar Romeny

Copromotor:
dr. A. Vilanova i Bartrolí

Contents

Chapter 1

Introduction

1.1 Visualization

Visual perception is our major source of information about the world around us. The well-known observation by William Glasser says that "We learn 10% of what we read, 20% of what we hear, 30% of what we see, ..." This suggests that presenting information in the form of images is more effective than, e.g., as plain text. Images help us to understand and remember complex information. If no actual image is available, we often try to construct it mentally by imagining given pieces of information. Such a process happening in our minds is called "mental visualization". If the written or spoken information is unclear or too complex, however, the mental visualization may not be possible. Such situations, when a proper image is indispensable, seem to prove the proverb "An image is worth a thousand words."

Fast development of modern technologies causes a constant growth of the data amount and complexity that users have to process. This fact has increased the need for computer-assisted visualization. Computer visualization has been recognized as an independent discipline since the late 1980's. As the technology of graphical displays improves, one can generate higher quality, more persuasive images. Visualization is, however, not only the art of generating spectacular images out of dry data. One of the visualization challenges is to present the information in a clear way that avoids wrong interpretations. Since "seeing is believing", it is one of the responsibilities of visualization not to be misleading.

The term *volume data* is used for a 3D image. The 3D image can be also looked at as a stack of 2D images that correspond to slices of a 3D object. Volume visualization is a specialized discipline that deals with volume data. Visualization of volume data can help to get an insight into, e.g., real medical data, simulated data and complex mathematical equations. The research presented in this thesis deals with volume data obtained by scanning real 3D objects. Scanned volume data is typically used in medicine for a non-invasive view into the patient's body, in industry to reveal faults in materials or constructions, in geology, etc. Although the techniques presented in this work aim at visualization of scanned volume data in general, the motivation of the work as well as most of the datasets come from the medical field.

1.2 Scanned volume data

There are four major 3D scanning techniques (modalities) used in the (bio-) medical field: CT, MRI, PET/SPECT, and US. Some of them (e.g., CT and US) have been also widely used in other fields.

Computed Tomography (CT) measures the absorption of x-rays in the scanned material. The x-rays are sent into the scanned object and detected on the opposite side. By irradiating the object from many different directions a 3D image can be constructed. Denser materials absorb more radiation and appear brighter in the image. CT scans are frequently used in both the medical and industrial area. The advantage of CT is a relatively high scanning speed and image resolution. The drawback is the radiation dose received by the scanned object. This is especially crucial in medical area, where the trade-off between the image resolution and the radiation dose received by the patient needs to be considered. An example of a CT scanner can be seen in Figure 1.1.

Magnetic Resonance Imaging (MRI) measures properties of materials in a magnetic field. First, a strong magnetic field is used to align hydrogen protons in the object being scanned. Then a sequence of magnetic pulses is applied that changes the orientation of the protons. The way the materials react to the pulses and the time it takes them to re-align back with the magnetic field can be detected in any point of the scanned object. In the medical practice there are many scanning protocols that exploit various tissue characteristics and help to establish contrast in the images. MR imaging is well suited for imaging soft tissues, which might be difficult to see, e.g., in CT images. An important advantage, compared to CT, is the absence of a radiation dose. On the other hand, MR imaging requires a relatively long scanning time. MR images are also known for containing a larger amount of image artifacts, such as bias or noise, that make their visualization difficult.

Figure 1.1: *A Philips CT scanner.*

Nuclear Imaging techniques, such as Positron Emission Tomography (PET) and Single Photon Emission Computed Tomography (SPECT), use small amounts of radioactive isotopes to highlight areas of abnormal metabolism. After injection into the body, the radioactive material is absorbed by healthy and diseased tissues at different rates. These differences in radiation can be detected by the scanner. The volume showing faster than usual metabolism might be, e.g., a malignant tumor. Since these techniques do not directly show the patient's anatomy, they are often coupled with another technique that is able to supply such information, e.g., SPECT/CT.

Ultrasonography (US) uses high frequency sound waves that penetrate the tissue and reflect back. The echoes are detected and transformed into an image. The main advantages are the portability and inexpensiveness of the system as well as the ability to generate live moving images. Ultrasound is considered a very safe imaging modality which makes it suitable for applications such as scanning the fetus. Ultrasound has, however, problems with penetrating deep into the body as well as through hard tissues such as bone. That is one of the reasons why US imaging suffers from a high level of noise and cannot be used in all anatomical regions.

1.3 Volume visualization and transfer functions

Volume visualization is mainly used to view and inspect internal structures of 3D objects without having to physically dissect them. In the medical field, for example, the views of the patient's anatomy may help to make a diagnosis.

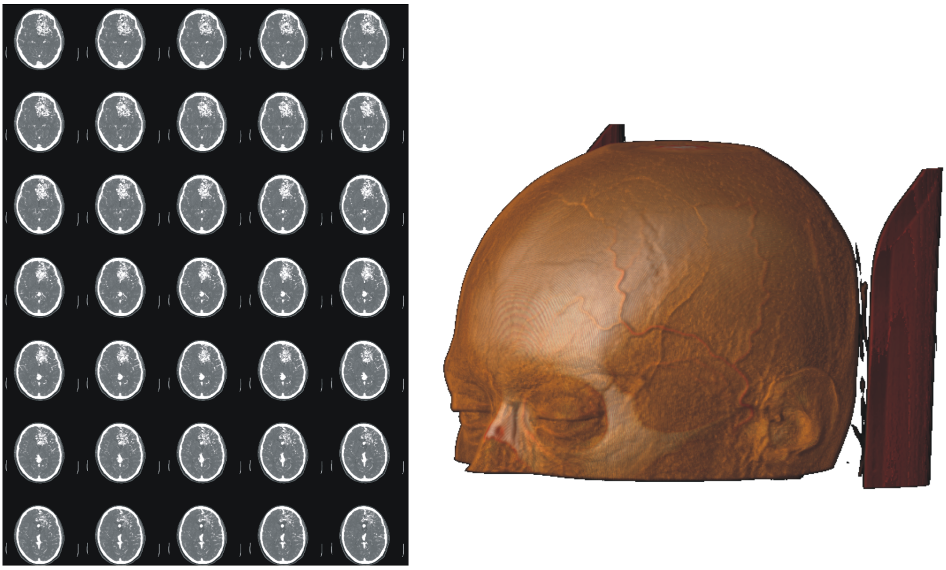


Figure 1.2: *A CT dataset of a human head viewed as a set of 2D slices (left) and as volume rendering (right).*

If the data slices are viewed (as shown in the left of Figure 1.2), there are only few relatively simple parameters needed for adjusting the visualization, such as contrast and brightness. Because of its simplicity, slice-by-slice viewing is commonly used in medical practice by radiologists making their diagnosis. Browsing through the 2D slices can be, however, a time-consuming process, since the scanned data may consist of thousands of slices. Furthermore, the user needs to mentally reconstruct the 3D shape of the objects and their spatial relations. In some complex situations this might be a very difficult or practically impossible task.

With volume rendering the volume data is shown as a projection of 3D objects (Figure 1.2 right), where the sizes, shapes, and relations between objects are usually easier to observe. Volume rendering has, therefore, the potential to facilitate the mental reconstruction of volume data and to make such applications as medical diagnosis more efficient.

Volume rendering, however, requires a larger number of parameters that need to be properly chosen. The complexity of the settings makes volume rendering difficult to use. The most critical appear to be the choice of the proper opacity and color for different parts of the data. The opacity and color settings (optical properties) are realized by a so-called *Transfer Function* (TF). The TF uses the measurements done by the scanner as a domain and converts them to optical material properties (color, opacity) that can be visualized. The common approach is to define the TF manually. That is often a cumbersome trial-and-error process since there may be no straightforward relationship between the TF and the final result. In order to overcome this drawback and to allow a wider use of volume rendering, the process of TF definition needs to be facilitated. This is the main motivation for the research work presented in this thesis.

1.4 Overview of the thesis

The thesis deals with transfer functions used in volume rendering. Two important aspects are addressed here: the TF domain and the TF definition. Emphasis is put on the visualization of boundaries between materials as well as on the intuitive user interaction with the TF itself. The research strategy was to develop general approaches and frameworks that could be possibly adapted to specific circumstances of a given application. The content of the remaining chapters is as follows:

Chapter 2 first establishes the position of direct volume rendering within the volume visualization methods. Then, the role of transfer functions is explained. The main part of the chapter deals with state of the art approaches toward TF design. Special emphasis is put on the role of the TF domain and the automation of the design process that helps to facilitate user interaction.

Chapter 3 addresses the visualization of material boundaries. First, it introduces LH space as a novel TF domain and shows its benefits in the visualization of material boundaries over existing approaches. The properties of the LH space and the appearance of the boundaries in the LH histogram are discussed. Second, an extended classification of boundaries is introduced which allows the visualization of both sides of the boundaries independently. Finally, it is shown that the LH space can help to define similarity measures and be used, e.g., in a boundary-based region growing approach.

Chapter 4 uses the properties of the LH space and the LH histogram in order to automate the TF design. A new framework is shown that allows the user to interact with a hierarchy of clusters and combine intuitive clustering criteria.

It is shown that the LH histogram could be used to generate the initial clusters and to define clustering criteria.

Chapter 5 introduces the concept of local transfer functions (LTF) that aim to overcome some of the limitations of the standard global TF. A general framework for the LTF is presented. Each piece of the framework is discussed and possible implementations are suggested.

Finally, Chapter 5.7 summarizes the research and achieved results, and provides suggestions for further research.

Chapter 2

Transfer Functions for Volume Visualization

2.1 Definitions

Volume data is a discrete representation of a continuous function $f(\vec{x})$, where $\vec{x} \in R^3$. In this thesis we assume the data represents a scalar function, i.e., $f(\vec{x}) \in R$. Volume data is usually acquired by sampling (scanning) real objects, simulation, or modeling. The discrete samples are typically defined on a regular rectilinear grid and stored as a 3D array of values called volume (a 3D image). Each discrete value and area of influence is also referred to as a *voxel* (volume element, 3D pixel). In order to estimate values in between the sample locations, the values of neighboring voxels are interpolated. The commonly used trilinear interpolation takes into account the 8 surrounding voxels. These 8 voxels form a so-called *cell* (see Figure 2.1).

Intensity at position $\vec{x}$ is the scalar data value $f(\vec{x})$, i.e. the value measured by the scanner.

Material is used for part of the data that has the same physical properties (density, chemical composition, etc.). Materials in medical datasets can be also referred to as *tissues*. Parts belonging to the same material are usually expected to appear with the same intensity (i.e., the same data value). *Material/tissue intensity* is then used to refer to the intensity with which the material/tissue appears in the scanned dataset. Different materials can appear with the same intensity and one material can have different intensities. Depending on the scanning modality and protocol, the contrast between different materials in the volume data may vary.

Object is a part of the data having an abstract meaning for the user. It can be, e.g., certain body part, an organ, parts containing certain material, parts having certain properties, etc. One object can consist of multiple materials and one material can be present in several objects.

Transfer function (TF) is a mapping from data properties to optical properties $\mathcal{T} : R^h \rightarrow \text{Opticals}$, where h is the number of data properties. The most commonly used property is the intensity $f(\vec{x})$. TFs based on the intensity often assume that materials correspond to objects and that points belonging to the same material appear with the same intensity. Objects of interest can then be selected by selecting corresponding intensities. Color and opacity are commonly used as opticals, i.e., the range of the TF. The TF will be discussed in more detail in section 2.3.

Segmentation maps the spatial position of a sample in R^3 to a label L
$\mathcal{S} : R^3 \rightarrow L$. The process of obtaining a segmentation is, in principle, a labeling (classification) of points in which their intensity or spatial position are often used. Segmentation approaches may range from simple spatial divisions to complex model-based methods. The segmentation may also use additional or alternative data properties.

Fuzzy segmentation is a set of mappings. For each label L_i a mapping $\mathcal{F}_i$ exists that maps the positions to a probability that the point belongs to the label $\mathcal{F}_i : R^3 \rightarrow < 0, 1 >$. The output of the segmentation is a (fuzzy) labeling of the data points. In order to visualize the segmentation an additional step has to be made that maps the labeled data to optical properties. Then usually the function $\mathcal{S}$ or $\mathcal{F}_i$ is shown instead of the original volume $f(\vec{x})$.

2.2 Volume visualization

Volume data represents a visualization challenge due to its typically large size and amount of information. In order to get an insight into the content of volume data several volume visualization techniques have been developed. The goal of volume visualization techniques is to display the volume data in a 2D image, typically on the computer screen. It is not easy to display the full 3D volume data in a 2D image. Therefore, the visualization techniques aim to display and emphasize only those aspects of the volume data that are of interest to the user.

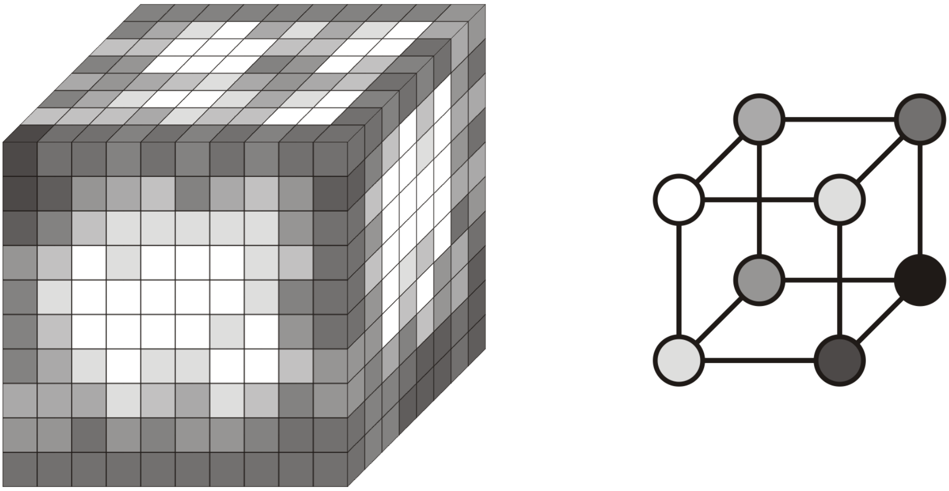


Figure 2.1: *Volume data is a 3D image composed of voxels (left). The cell representation consisting of 8 voxels (right) is commonly used in order to interpolate between the voxels.*

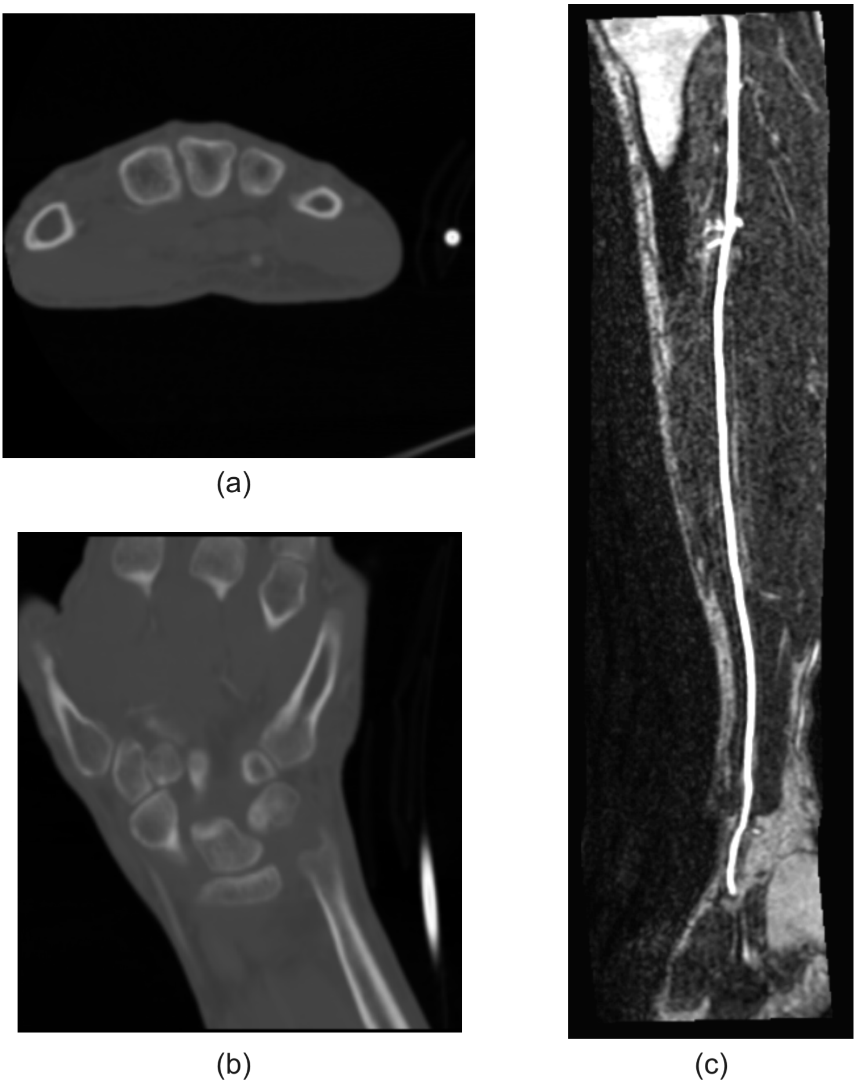
There are basically two main categories of approaches to visualize volume data: as a set of 2D views representing 2D cross-sections of the volume data or as a 3D view representing the entire volume. However, there are a number of approaches that combine both, e.g., visualization of slabs (thick slices) or 2D slices inserted into a 3D view.

2.2.1 2D visualization

2D views show volume cross-sections as a 2D image. There are no problems with occlusion: all voxels of the cross-section are visible in the image. There is usually a limited number of settings involved. Besides the position (and orientation) of the cross-section to be displayed, the contrast and brightness of the image might be adjusted. There are several approaches to generate the cross-sections. The most common techniques are:

- **Slice-based**

  The scanners typically capture the volume as a set of axis-aligned 2D slices. The straightforward approach is, therefore, to visualize the data as a sequence of these slices (Figure 2.2a). In this case the data as it is obtained by the scanner is shown. Since the slices stacked on top of each other create the volume (see Figure 2.1), it is also common to show slices perpendicular to any of the three orthogonal axes.



(a) (b) (c)

Figure 2.2: *CT scan of a hand. (a) A 2D slice as taken by the scanner, (b) A reformatted slice that can be oriented in an arbitrary direction. (c) An MR scan of a leg, the cut is curved in order to follow the vessel.*

- **Multi planar reformatting (MPR)**

  Multi planar reformatting enables the user to slice the volume in an arbitrary direction. The data is interpolated at the location of the cross-section plane. This allows the user to align the cross-section plane with the objects of interest in order to see more relevant information in the same image (see, e.g., Figure 2.2b).

- **Curved planar reformatting (CPR)**

  Curved planar reformatting extends the slicing possibilities by allowing curved cross-sections. The curved reformatting can show data along a curved plane that follows the object of interest. Typically, a path is first defined along the object. The cross-section then follows this path and can be rotated around the path. The advantage of this approach is that one can view curved objects, such as the spinal cord or the vessels, in one image in order to, e.g. inspect their diameter. Figure 2.2c illustrates an example with a curved cross-section along a vessel.

2D visualization techniques show the complete data available in the current slice or cross-section. This fact helps to ensure that all available data is presented to the user. The drawback of the 2D techniques is, however, the amount of images that need to be viewed by the user in order to see the complete 3D dataset. Currently available scanners may produce datasets containing thousands of slices. The main disadvantage of the 2D views is that the user has to mentally reconstruct the 3D information. Although radiologists are trained to do that, the mental reconstruction is often difficult due to the complexity of the data and important 3D information can be missed.

2.2.2 3D visualization

In order to visualize the 3D information in a 2D image, a number of approaches have been developed. These 3D rendering methods basically project the 3D volume data into a 2D plane. The 3D rendering methods can be divided into two main categories: *Surface Rendering* and *Direct Volume Rendering* (DVR). In general, surface rendering approaches display opaque surfaces, such as iso-surfaces (surfaces corresponding to $f(\vec{x}) = I$, where I is a given data value) or surfaces of segmented data. Direct volume rendering extends the visualization possibilities by allowing the display of not only the surfaces, but also the inside of the objects. Surface rendering can be further divided into indirect and direct. Indirect methods first extract the geometry of the object to be visualized and display it as a polygonal mesh. Probably the most commonly used method for 3D visualization is the extraction of iso-surfaces by using the marching-cubes algorithm [1]. This algorithm reconstructs an iso-surface by generating a triangular mesh that can be displayed using standard graphics hardware. DVR, on the other hand, uses directly the volume data without extracting any intermediate representations of the objects. Further in this thesis we only will focus on DVR that enables the display of the internal structure of objects and does nor require any intermediate representation of objects.

Figure 2.3 shows a typical pipeline for DVR. Volume data samples are projected in a 2D image using ray-casting. For each sample optical properties in given lighting conditions are determined. Finally, visibility issues of the samples projected to the same point are resolved. The most common projection approaches used in DVR are ray-casting [2], splatting [3, 4] and texture mapping [5, 6].

The *Transfer Function* (TF) assigns optical properties, such as color and opacity, to every data sample. The TF, which is the main subject of this thesis, will be discussed in detail in the rest of the chapter.

The *Illumination and Shading* stage of the pipeline evaluates light conditions at the sample location [7] and simulates the interaction of light with the sample. Usually, Lambert or Phong [8] models are used to simulate effects such as reflection and scattering.

In order to display the volume data in a 2D image, only limited information can be shown. Since many samples of $f(\vec{x})$ project into the same point of the 2D image (see Figure 2.3), a method needs to be defined that combines the information of all these points into one. The most commonly used methods are:

- **Maximum/minimum intensity projection (MIP/mIP)**

  The maximum and minimum intensity projections usually use neither the transfer function nor the illumination/shading. They combine the sample points contributing to the same pixel to the maximum/minimum intensity of all the points [9]. These methods are commonly used for data where points of interest have higher/lower intensity values than the rest of the data, such as the bones in a CT scan or blood vessels with a contrast medium (Figure 2.4). The most serious drawback is the lack of depth information in the images.

  A special case of the MIP technique is the *Closest Vessel Projection* (CVP) [10] or *Local MIP* (LMIP) [11] which shows the value of the local maximum closest to the observer. This may help to give better depth cues than the standard MIP.

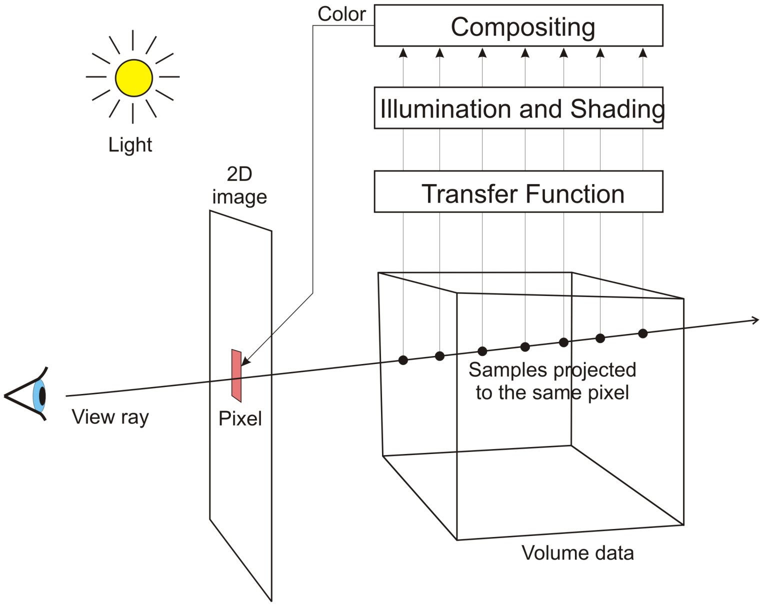


Figure 2.3: *The pipeline for direct volume rendering.*

Figure 2.4: *A MIP rendering of the CT scan of a hand.*

- **Shaded volume rendering**

  Shaded volume rendering generates more realistic-looking images by simulating the light conditions and light behavior [7, 8] (Figure 2.5). First, optical properties of the samples are determined by the transfer function. Then, the optical properties are used in the illumination/shading to evaluate the color of the light that is reflected from the light source towards the observer. Finally, the compositing takes into account the opacities of the samples, simulating the absorption of light in the material, enabling semi-transparent projections of multiple volume samples into the same image pixel. We will only consider the most commonly used absorption model, when samples are characterized by the color they reflect and by the opacity. In general, more complex illumination models could be used, enabling emission or scattering of light [7].

It is outside the scope of this thesis to give a complete overview and description of methods that can be used to implement the volume rendering pipeline. For further details on the volume rendering pipeline the reader is kindly referred to the overview of volume rendering techniques in [12, 13].

2.3 Transfer functions

The transfer function (TF) plays an important role in the volume rendering pipeline. It determines which data samples will be visible and how they will be visualized. The TF can be defined as a mapping from the transfer function domain to optical properties (see Figure 2.6)

$\mathcal{T} : R^h \rightarrow \text{Opticals}$

The transfer function domain may consist of h data properties. The TF assumes that points of interest can be distinguished by their data properties (the domain of the TF). The example in Figure 2.5 uses the data value, i.e., the range of $f(\vec{x})$, as domain. Visual contrast between objects is achieved by using different optical properties for different values of $f(\vec{x})$. Color and opacity are typically used as optical properties (i.e., the range of the TF). An appropriate opacity setting may reveal objects or their parts that would otherwise be hidden from the observer. The color establishes visual contrast between different objects of interest. Depending on the rendering model, more complex illumination coefficients [7] or complete color spectra [14] may be defined. In this thesis we only consider color and opacity as the range of the TF.

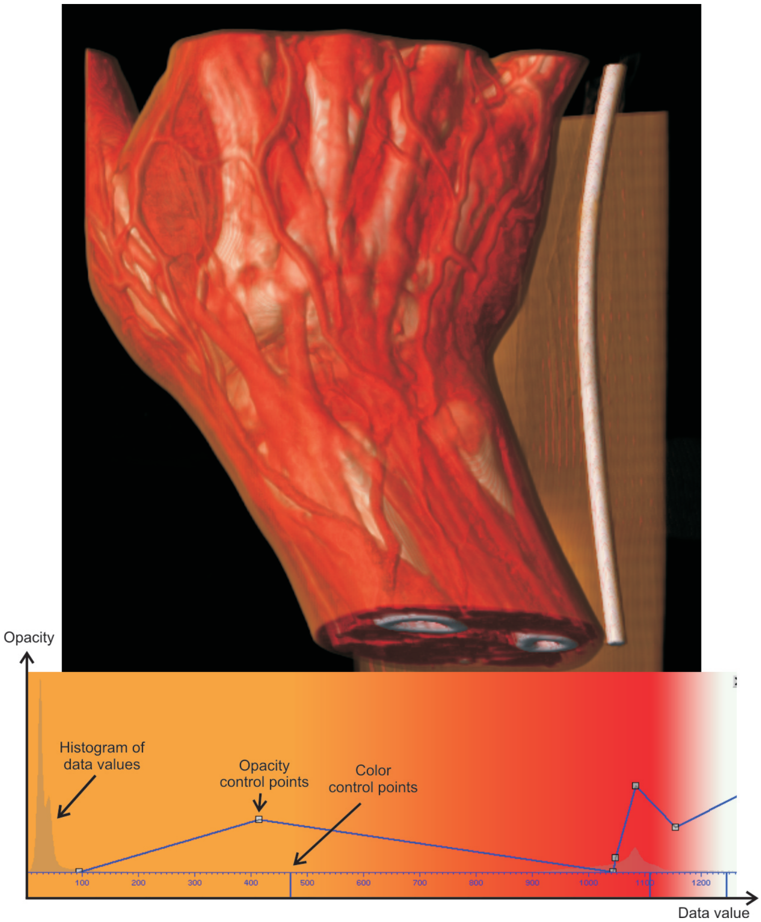


Figure 2.5: *Shaded volume rendering of a CT scan of a hand. In the bottom the transfer function interface is shown. The histogram of data values may help the user to identify the important data ranges.*

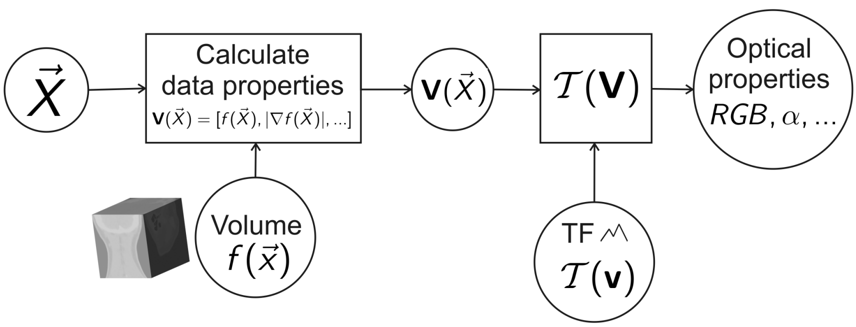


Figure 2.6: *The pipeline of transfer functions. The data properties V are computed at the sample location $\vec{X}$ and mapped by the TF $\mathcal{T}$ to optical properties.*

The shape of the TF can be complex and its design can be difficult and frustrating. It is often hard to predict how the rendering will respond to a change of the TF. The facilitation of the TF design and making it more intuitive is, therefore, an important research task that would make the use of volume rendering in practice easier. Moreover, the choice of the TF domain, as well as the set of used optical properties, have a major influence on the visual classification of the data. The following chapters discuss the TF domains and design approaches that have been presented in the literature.

2.4 TF domain

As mentioned above, the TF is a mapping from data properties to optical properties. The question is what properties are suitable for the visualization goal. In general, any data properties could be used that help to classify relevant points in the data. The TF, however, usually does not consider the spatial position in the volume. In the example from Figure 2.5 the data values $f(\vec{x})$ were considered as data property that has been mapped to color and opacity.

The following sections give an overview of some of the most commonly used data properties in the TF domains.

2.4.1 Intensity

The most commonly used property for the TF domain is the intensity (i.e., the data value $f(\vec{x})$ itself). This can be very effective when different intensities correspond to different materials. One can also compose a multi-dimensional TF domain on intensities in case of multi-modal images [15], e.g., registered images [16] taken by multiple scanners or scanning protocols.

The use of transfer functions based on the intensities has a serious drawback. One cannot determine without ambiguity whether a sampled value $f(\vec{x})$ corresponds to the material intensity at position $\vec{x}$ or whether it is result of a partial volume effect (the value is a mixture of neighborhood values) or the common assumption that $f(\vec{x})$ is a continuous function. Figure 2.7 illustrates a situation in which an iso-surface is used in order to visualize materials F_1 and F_2. This assumption of continuity can only work if all the materials in the data are present in the expected order as we go from the lower intensities to the higher intensities (or from higher to lower). In Figure 2.7b the material of intensity F_1 is missing between materials F_0 and F_2. The iso-surfaces of values F_i (solid line) and F_j (dashed line), $F_2 > F_i > F_1 > F_j > F_0$, were selected in order to visualize the spheres of material F_2 and F_1 respectively. In Figure 2.7b the dashed iso-surface, found due to the partial volume effect and due to the assumption of continuity, is falsely signalling the presence of material F_1.

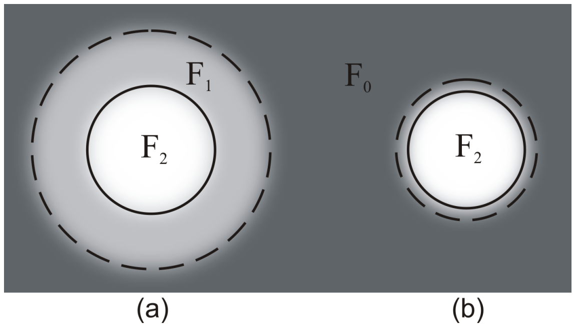


Figure 2.7: *A slice through two spheres. In (a) there is a sphere of material intensity F_2 placed inside another sphere of lower intensity F_1 such that $F_2 > F_1$. In (b) there is a second sphere of intensity F_2. In (b) the presence of material F_1 is falsely signalled. The background has intensity F_0 such that $F_2 > F_1 > F_0$. The edges are blurred.*

2.4.2 TFs based on boundaries

In addition to the intensity, the gradient magnitude $|\nabla f(\vec{x})|$ is often used to emphasize the boundaries between materials [2]. The basic assumption is that the boundaries are more important for the visual perception of the shapes and spatial relations of objects than the other parts of the volume. The opacities of samples being rendered are modulated by the local gradient magnitude. The larger the gradient, the more important the boundary is considered to be, and the sample is thus rendered with a higher opacity. Transfer functions that analyze boundaries between materials may help to solve problems such as the one illustrated in Figure 2.7.

The boundaries can be modeled as step edges blurred by the point-spread function of the scanner [17] (Figure 2.8). Kindlmann and Durkin [18] showed that the gradient magnitude $|\nabla f(\vec{x})|$ in combination with the intensity $f(\vec{x})$ reveals the boundaries as arches (see Figure 2.9). They showed the arches for the volume data by creating a 2D histogram of the intensity and the gradient magnitude. One can then distinguish between boundaries as they correspond to different arches. Kniss et al. [19, 20] used this space of arches as a 2D transfer function domain.

Kindlmann and Durkin [18] and Kniss et al. [19, 20] used the second order derivative in the gradient direction to select data samples that lie close to the edge location.

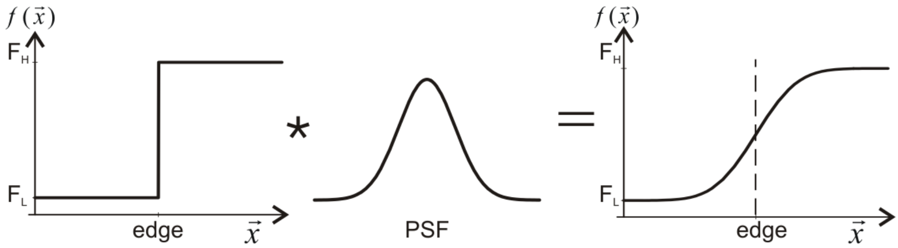


Figure 2.8: *The model of a boundary between two materials of values F_L and F_H. The step edge boundary present between two scanned materials (left) is blurred by the point spread function (PSF) of the scanner resulting in partial-volume intensities across the boundary in the dataset (right).*

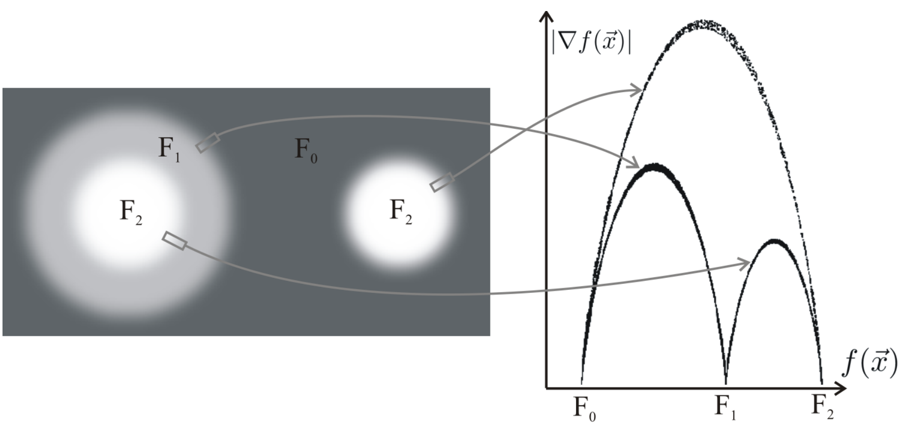


Figure 2.9: *Three different boundaries in the dataset from Figure 2.7. Each boundary appears as an arch.*

Looking at multiple data values $f(\vec{x}')$, where $\vec{x}'$ are points in the neighborhood of $\vec{x}$ may also help to identify the boundary at $\vec{x}$. Lum and Ma [21] used two samples lying in the gradient direction in order to help classify the boundary in each voxel. Their approach assumes that for voxels lying on the boundary, the two extra samples are taken in materials that form the boundary.

In Chapter 3 we present a novel TF domain, the so-called LH space, in order to classify boundaries by combining two intensities. We show there that the LH space offers a better selection of boundaries than the commonly used arches.

2.4.3 Higher order derivatives, curvatures

One can think of using higher order derivatives of $f(\vec{x})$ for highlighting other features in the data. For example, the second order derivative in the gradient direction, the Laplacian, a corner detector, the curvature or any other feature used in computer vision [22] might be used to define the TF domain. The common problem of higher order features is, however, an increased sensitivity to noise which may hamper the visual performance of the TF.

TFs based on curvatures are described in [23]. In every point the two principal curvatures κ_1 and κ_2 can be computed in the plane perpendicular to the gradient direction. The curvature κ_1 has higher absolute value than κ_2. The ratio of these two numbers can help to distinguish local shapes of the iso-surface:

1. plane (if $\kappa_1 \approx \kappa_2 \approx 0$)
2. parabolic cylinder (if $\kappa_1 > \kappa_2 \approx 0$ or $0 \approx \kappa_1 > \kappa_2$)
3. paraboloid (if $\kappa_1\kappa_2 > 0$)
4. hyperbolic paraboloid (if $\kappa_1\kappa_2 < 0$)

These two principal curvatures create a two-dimensional domain in which a TF can be defined as shown in figure 2.10. Every point in the volume corresponds to a location in this curvature-based plane which then defines its color and opacity. In this example the color scheme has been chosen so that the planar areas are rendered in green, the cylindrical in yellow and the spherical in red.

Kindlmann et al. [24] further investigated the possibilities of curvature-based transfer functions. They showed that the curvatures can be used to, e.g, emphasize ridges and valleys. They used the curvature to improve non-photorealistic contour rendering.

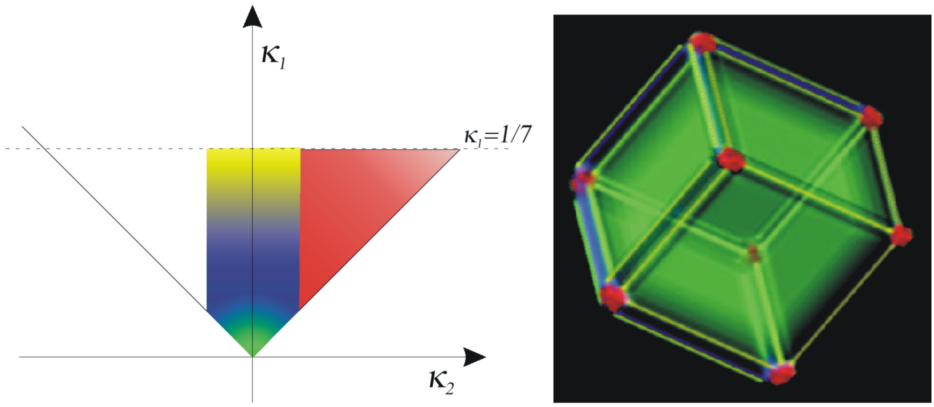


Figure 2.10: *A 2D curvature-based transfer function domain with a defined TF. Different colors depict different local shapes. In the right image the visualization is shown using the TF on the left. Image courtesy Hladůvka et al. [23].*

2.5 Defining transfer functions

In this section techniques for the definition of transfer functions are discussed. The structure of the overview is based on the amount of user interaction needed for the TF definition.

2.5.1 Manual definition

Usually, the domain of the TF is being shown to the user and some interaction tools are provided to assign optical properties to each value of the domain. The user then manually changes the shape of the transfer function, typically by moving a few reference points (see Figure 2.5) and assigning colors. However, this way of definition has some serious drawbacks. The user usually has to explore a large range of possible shapes of the TF to find a suitable one since it is not easily predictable what will be the result of the TF definition. This makes it a time consuming and possibly frustrating process. Furthermore, the user needs to have some knowledge about the TF domain and the visualization algorithm in order to fully understand what is happening. These disadvantages have been the source of motivation for developing more sophisticated, automated and intuitive interaction methods. We believe that the cumbersome definition of the TF is one of the main reasons why volume rendering is not more widely used. On the other hand, one could still argue [25] that the manual methods allow a step-by step exploration of the data avoiding possibly misleading visualizations that could be introduced by the automated methods.

2.5.2 Manual definition with assistance

This group of techniques is based on the previous manual approach. In this case, however, the user is not left to try all possible TF shapes to discover which is suitable and which not. There is additional information that guides the user through the definition process. Such extra information can point out the values of data properties on which the user should further concentrate in order to explore the interesting parts of the volume.

The histograms of intensities are commonly used to guide the user (see also the example from Figure 2.5). From such a histogram one can, for example, guess the data properties of certain objects since the objects often appear as peaks. The user can then adjust the TF accordingly in order to include or exclude an object or to change its color and opacity. The weak point of the histograms is that small objects are hardly visible in the histogram. The work of Lundström et al. [26] introduced so-called α-histograms that can emphasize peaks corresponding to small objects.

Kniss et al. [19, 20] used a set of interaction widgets to help the user interact with the 3D TF (Figure 2.11). The transfer function was based on the data value, gradient magnitude and the second directional derivative. The interaction widgets support and facilitate the exploration of data properties in searching for the optimal visualization.

A contour spectrum was introduced by Bajaj et al. [27]. For every scalar value, there is a corresponding contour (an iso-surface for the case of 3D volume data). One can observe the properties of these contours such as the surface area, volume inside or outside the contour, and the gradient integral The interesting iso-values are, e.g., those which correspond to borders between tissues. Such values can be observed from the contour spectra as peaks in the gradient integral (integral of the gradient magnitude over the iso-surface). Also features such as the ratio of the volume inside the iso-surface to the whole volume can be used.

Similar histograms to the contour spectra, however computed in a different way, were shown by Pekar et al. in [28]. Besides the characteristics used by Bajaj et al. [27], they introduced the mean gradient over the iso-surface (which does not depend on the area) and the sum of curvatures computed over the surface. All these features might be useful in some applications. See Figure 2.12 for an example of a visualization of a phantom dataset.

Another kind of a histogram-oriented guide was presented by Kindlmann and Durkin [18]. They used 3D histograms in order to observe the relationship between $f(\vec{x})$, $|\nabla f(\vec{x})|$ and the second derivative of $f(\vec{x})$ in gradient direction.

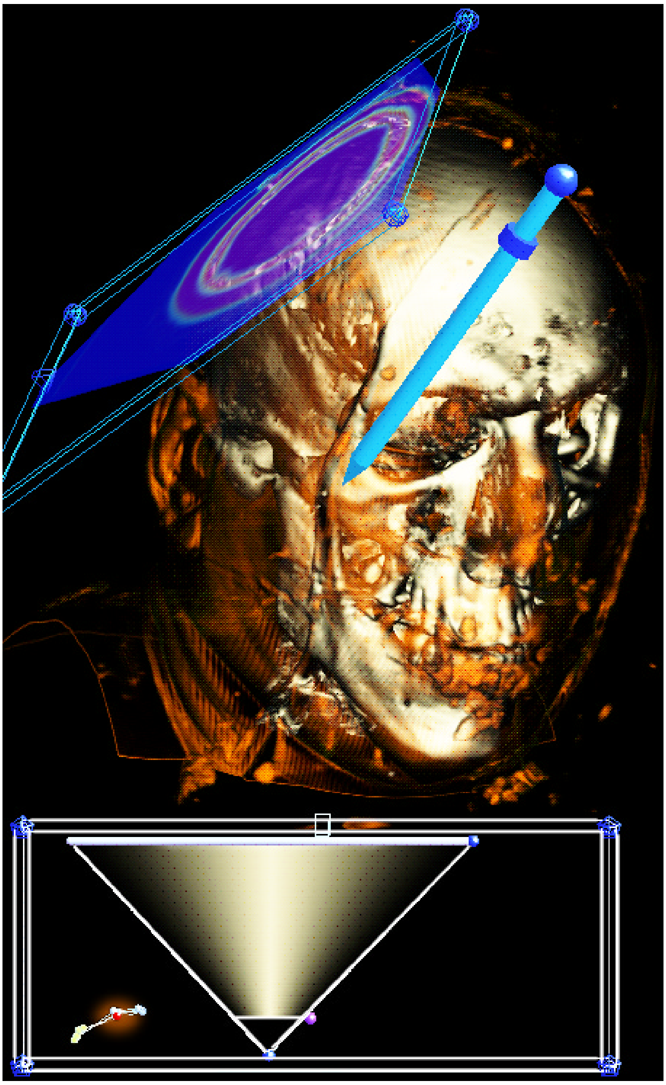


Figure 2.11: *The volume interaction widgets (pen and clipping plane in the top), the transfer function widget (the rectangle in the bottom) and the classification widgets inside the rectangle. Image courtesy Kniss et al. [19].*

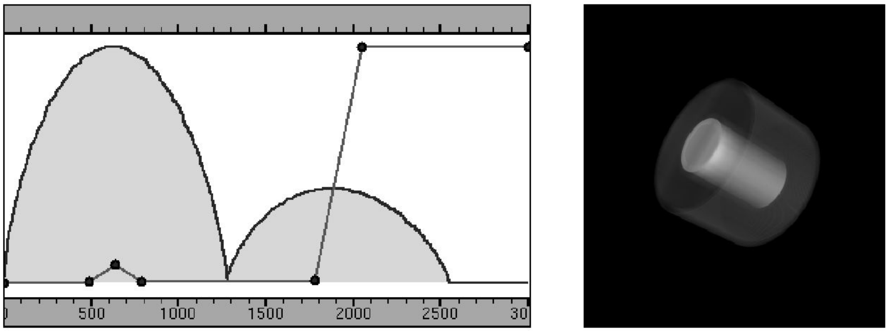


Figure 2.12: *A phantom dataset with two cylinders. Left: the gradient integral curve used as a guide for the opacity transfer function. Right: volume rendering of the dataset. Image courtesy Pekar et al. [28].*

From this relationship, one can determine the intensities that correspond to the most important tissue boundaries. This can also be used to semi-automatically generate the TF (see next section). Furthermore, by using such a histogram one can distinguish between different tissue boundaries (e.g., bone-air or skin-air).

2.5.3 Semi-automatic definition

There is another group of approaches that attempt to perform either part of or the entire TF definition automatically.

Design galleries

The principle of design galleries shown by Marks et al. [29] allows the user to choose the way the data is rendered without any direct interaction with the TF. Figure 2.13 shows an example of a design gallery that offers different combinations of tissue opacities. These sample combinations are displayed as thumbnails. The user then chooses the sample that is closest to the requirements. The fact that the user does not handle the TF directly, but only evaluates the image appearance, makes the design galleries easy to use. On the other hand, there are some problems that have to be considered. The most important is how to sample the TF domain in order to get all relevant thumbnails. This can be partially solved by introducing more iterations. In every iteration the best sample is chosen. In the next iteration its neighborhood in the TF domain is sampled at a higher resolution. The iteration stops when the user is satisfied with the result. This strategy assumes that from the offered set of thumbnails one can clearly say which thumbnail is getting closer to the visualization goal. With an increasing dimensionality of the TF domain the sampling issues become more severe.

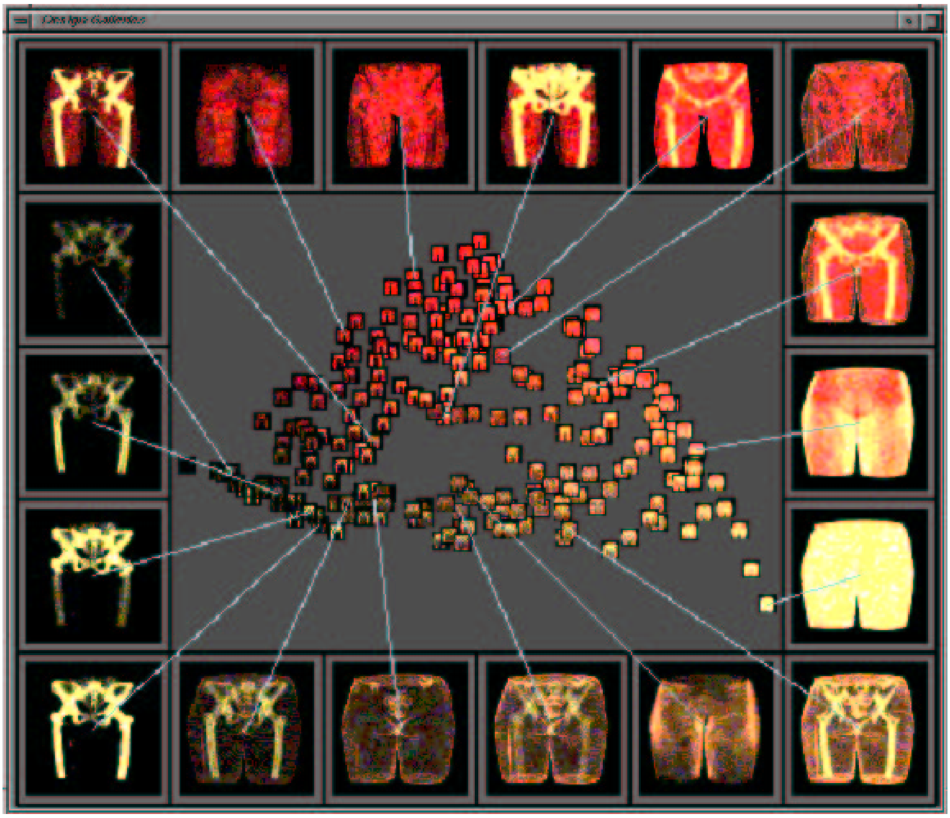


Figure 2.13: *A design gallery with two opacity transfer functions. Image courtesy Marks et al. [29].*

Producing a large number of thumbnails is computationally expensive. One may need to produce thousands of them in order to give a reasonable choice. To reduce the computation costs, thumbnails can be produced from a smaller sub-sampled volume or an alternative acceleration technique has to be used. König and Gröller [30] used a simplified design-gallery approach. Rather than having the freedom of defining a TF of an arbitrary shape and color, the definition is done in three simpler steps. First, a predefined peak (typically trapezoid) is placed to multiple sample locations through the scalar intensities. An image is generated for each sample and offered to the user in the form of a gallery. The position of samples, peak width and shape can be manually changed. The user selects a set of samples. In the next step, a color is assigned to these samples (see Figure 2.14). Finally the opacity is determined for each sample by using a design gallery view, similar to the one in Figure 2.13, and images are blended together.

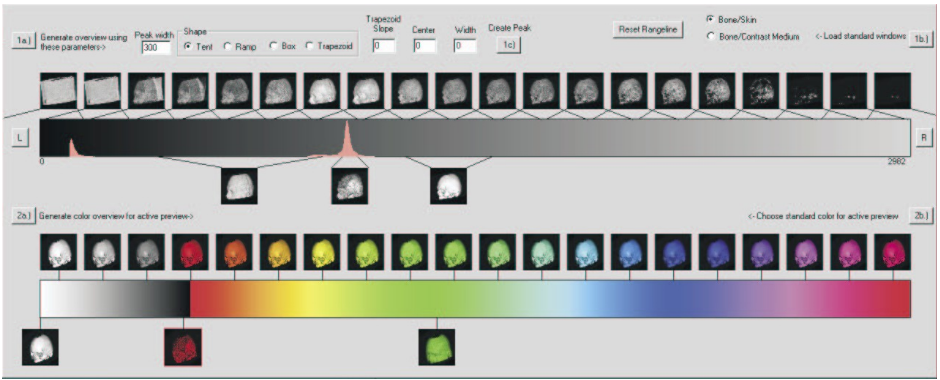


Figure 2.14: *Top: samples through the intensities and corresponding thumbnails. Bottom: assigning colors to selected samples. Image courtesy König and Gröller [30].*

Data-driven automation

Unlike the design galleries, the semi-automatic method described by Kindlmann and Durkin [18] determines the positions of the opacity peaks automatically. Based on the data analysis the peaks are placed so that the boundaries are visualized. The user can only define the shape of the peaks.

Another data-driven approach to define both the colors and opacities was described by Fujishiro et al. [31]. The Reeb graph is constructed in order to describe relations between different iso-surfaces. The critical iso-surfaces (threshold values at which objects split or merge) are then emphasized by accordingly placed opacity peaks.

In order to be able to distinguish tissues within ambiguous mixtures of materials, Lundström et al. [32, 33] extended the TF domain by using information obtained from local histograms. Further, they performed automatic tissue detection using the so-called partial range histograms. Another approach using histograms to resolve the partial volume mixtures was shown by Laidlaw et al. [34]. They used a Bayesian classifier to resolve materials in MR datasets.

Classifiers and clusterings

Automation methods based on a classifier or clustering could also be considered as a special case of the data-driven automation. With an increasing dimensionality of the transfer function, direct user interaction becomes very difficult.

Tzeng et al. [35] used a high-dimensional classification for the volume visualization. Instead of interacting with a multidimensional function, they used a learning classifier. The user interaction with the transfer function was done by painting into the data slices. The main difference to traditional transfer functions is that their classifier also used the voxel positions.

Another technique that aimed at the simplification of user interaction with the transfer function was shown by Tzeng et al. [36, 37]. They clustered the voxels into material classes by considering multiple material properties. The user then interacted directly with the clusters.

An indirect selection in the TF domain based on the intensity and gradient magnitude was shown by Huang and Ma [38]. They used a partial region growing in the volume. The selection was then defined by mapping the grown voxels onto the domain of arches. Roettger et al. [15] used a clustering method that groups two bins of the histogram if the corresponding tuples are similarly distributed in the volume.

In Chapter 4 a novel hierarchical clustering framework is introduced together with several similarity measures that enable grouping of material boundaries.

Image-driven automation

He et al. [39] suggested that the TF definition could be automatically driven by analyzing the output image. A genetic algorithm was used to develop a 1D TF. At the beginning an initial population was given consisting of various shapes. The successfulness of the generation was defined as the quality of the resulting image. The quality was computed using predefined criteria such as entropy or variance. The idea that the transfer function design can be automatically steered by the quality of the resulting visualization is very promising. However, choosing good criteria for an automatic assessment of the image quality is quite problematic.

2.6 Speed and quality of visualization

The interactivity as well as the quality of the visualization are key factors in the interaction process. The speed is important not only while interacting with the volume, but also during the transfer function definition. If, while manually designing the TF, one has to wait several seconds to get a new rendering, one tends to make larger changes in order to speed up the whole process. This could possibly result in missing some important parts of the TF space. In order to fully explore the content of the data, one needs to be able to interactively change the TF.

There are many techniques that are used to speed up volume rendering. Giving an overview of the methods would be beyond the scope of this thesis. Hardware-accelerated techniques are currently at the center of attention since the modern graphics cards can execute rendering algorithms often faster than the CPU. An overview of hardware-accelerated volume rendering can be found, e.g., in [40].

The quality of the volume rendering is influenced not only by the way the data volume is sampled, but also by the way in which the transfer function is sampled. Pre-integrated volume rendering helps to solve problems related to high frequencies in the TF by integrating the opacity function over the TF domain [41, 42].

Although both the speed and the quality of the volume rendering are related to the TF, this thesis does not focus on these aspects.

2.7 Conclusions

As can be seen from this overview, the field of transfer functions is a challenging and intensively investigated area. The TF domain and the process of defining the TF itself seem to be the most crucial issues in using the TF. Although many approaches for the TF definition have been introduced, the problem of facilitating the TF definition is still far from solved.

Introducing TF domains consisting of multiple dimensions may help to differentiate objects. However, a more complex TF domain represents a challenge for the TF definition. Therefore, the complexity of the TF domain should stay as low as possible if manual definition is used. In addition, the use of visual guides can help the user to identify the objects of interest and therefore can facilitate the interaction.

Material boundaries are often desired by the user to appear in the visualization. It is, therefore, important to have a TF domain that enables an easy selection of boundaries, either manual or automatic. In Chapter 3, we will introduce a new TF domain that facilitates both manual and automatic selection of boundaries.

Automation facilitates the TF design by reducing the amount of user interaction and shielding a possibly nonintuitive manual interface. Another important advantage of automation is the fact that the results can be easily reproduced, provided that a deterministic algorithm is used. The drawback, however, of full automation is the lack of interactivity in improving the result in case the user is not fully satisfied. In Chapter 4 we present a framework for semi-automated definition of TFs. Our approach is capable of shielding the user from the TF, yet allowing the user to explore and tune the TF by using a cluster interface.

One of the problems connected to the TF definition is its globality, i.e., the use of the same TF for the whole volume. It is difficult to tune a TF if it gives good results only in a part of the volume. In Chapter 5 we present a framework that can be used to define the TFs locally in order to adapt to possible changes of the data properties across the volume.

In this thesis we introduce several novel approaches for both the TF domain and for the TF definition. The basic idea in this thesis is to develop general techniques that facilitate the definition of transfer functions.

Chapter 3

Visualization of Boundaries Using LH Histograms

3.1 Introduction

It is not trivial to choose the domain of the transfer function, i.e., to choose data properties that enable a good distinction between objects of interest. For many applications, the attention of the user is focused on visualizing the boundaries of objects. The boundaries can reveal important information such as the shape of the object, its extent, size, and spatial relation with other objects. In order to effectively select the data points that lie on the boundary and to differentiate between boundaries, one needs to use an appropriate TF domain. It has been shown in Chapter 2 that boundaries cannot be well classified by using only intensity. Additional information such as the gradient magnitude or the boundary profile helps to differentiate boundaries.

Although the domain of intensity and gradient magnitude substantially improves the selection of boundaries, it still suffers from several problems. In Figure 3.1 an example is shown where two arches intersect. It is obvious that such intersections cannot be avoided in this domain. These overlaps cause ambiguities in the classification of boundaries since the points at overlapping areas may belong to either of the arches. Because of this, Kniss et al. [19, 20] used a threshold on the second derivative in the gradient direction to visualize only the peaks of the arches (i.e., voxels lying close to the edge). That may solve some of the overlaps. However, noise, partial volume effects and bias along the boundary are reflected in the histogram as multiple shifted or scaled copies of one arch. This causes more overlaps and makes it difficult to distinguish boundaries.

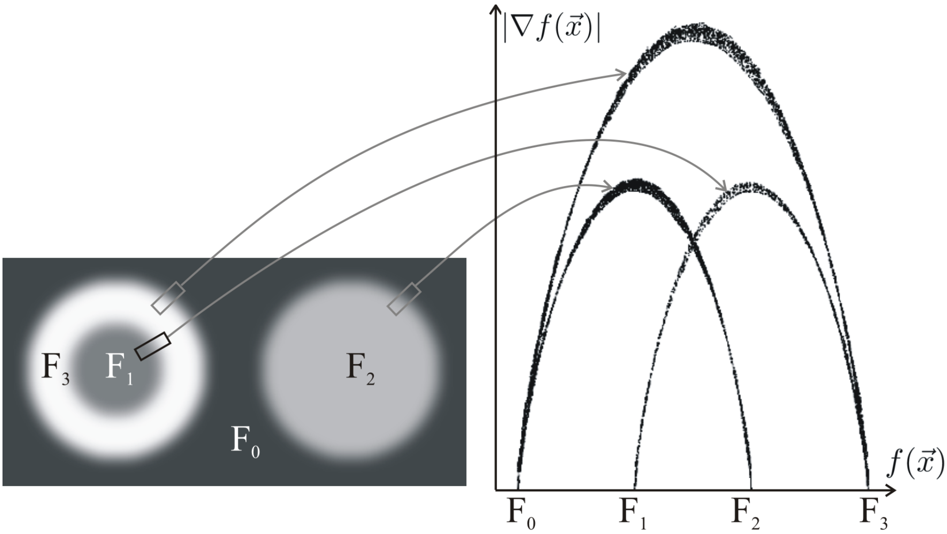


Figure 3.1: *Arches can cross due to the overlapping ranges of values.*

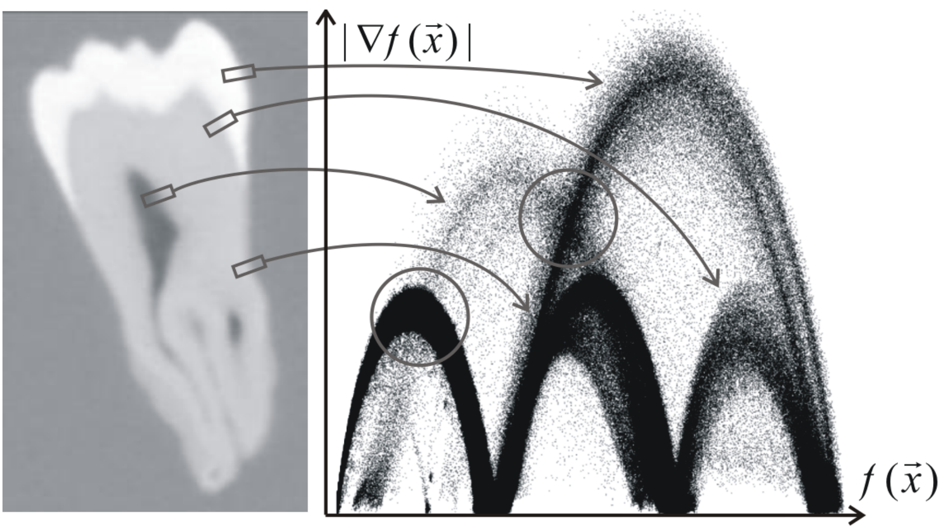


Figure 3.2: *CT dataset of a tooth (256x256x161) with corresponding arches. Two most obvious overlaps of the arches are marked by circles. Several approaches to visualize this dataset were shown in the Transfer Function Bake-Off [25].*

The approach presented in this chapter aims to improve the separability of the information shown by the arches. We detect the material values at both sides of the boundary. Knowing both values, we can construct a so-called LH (low-high) histogram (Serlie et al. [43]). However, it is not easy to find these values by detecting the start and the end of an arch (see Figure 3.2). Serlie et al. [43] used local fitting of arches. We propose an alternative method that does not require a model of the arch. Our method, in general, only assumes that the intensity profile of the boundaries is strictly monotonic. This assumption is valid for the step-edge model blurred by the PSF (Figure 2.8). This chapter presents the LH histogram as a novel multi-dimensional transfer function domain that is aimed at facilitating the selection of boundaries between materials. It further shows that the LH information can be used in segmentation algorithms such as region growing.

In the following section, we describe the construction and properties of the LH histogram. Section 3.3 shows how to use the LH histogram for transfer functions. Section 3.4 introduces an extension of the LH histogram that allows an independent classification of both sides of the boundaries. Finally, section 3.5 shows that LH values may be used to improve region-growing based segmentation.

3.2 The LH histogram

We label the higher intensity of the two materials that form the boundary F_H and the lower intensity F_L (see Figure 3.3). The LH histogram is a 2D histogram whose axes correspond to F_L and F_H. The concept of the LH histogram is similar to that of the Span Space [44]. However, in the Span Space each point is indexed by the minimum and maximum values within a cell instead of the lower and higher value of materials that form a boundary. We assume that every voxel of the data lies either inside a material or on a boundary between two materials. After finding F_L and F_H at each voxel position, we can build an LH histogram by accumulating voxels with the same $[F_L, F_H]$ coordinates.

3.2.1 Construction

For each voxel of the volume, we first determine if it lies on a boundary by looking at the gradient magnitude. Voxels having $|\nabla f| \leq \epsilon$ are considered to be inside a material and are assigned $F_L = F_H = f(\vec{x})$. Such voxels project on the diagonal in the LH histogram. The remaining voxels are supposed to belong to a boundary and we continue to determine the intensities of both materials that form the boundary.

For a non-biased data (e.g. CT) epsilon can be set to zero or smaller than the weakest boundary we want to detect. However, in data with a bias field (e.g. MR) epsilon needs to be large enough to distinguish gradients caused by the bias field from those caused by the presence of a boundary.

The lower intensity F_L and the higher intensity F_H can be found by investigating the intensity profile across the boundary. Given the voxel position, we track a path by integrating the gradient field in both directions (see Figure 3.3).

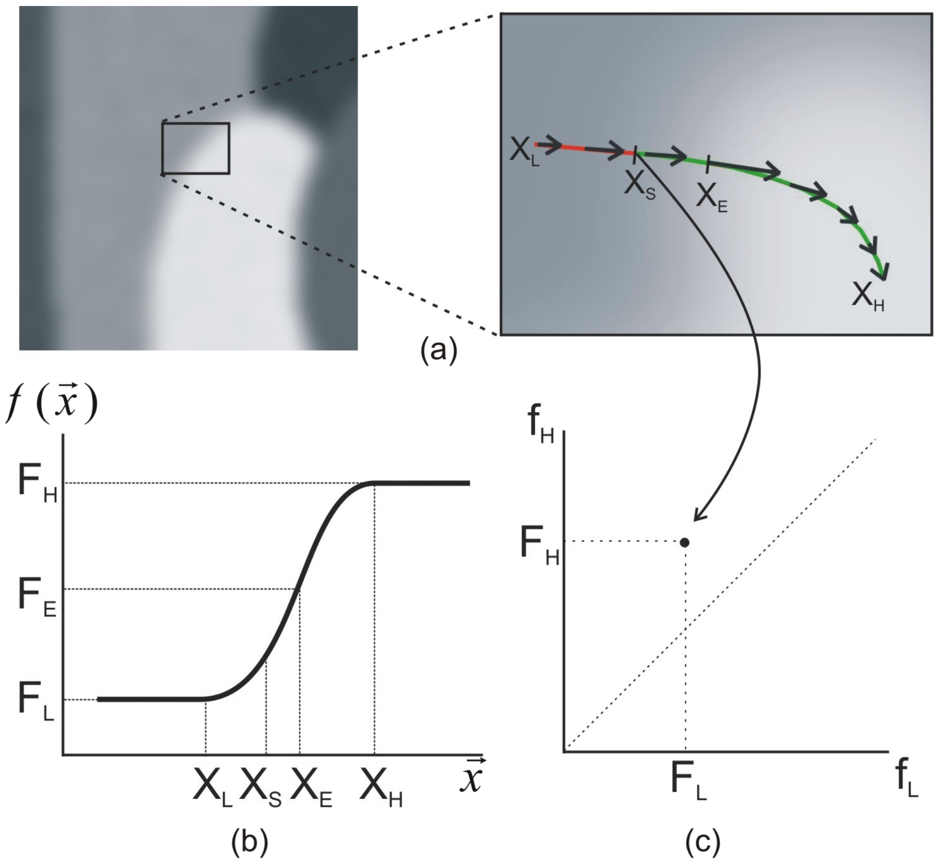


Figure 3.3: *(a) Starting at position X_S we generate a path across the boundary by integrating the gradient field. The positive gradient direction leads to X_H. Following the opposite direction ends by finding X_L. (b) The intensity profile along the path. At points X_L and X_H where the tracking stops we read the values F_L and F_H respectively. F_E is the intensity value at the edge location X_E. (c) F_L and F_H are used as coordinates in the LH space.*

The positions at which F_L and F_H are reached and the integration stops are determined by stopping criteria based on the shape of the intensity profile. The intensity profile along the path is examined while it is being constructed. Figure 3.4a shows a common type of step edge boundary that can be easily detected by looking when the profile becomes constant. Figures 3.4b and 3.4c show two common cases where the step edges are close to each other in comparison to the blurring of the point spread function. In these cases the profiles never become constant. Therefore the stopping criterion is a local extremum or an inflexion point. In real data we can find combinations of all these cases.

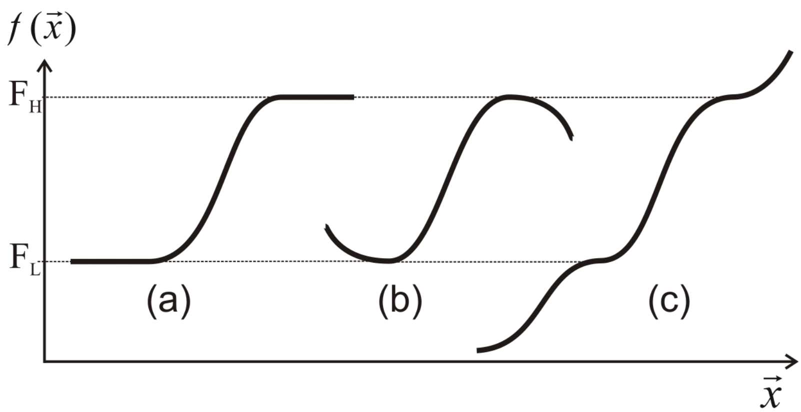


Figure 3.4: *Three types of intensity profiles across the boundary. Boundary ends by (a) large constant areas, (b) local extrema or (c) inflex points.*

The first and second order derivatives needed in the process described above were pre-computed by convolutions with Gaussian derivative kernels (see, e.g., ter Haar Romeny [22]). To limit the amount of blurring introduced by the convolution we used a Gaussian with $\sigma = 1$ voxel. The typical size of our kernel was 6 voxels per dimension. Values between voxels were then obtained by using trilinear interpolation.

3.2.2 Properties

In this section, we will present some properties of the LH histogram. In Figure 3.5, an artificial dataset with two concentric spheres is shown. The LH histogram is shown in Figure 3.5c. Each boundary appears as one point instead of an arch. This compact display of the boundaries allows an easier detection of boundaries. One advantage of the interactive or semi-automatic specification of the transfer function is that LH peaks are easier to select than arches since they have less overlaps.

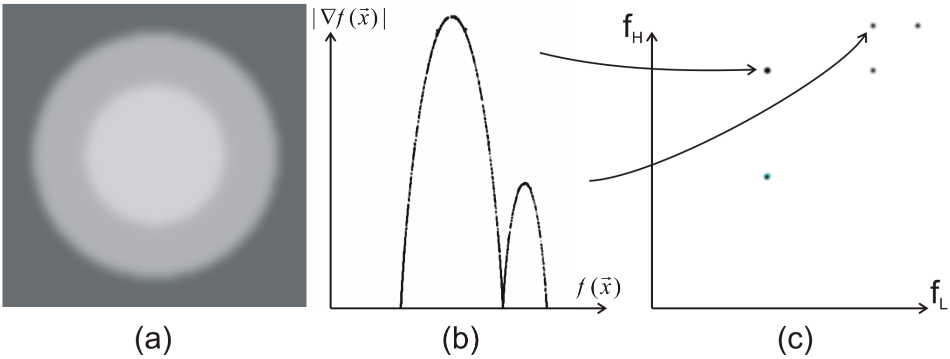


Figure 3.5: *Artificial dataset of two spheres blurred with Gaussian. See the correspondence between the boundaries in the slice (a), in the arches (b) and in the LH histogram (c). Constant areas and boundaries appear as points in the LH histogram. Constant areas of all three materials are projected onto the $f_L = f_H$ axis.*

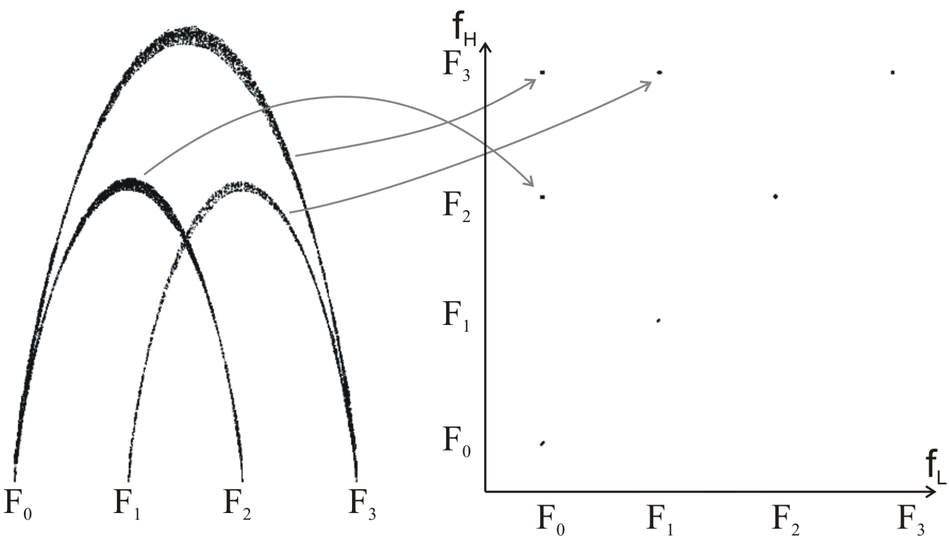


Figure 3.6: *Crossing arches appear in the LH histogram as separated points.*

Figure 3.6 illustrates how the LH histogram improves the separability of boundaries that were shown in Figure 3.1. Whereas selecting the whole arch would take a lot of effort and would be impossible in the places where arches overlap, to select one point in the LH histogram is very easy.

However, in real data there are several phenomena that might influence the compactness of both the arches and the points in the LH histogram:

1. Noise

   The model in Figure 3.5 is noise free. In order to simulate the behavior of the histogram in more realistic circumstances, we added Gaussian noise to the data. Figure 3.7 shows how the arches and the LH histogram blur. We get multiple arches for the same boundary and the single points in the LH histogram become blobs. The hue color coding shows the amount of contributions: magenta is lowest, red highest. The amount of contributions in both histograms is shown in logarithmic scale.

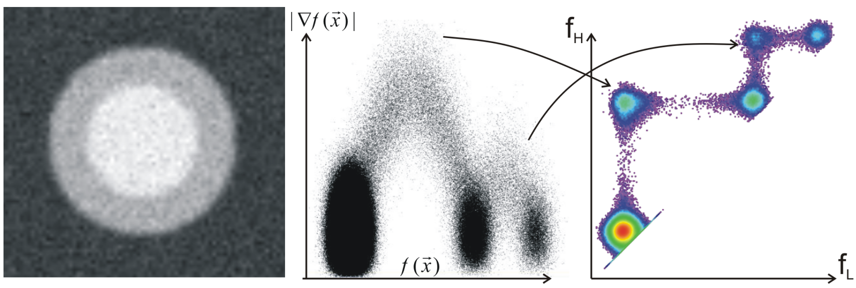


Figure 3.7: *Spheres after adding noise. The arches become blurred and project into the LH histogram as blobs rather then points. The amount of contribution is shown in logarithmic scale: magenta is lowest, red highest.*

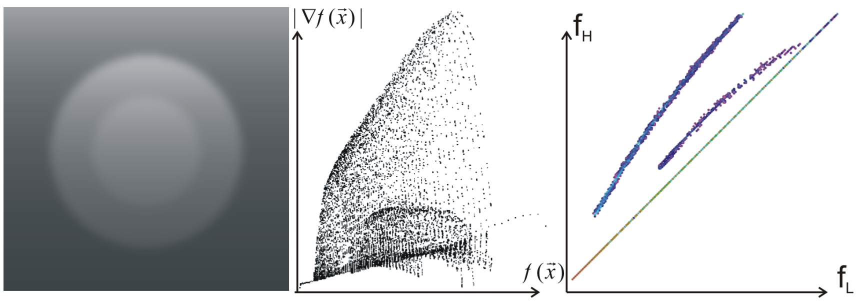


Figure 3.8: *Data after adding a rather strong bias. Notice that the information shown by the LH histogram is much more compact than that shown by the arches.*

2. Bias

   Especially in MR data we can often observe the presence of a bias field. In Figure 3.8, the data from Figure 3.5 is shown after applying a simple multiplicative bias field caused by one surface coil [45]. In the case of the arch, the bias causes multiple shifted copies of both arches which are hard to interpret. In the LH histogram boundaries appear as separated lines (instead of points) but remain relatively easy to interpret.

3. Thin objects

   For thick objects that are becoming thin we can observe that the intensity values f_L and f_H change considerably. As their thickness becomes relatively small compared to the point spread function, their intensity further resembles the background intensity. The intensity profile across the boundary of such a thin object is similar to that shown in Figure 3.4b. The result of the intensity change is either an increasing F_L or a decreasing F_H which reflects as horizontally or vertically elongated blobs in the LH histogram (Figure 3.9).

In Figure 3.10, the LH histogram is shown of the same tooth dataset as in Figure 3.2. In the LH histogram, the boundaries appear to be more compact with a considerably better separability than in the arches. In this LH histogram, we can observe two previously described effects: The boundaries appear as blobs due to the noise. The partial volume effect on thin objects causes elongation in either the horizontal or vertical direction.

3.3 Transfer functions based on the LH histogram

The F_L and F_H values could be, in principle, computed in the rendering process for any point in the volume (post-classification). However, we pre-compute them for the sake of speed (pre-classification).

We can base a 2D transfer function on the LH histogram by selecting different areas and by assigning them color and opacity. Voxels with f_L and f_H that fall inside that area are then visualized by using that color and opacity. Since we do not want to visualize all voxels that belong to the boundary, but only those lying close to the edge, we may use the gradient magnitude as the third dimension in our transfer function. The opacity of each voxel is then modulated by the gradient magnitude so that the voxels close to the edge are emphasized.

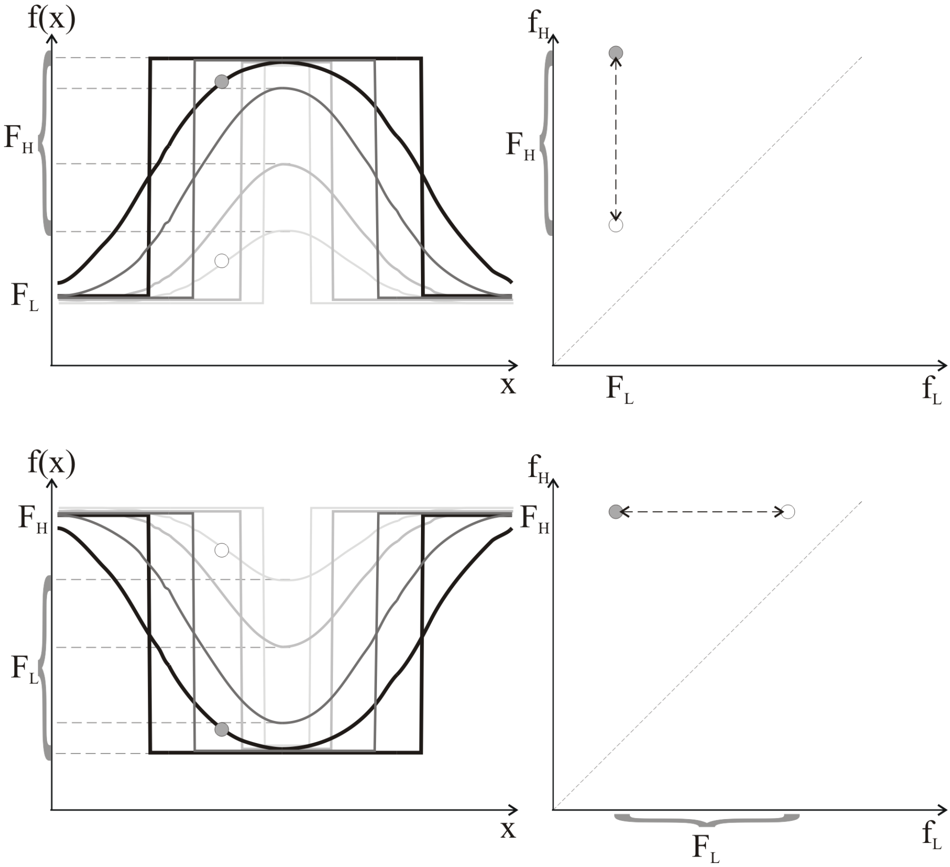


Figure 3.9: *As the objects become thinner relatively to the point spread function, the corresponding points in the LH histogram move. Top: a thin bright object, bottom: a thin dark object.*

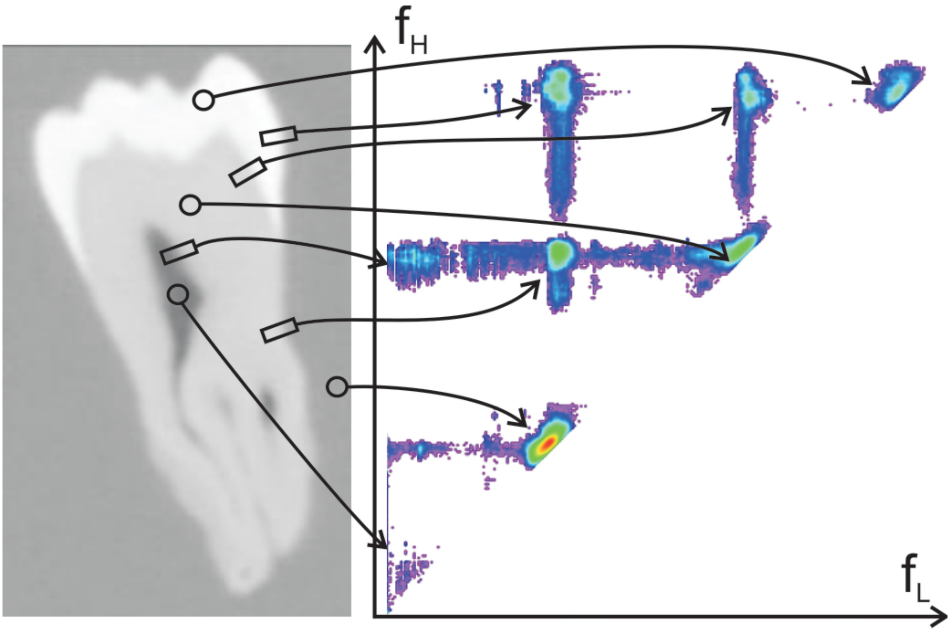


Figure 3.10: *LH histogram constructed for the tooth CT dataset from Figure 3.2.*

Figure 3.11 shows the specification of a transfer function in the corresponding LH histogram together with the resulting rendering. Both boundaries are selected by drawing polygons and assigned colors. The boundary of the outer sphere is set to be semitransparent. The illustrated simple user interface allows the user to select blobs by drawing polygonal regions. In general, more sophisticated widgets could be used in order to, e.g., allow gradual decays in opacity.

For achieving interactive rendering speed, we used the VolumePro 1000 board [46]. Since this card allows only one-dimensional transfer functions, we label the volume according to the regions selected in the LH histogram. The advantage of this approach is that we can easily combine selections in the LH histogram with those made by using the region growing (see section 3.5). The labeled volume is loaded onto the VolumePro board in addition to the original data. Two one-dimensional functions are defined for the color and opacity. The labels are used during the ray-casting for determining the color and opacity of samples. The board uses the original data for computing the gradients. The opacity is, in addition to the opacity given by the selection widget, modulated by the gradient magnitude.

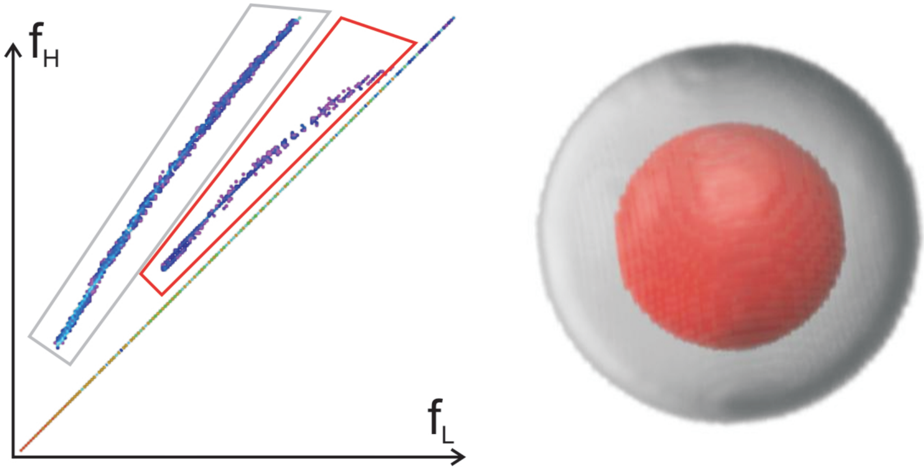


Figure 3.11: *The biased dataset of spheres. Both elongated blobs that correspond to spheres were selected and assigned different colors for rendering.*

Results

In this section, we show several visualizations by using selections in the LH histogram. As a demonstration of our methods we will show all three datasets used in the Transfer Function Bake-Off [25] (i.e. the CT scan of a tooth and the MRI scans of a knee and a sheep heart). We will also visualize a CT dataset of a hand and an MRI data of a human head.

For tracking the f_L and f_H values we used the second order Runge-Kutta method with an integration step of one voxel. Although such a step size might seem to be large, experiments with the datasets used in this chapter showed that choosing a smaller integration step does not add any observable improvements to the quality of the LH histogram.

In order to make selections in the LH histogram, there is a simple user interface to define polygonal regions. The user can then assign color and opacity to each region. In order to make the orientation in the LH histogram easier, it is possible to click at a boundary in a data slice. The corresponding position of that point is then shown in the LH histogram. This interface has only been used to prove the concept of LH based TFs and could be of course improved.

Figure 3.12 shows a visualization of a CT dataset of a hand via a transfer function that was based on a selection in the LH histogram. It is important to note that the LH histogram shows contributions of all voxels. Due to the logarithmic scale of the amount of contributions, there is a difference of several orders of magnitude between colors. The magenta areas correspond to only

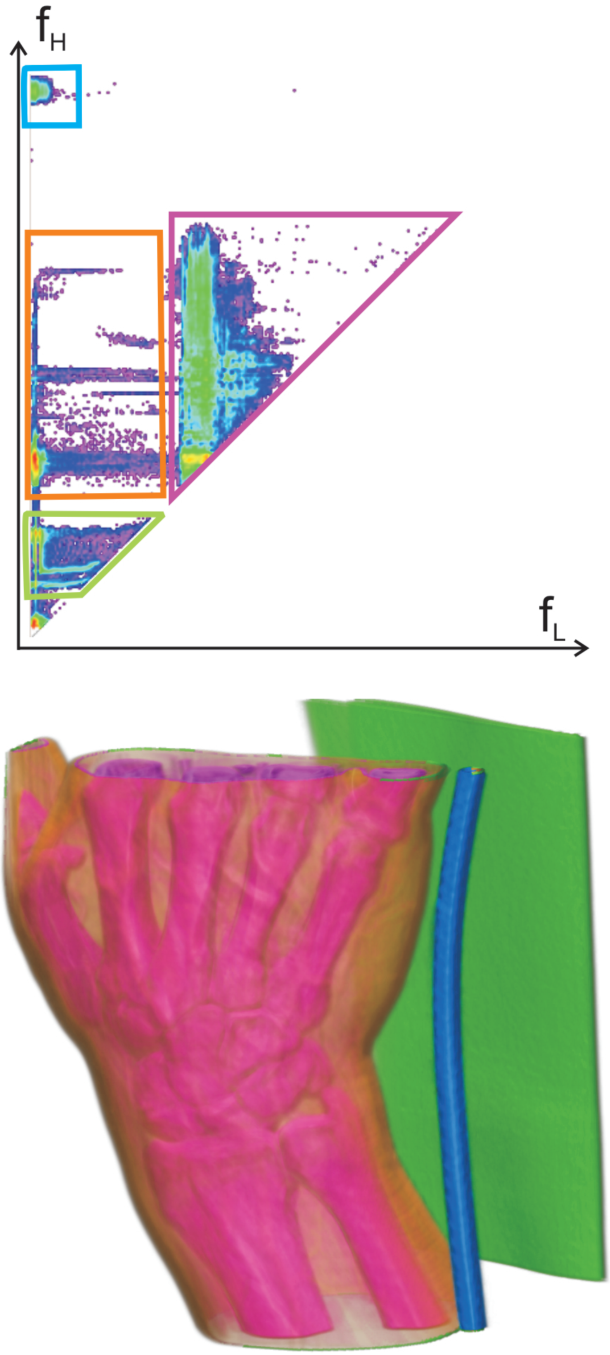


Figure 3.12: *Volume rendering of a CT scan of a hand (256x256x232) using a TF based on the LH histogram.*

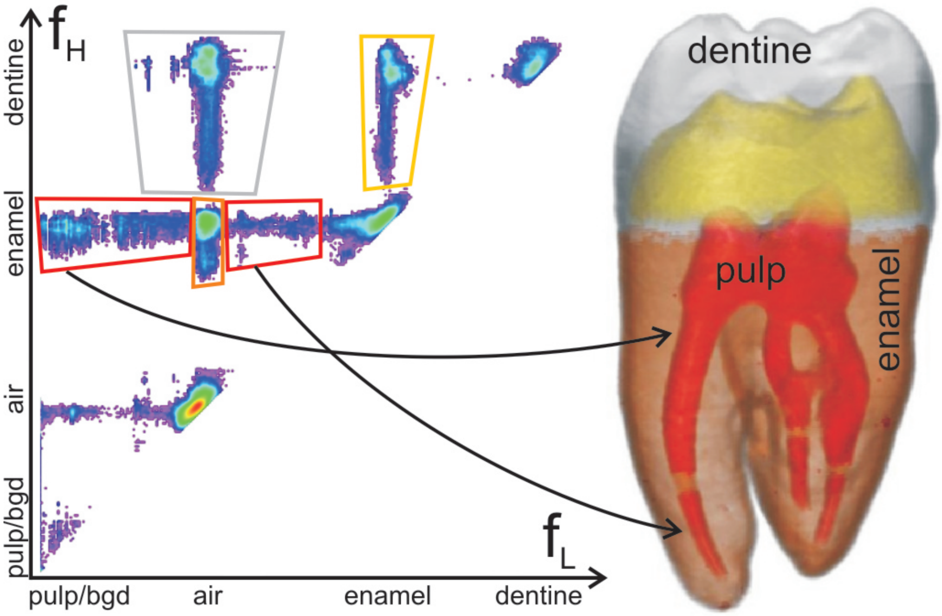


Figure 3.13: *CT of a tooth (256x256x161). The dentin-air (ochre) and enamel-air (white) boundaries are set to be semi-transparent to reveal the inner boundaries. Note that part of the pulp-dentine boundary (red) is identical with the ochre boundary.*

a very small amount of voxels, therefore their selection hardly has any visible influence on the visualization. On the other hand, a correct classification of the red, yellow and green blobs is important since they contain most of the voxels. Selection of voxels that project close to the diagonal of the LH histogram does not influence the visualization since such voxels lie in constant areas or in very weak boundaries. Due to the low gradient magnitude these voxels are rendered transparently anyhow.

Searching for the F_L and F_H values and constructing the LH histogram is done in preprocessing. Finding the F_L and F_H for the example shown in Figure 3.12 took 1 minute and 36 seconds (P4, 1.7GHz). However, this has to be done only once since the values can be stored and reused next time. Moreover, there could be many optimizations done in our algorithm, which would considerably reduce this time.

In Figure 3.13, a 3D visualization of the tooth dataset used in Figures 3.2 and 3.10 is shown. Computing of the F_L and F_H took 1 minute and 12 seconds. At the pulp-dentine boundary (red) a discontinuity is visible. This discontinuity is due to the partial volume effect. As the pulp (red) is thinning, its originally low intensity value rises and so does the F_L value found at the boundary profile. At one point the F_L value reaches the air intensity. Then the boundary inevitably looks like the dentine-air (ochre) boundary, lies on the same arch, and projects into the same point in the LH histogram. Therefore, the blob in the LH histogram that corresponds to the pulp-dentine boundary is horizontally elongated and intersects the blob corresponding to the dentine-air boundary. As the pulp becomes even thinner, the F_L is again different from the intensity of air and the rest of the boundary can be selected.

Figure 3.14 shows a selection in the LH histogram and the corresponding coloring of the arches. It is important to note that in the arch domain we could not have properly selected the boundaries due to overlaps. In Figure 3.14b the selection corresponding to the lower part of the pulp is shown. In the arch domain, it is hard if not impossible to select the corresponding data, as it contains only a small number of voxels and has a substantial overlap with other arches (one can also see this problem in the tooth renderings in the Transfer Function Bake-Off [25]). On the other hand, in the LH histogram this part is much easier to select.

Figure 3.15 shows a visualization of the MRI dataset of a sheep heart. This dataset is rather noisy which results in less compact boundaries in the LH histogram (construction took 4 minutes). However, the intensities of tissues do not have substantial overlap. It is, therefore, still possible to obtain a reasonable visualization by a selection in the LH histogram (Figure 3.15b). The blue color shows boundaries between air and tissue. The yellow shows fat tissue and boundaries between fat and muscle. Finally, red shows the muscle tissue. Large overlaps in Figure 3.15d indicate that it would not have been possible to make a similar selection by using the arches.

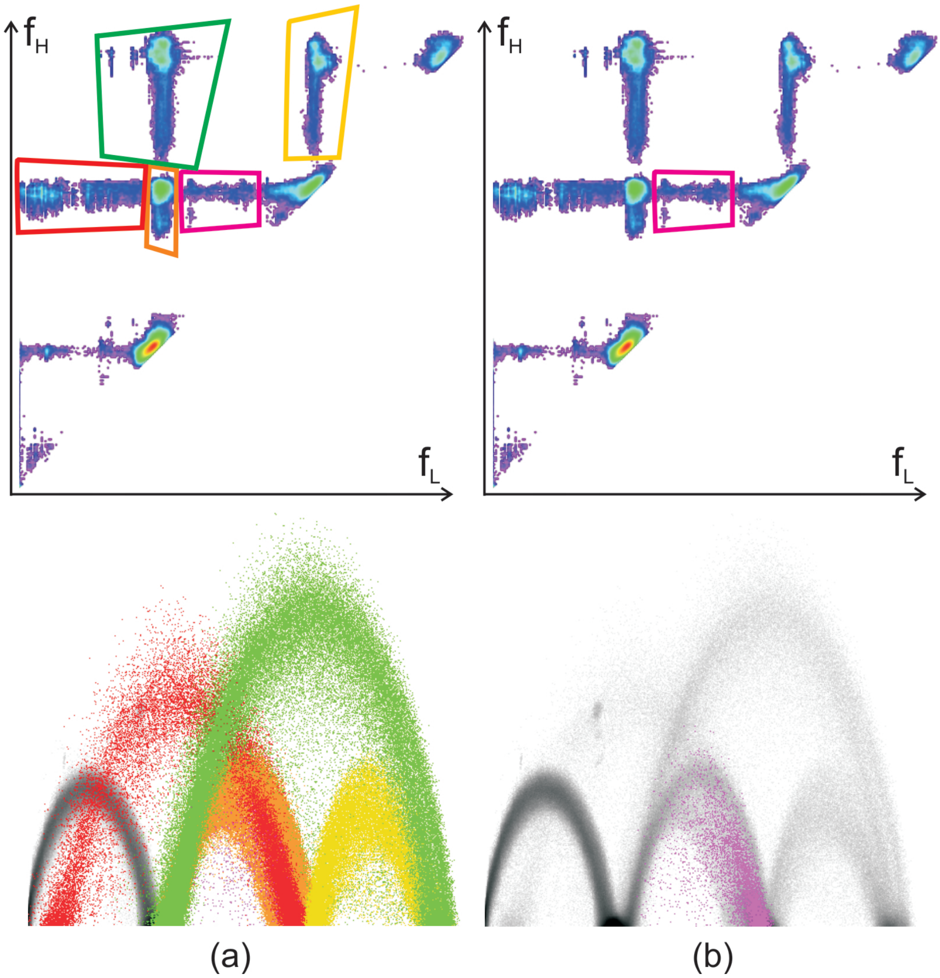


Figure 3.14: *(a) A selection in the LH histogram and corresponding coloring of the arches. In (b) only the magenta selection is colored.*

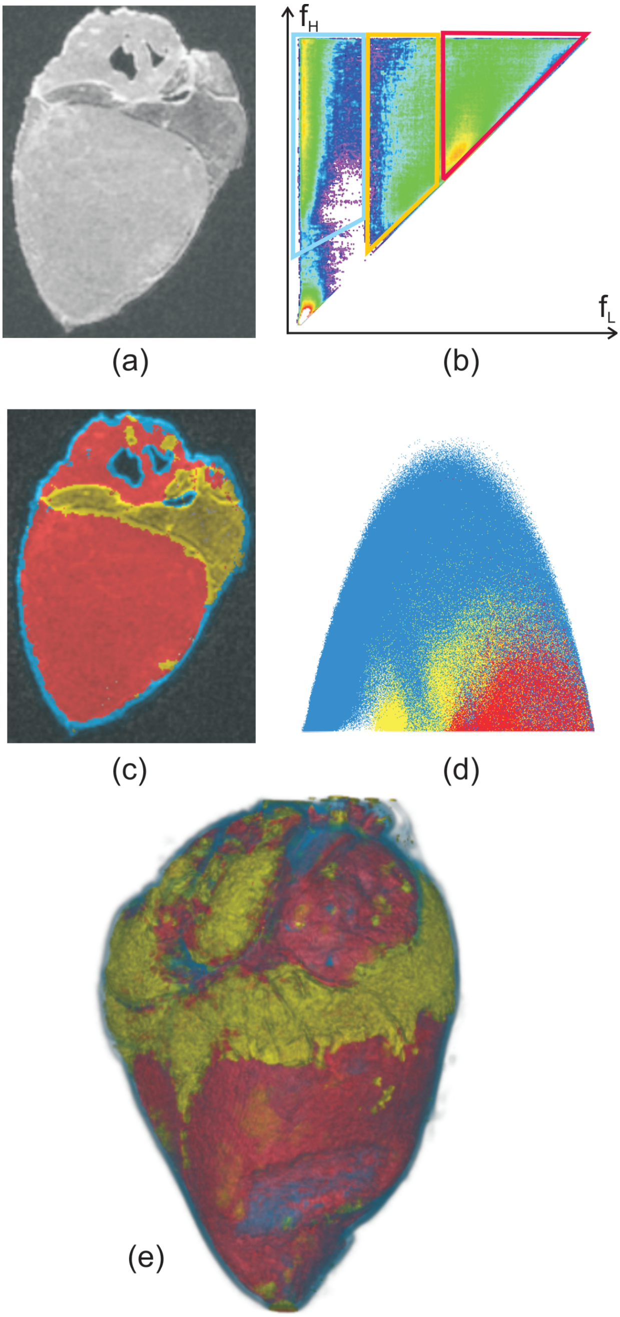


Figure 3.15: *MRI of a sheep heart (352x352x256). (a) An original slice, (b) shows the LH histogram with three selections, (c) is the slice from image (a) colored according to the selections made in the LH histogram, (d) is the corresponding coloring of arches, (e) 3D rendering.*

3.4 Mirrored LH histograms

Current TF domains enable the selection of a boundary and the assignment of certain color and opacity values. Every boundary between two materials, however, belongs partially to both materials. If the user wants to visualize both materials by assigning different optical properties to their boundaries, the classification of the common boundary will be ambiguous.

This section presents a TF domain in which the boundaries are divided into two parts by the edge location. This domain extends the possibilities of the LH space by enabling the selection of divided boundaries. In the LH space the divided boundaries are positioned in such a way that their relation is intuitive and the user can easily select all boundaries of a certain material. It is further shown that by a simple projection of the extended LH space one can obtain a new 1D TF domain that further facilitates selection of materials including their boundaries.

3.4.1 Division of the boundary

The standard LH space is designed to classify the whole boundary at once. Every boundary does, however, exist between two materials and is therefore shared by both of them. Figure 3.16 illustrates the problem that originates in shared boundaries. If we want to visualize two neighboring materials, we need to decide how to classify their common boundary. Figure 3.16b shows the two critical boundaries between objects 1-4 and 4-7 (labeled by ovals). In this example the boundaries were assigned to the object 4. As the opacity of this object decreases, so does the opacity of the common boundaries. Therefore the objects 1 and 7 appear to have holes since part of their boundary is missing (Figure 3.16d).

In order to differentiate between both sides of the boundary, we divide the boundary profile into two parts by the edge. The edge location is determined during the boundary profile tracking by looking at the zero crossing of the second derivative in the gradient direction. There are, however, more advanced methods for locating edges, especially on highly curved surfaces [47, 48].

Voxels lying above the edge, i. e., voxels with intensity $f(x) > F_E$, where F_E is the intensity at the edge location X_E (see Figure 3.17), project above the diagonal (same as in the standard LH space). Voxels lying below the edge, i.e., voxels with $f(x) < F_E$, project below the diagonal. This is achieved by swapping their F_L and F_H so that $F_H \leq F_L$. Since there is approximately the same number of voxels on both sides of the boundaries, the lower part of the extended LH histogram seems to be a mirrored version of the upper part.

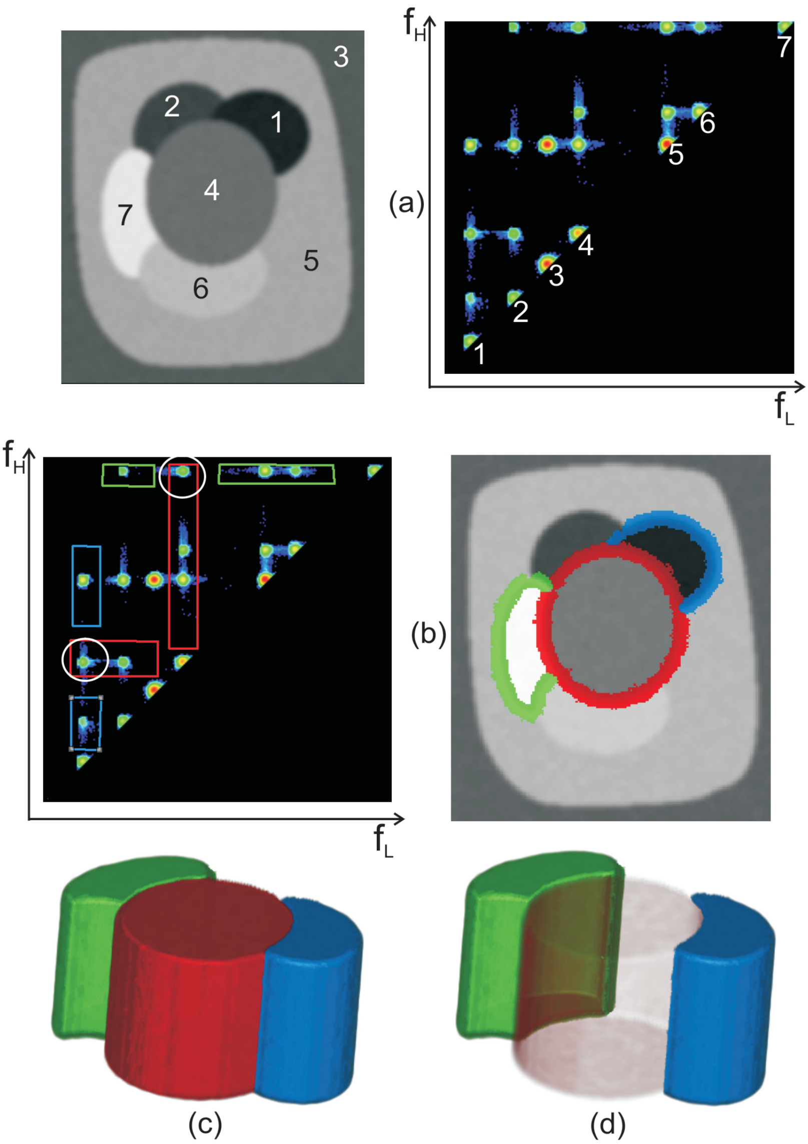


Figure 3.16: *A phantom dataset containing seven materials. (a) A slice of the dataset and the standard LH histograms with labeled interiors of materials. The constant areas project as blobs onto the diagonal. The other blobs correspond to boundaries. (b) Boundaries of materials 1, 4 and 7 are selected. The classification of the two boundaries (1-4 and 4-7) labeled by an oval is ambiguous. (c) and (d) show rendering with full and partial opacity of the cylindrical object 4.*

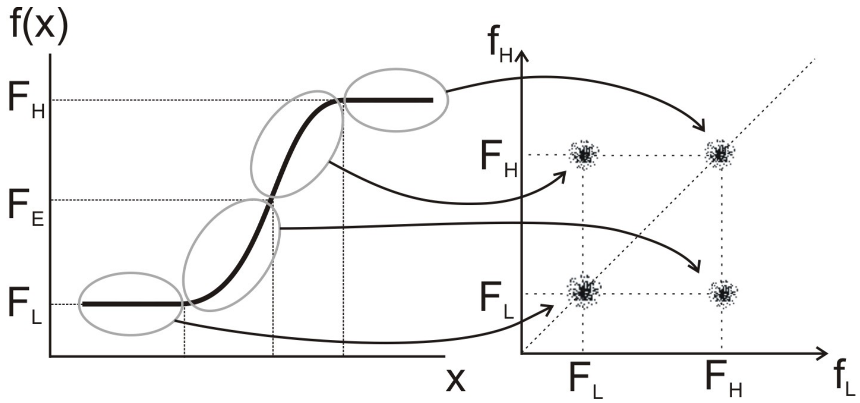


Figure 3.17: *The voxels lying above the edge intensity project into the upper section with coordinates [F_L;F_H]. The voxels with lower intensity project below the diagonal at [F_H;F_L].*

3.4.2 Properties

The phantom dataset in Figure 3.18 illustrates the possibilities of the mirrored LH space. Both parts of the split boundary can be selected independently. This allows to select only the part of the boundary that belongs to the object and resolve the ambiguous selections. Especially interesting is the fact that all partial boundaries belonging to the same material are horizontally aligned in the mirrored LH space. This allows us to easily select all relevant boundaries (see Figure 3.18b).

The boundary division allows for visualization of boundary voxels lying on one side of the edge. Therefore, assuming the edge is an appropriate estimate of objects' borders, our new approach may more accurately depict the true size and shape of the objects.

3.4.3 Horizontal projection

As shown above, the constant regions inside the material and the boundaries of the same material are horizontally aligned in LH space. Therefore one could, in principle, select all voxels belonging to a certain material knowing only F_H. By a horizontal projection onto the F_H axis, we can discard the F_L values and generate a 1D histogram. In the new projected histogram one can still select all boundaries belonging to one material although the second material that forms the boundary cannot be identified anymore. As shown in Figure 3.19 the user could then make a simple selection in the 1D histogram, which is equivalent to a horizontal selection in the mirrored LH histogram.

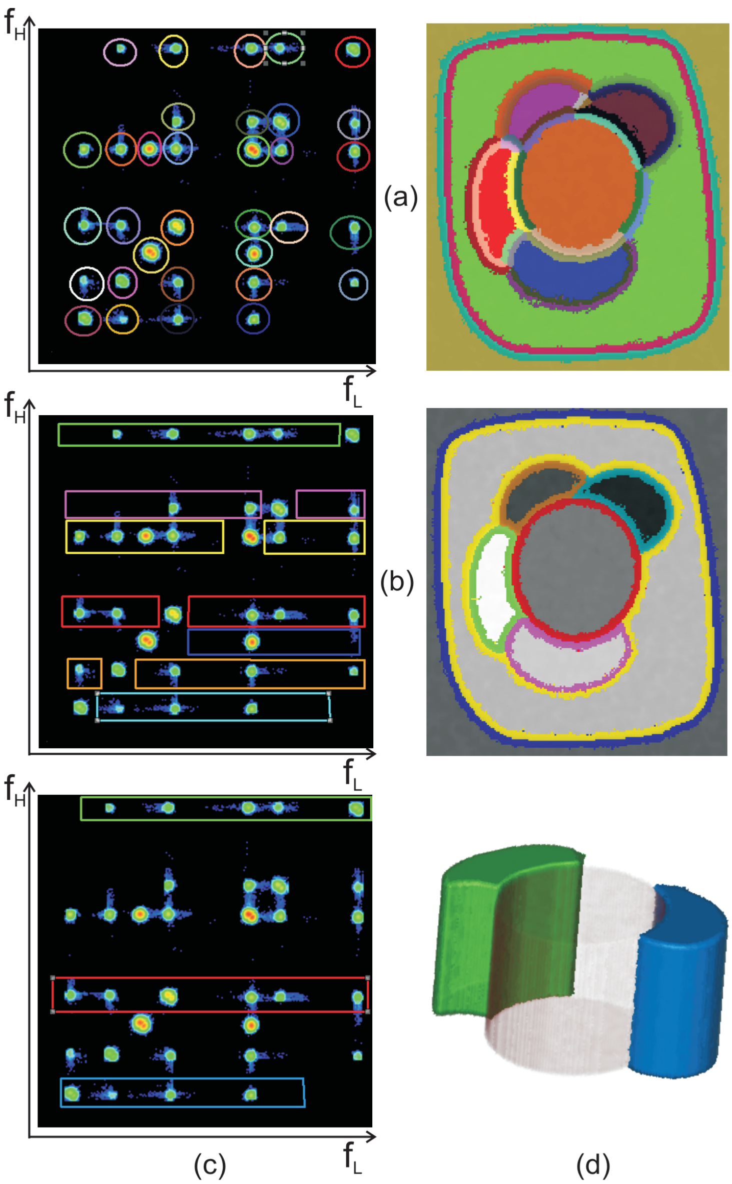


Figure 3.18: *The possibilities of the mirrored LH space illustrated on a phantom. (a) Partial boundaries can be selected independently. (b) All (parts of) boundaries that belong to one material can be easily selected, since they lie on a horizontal line. For the sake of clarity the interiors have not been selected. (c) Improving the selection from Figure 3.16. Three horizontal selections contain everything that belongs to each of the materials. (d) The rendering with the cylindrical object set to semitransparent.*

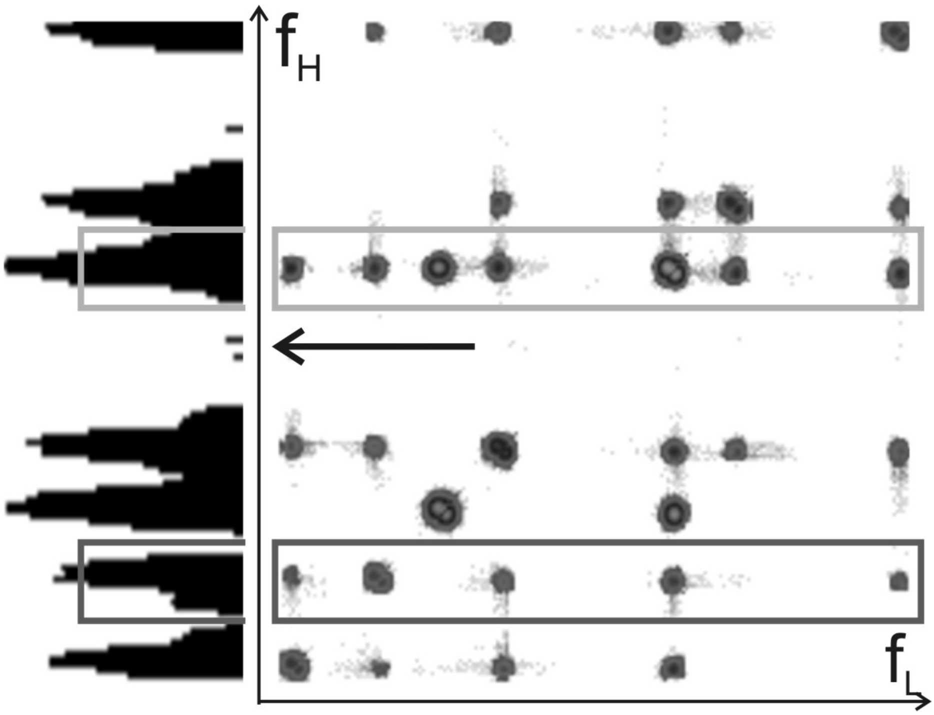


Figure 3.19: *Discarding f_L information by horizontally projecting the LH histogram. The selections in both histograms are equivalent.*

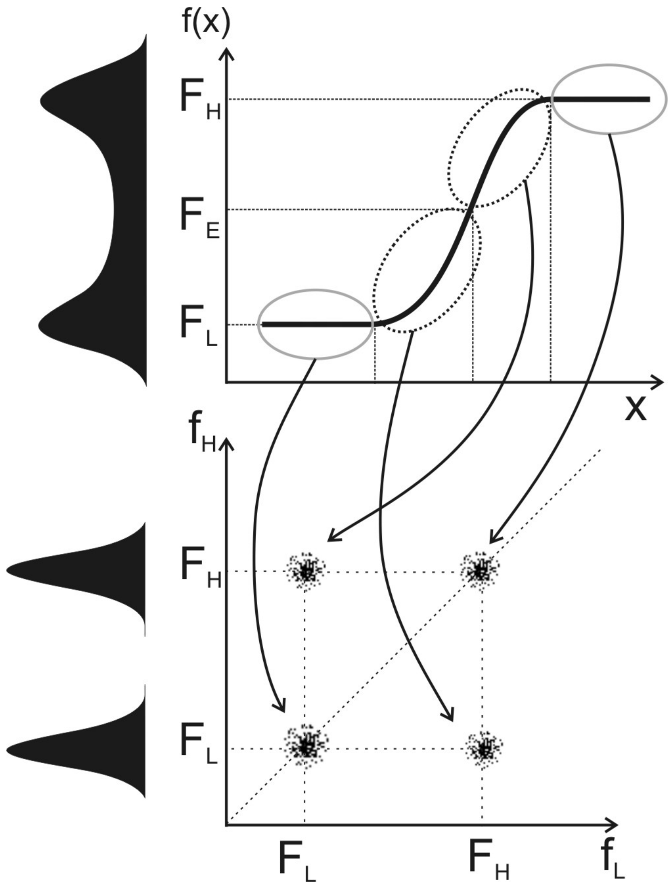


Figure 3.20: *Resolving the partial volume voxels on the boundary. In the top the boundary contains intensities in the range between F_L and F_H. These partial volume values (labeled by the dotted ovals) weaken the contrast of the material peaks in the histogram. In the mirrored LH space they are, however, mapped to F_L or F_H and therefore in the projected histogram emphasize the pure material peaks.*

The clarity of the peaks in standard 1D histograms is often hampered by the partial volume voxels on the boundaries. The mirrored and projected histogram, however, resolves these boundary voxels. As a result, the peaks of the histogram become more clear. Figure 3.20 illustrates the mechanism of resolving the partial volume voxels on the material boundaries. Figure 3.21 shows the comparison of both histograms for the engine dataset.

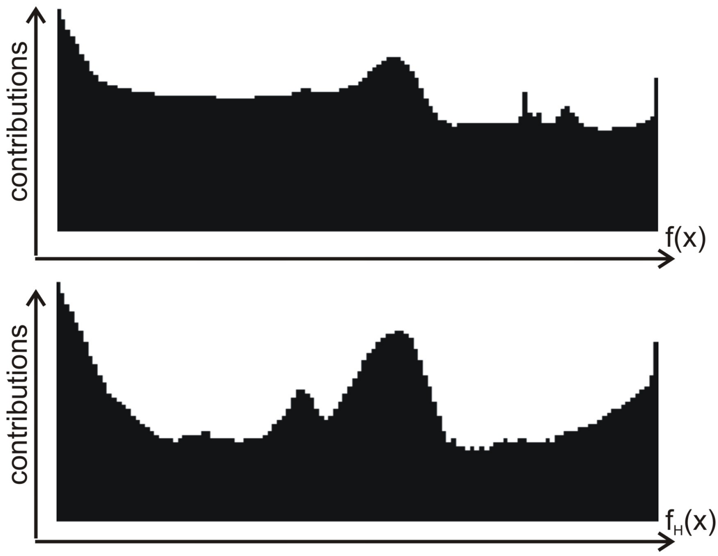


Figure 3.21: *The engine dataset. A comparison of the original histogram of intensities and the projected histogram containing F_H values. By resolving the partial volume voxels on the boundaries, the peaks become more clear. The histogram contributions are shown in logarithmic scale.*

3.4.4 Results

We will demonstrate the use of the mirrored LH space on several datasets. Figure 3.22 shows a rendering of the engine dataset. On the left a selection in the standard LH space is used. The hard material-air and the hard-soft material boundaries are shown as part of the hard material components. On the right, however, the boundaries are divided and only the part of the boundary that belongs to the material is selected. It is clearly visible that the hard components appear to be wider in the left rendering, which might be misleading. In the right image only voxels that lie within the object boundaries (detected by the edge location) contribute to the rendering. Therefore the selection of boundaries in the mirrored LH space has the potential to better visualize the size and shape of objects.

Figure 3.23 shows the selection and rendering of three materials including their boundaries. When the soft material is set as semi-transparent one can see that the boundaries consist of two parts.

The tooth dataset in Figure 3.24 is an example where a combined boundary selection can be used. Two tissues are selected using the horizontal selection technique. The third tissue is selected by restricting the F_L values in order to avoid the selection of boundaries with the background.

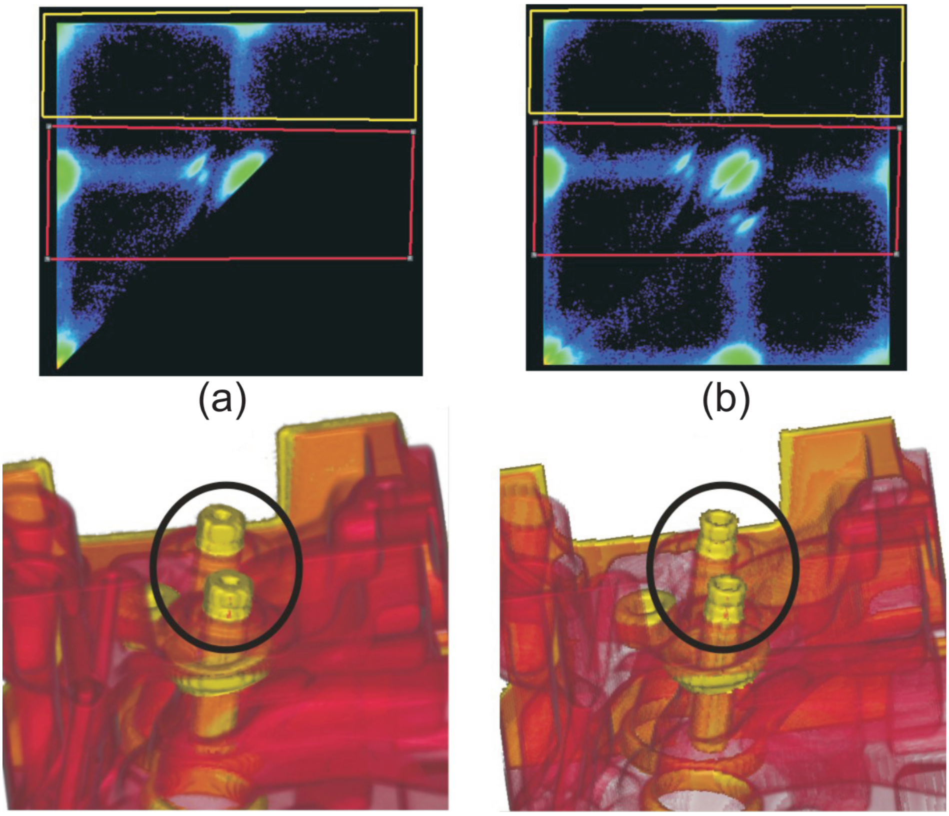


Figure 3.22: *The engine dataset. (a) A TF based on the standard LH histogram. (b) Selection of all voxels belonging to the same material in the mirrored LH histogram. In (a) the objects appear thicker due to the contributions of voxels on both sides of the edge. This effect is well visible on the hard material components.*

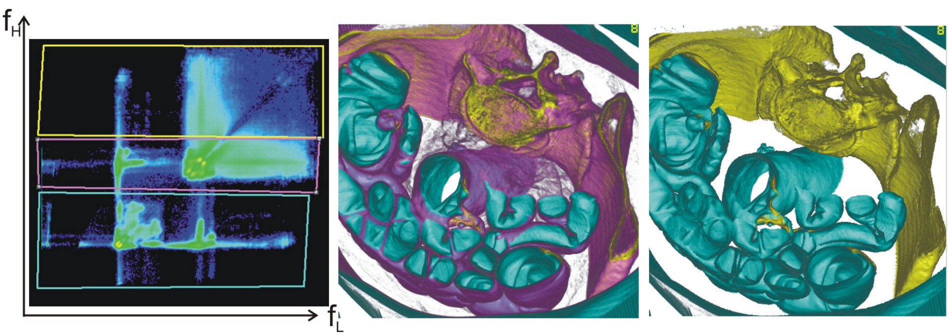


Figure 3.23: *Abdominal CT scan. Three materials are highlighted: the air in the colon, the bone and the soft tissues. Notice the effect of both sides of the boundaries being classified separately. The right rendering shows only the colon wall and the pelvis bone.*

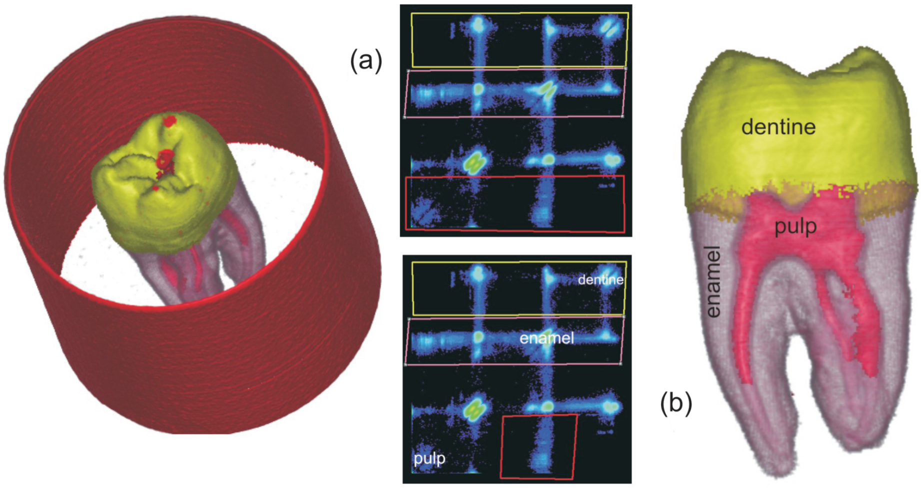


Figure 3.24: *The CT tooth dataset. The dentin and enamel boundaries are selected by using the horizontal selection of all relevant boundaries. In (a) the pulp was selected by using the same technique. Since the pulp and background have similar intensities, in (b) the selection is restricted only to the pulp-enamel boundary.*

Use of the mirrored LH space does not involve any additional performance or memory requirements as compared to the standard LH space. The only difference is that our tracking algorithm differentiates voxels lying above and below the edge. For those lying below the edge it returns F_L instead of F_H and vice versa. The rest of the visualization pipeline remains unchanged.

3.5 Region growing using LH histogram and boundary information

In most types of transfer functions the spatial information is not taken into account. However, we can often find situations in which different objects separated in space have similar data properties and therefore the transfer function cannot distinguish between them. One may hope that the introduction of more dimensions into the transfer function domain could possibly help to separate such objects. However, often this is not the case and it may then be appropriate to use a spatial segmentation such as the region growing method.

Huang and Ma [38] presented a region growing method that was aimed at facilitating a good visualization. They based their cost function on the intensity $f(\vec{x})$ and gradient magnitude $|\nabla f(\vec{x})|$ and reported that this helped to solve those cases where a simple selection in the arches was impossible due to a significant overlap.

Below we explain some basic notions of region growing and propose our method of region growing. We show that a cost function based on the LH histogram has important advantages.

In region growing, a voxel of the object to be labelled is chosen by the user as a seed. From this seed voxel the region is grown by repetitively labelling neighbors of already labelled voxels. At each step a neighbor is selected that is most similar to already labelled voxels. To measure the similarity a cost function is used. The cost function evaluates for each voxel how expensive it is in terms of dissimilarity to extend the region to that voxel.

Our region growing is based on growing the boundaries, based on the assumption that it is undesirable to grow outside the current boundary, into another boundary, into areas of constant intensities or into small noisy boundaries. Therefore, the presented cost function is based on the similarity of boundaries and their properties.

3.5.1 Similarity Measure

We introduce a set of criteria that measure the cost of growing from a voxel to its neighbors. We evaluate the similarity of the voxel and its neighbor as well as the similarity of the neighbor to the grown object (i.e. voxels already grown).

The criteria are the following:

1. *Object distance* reflects similarity to all voxels already labelled. We use Euclidian distance in the LH space to measure the difference between two boundaries. C_{OD} is hence given by

$$C_{OD} = \sqrt{(f_L(\vec{x}_{i+1}) - \overline{f}_L)^2 + (f_H(\vec{x}_{i+1}) - \overline{f}_H)^2}$$

   In our notation $f_L(\vec{x}_i)$, $f_H(\vec{x}_i)$ are values of an already labelled voxel and $f_L(\vec{x}_{i+1})$, $f_H(\vec{x}_{i+1})$ values of its neighbor. Finally, $\overline{f}_L$ and $\overline{f}_H$ are average values of all the already labelled voxels.

2. *Neighbor distance* reflects similarity between neighbors. Our experience indicates that this measure is useful to allow gradual changes of the F_L, F_H values that occur, for example, in a biased boundary or in a boundary that changes due to thinning of an object. We need to accept

small changes between neighbors although the object distance might be large. C_{ND} is defined as

$$C_{ND} = \sqrt{(f_L(\vec{x}_{i+1}) - f_L(\vec{x}_i))^2 + (f_H(\vec{x}_{i+1}) - f_H(\vec{x}_i))^2}$$

3. *Neighbor coherence* measures the directional coherence of gradients. It is weighted by the average strength of the neighbors. To prevent growth into a neighboring boundary, an additional cost is assigned to the neighbor that has a different gradient direction.

   The direction is measured at the strongest gradient magnitude along the boundary profile.

$$C_{NC} = \frac{(f_H(\vec{x}_{i+1}) - f_L(\vec{x}_{i+1})) + (f_H(\vec{x}_i) - f_L(\vec{x}_i))}{2} * a$$

   where a is a value between 0 and 1 that is proportional to the angle α between the directions.

$$a = \frac{1 - cos(\alpha)}{2}$$

4. *Boundary strength* emphasizes the importance of growing of strong boundaries (i.e. voxels where $f_H(\vec{x})$ and $f_L(\vec{x})$ have a large difference).

$$C_{BS} = \frac{max\{f_H(\vec{x}) - f_L(\vec{x})\}}{1 + (f_H(\vec{x}_{i+1}) - f_L(\vec{x}_{i+1}))}$$

   Where $max\{f_H(\vec{x}) - f_L(\vec{x})\}$ representing the strongest boundary in the volume is used for normalization.

The costs stated above are weighted by constants k_{OD}, k_{ND}, k_{NC}, k_{BS}. Since each criterion is meant to help in different situations, we need to adapt the weight constants according to the problems we experience in the data. For example, for selecting boundaries where F_L and/or F_H gradually change, we need to increase k_{ND} and decrease k_{OD} in order to allow these changes. Also, using higher k_{NC} is beneficial in cases where boundaries are strong. On the other hand, for weak boundaries that are affected by noise we rather need to use high k_{OD} and low k_{NC} since the direction of such boundaries is less reliable.

The resulting cost can be then evaluated as:

$$C = k_{OD} C_{OD} + k_{ND} C_{ND} + k_{NC} C_{NC} + k_{BS} C_{BS}$$

The main advantage of our approach compared to the one based on the intensity $f(\vec{x})$ and gradient magnitude $|\nabla f(\vec{x})|$ is that the combination of $f(\vec{x})$ and $|\nabla f(\vec{x})|$ can exist in more boundaries because the arches often overlap (e.g. see Figure 3.2). Such ambiguities may easily mislead the region-growing process. Using costs based on the LH histogram will more likely resolve such situations. An important feature of our cost function is that the region growing selects all voxels that belong to the boundary profile and not only voxels close to the edge.

A common visualization task would be showing not only one boundary, but the whole object. In the terms of boundaries, it would mean showing all boundaries between that object and surrounding objects. This could be done, in principle, by selecting all relevant boundaries either in the standard or in the mirrored LH histogram. However, the region growing as presented here is rather suited for selections of single boundaries in the standard LH histogram. If the cost coefficient k_{ND} does not allow to grow more different boundaries, the user needs to select them one by one.

3.5.2 Results

The pulp-dentine boundary illustrates a typical problem of transfer functions. The two boundaries could not be separated as they overlap in the transfer function domain. Figure 3.25 shows the result of segmenting the pulp-dentine boundary by using region growing. The boundary could be grown because gradual changes of intensities were allowed. In this case also the growth of a coherent boundary was important because the dentine-air boundary appears very similar to the pulp-dentine and they get very close to each other. Growing of the boundary took 3 seconds.

In Figure 3.26, an MRI dataset of a human head is shown. Due to the noise and partially overlapping intensities of tissues it was not possible in the LH histogram to select the skin and brain separately. Therefore we used a combination of a selection in the LH histogram and region growing. After selecting the air-tissue boundary in the histogram, the brain was grown (in 34 seconds). It is important to note that we were not able to obtain an equally good selection by using a cost function consisting of only intensity and gradient magnitude.

Finally, we visualized the MRI dataset of a knee (Figure 3.27). The size of the original version was 512x512x87. For the visualization we used a sub-sampled version of size 256x256x87. For this data it was not possible to obtain any meaningful visualization by using selections in the LH histogram. Therefore we used region growing. Four different parts of the knee were grown. Growing of each part took between 5 and 10 seconds.

Figure 3.25: *The pulp boundary (red) was selected by region growing (right). Note that the ambiguous part of the boundary (left) is now classified correctly. The cost coefficients were* $k_{OD} = 2$, $k_{ND} = 1$, $k_{NC} = 2$ *and* $k_{BS} = 0$.

3.6 Summary and Conclusions

We showed that the LH histogram and its mirrored extension allow easier identification and selection of boundaries than we could obtain from the arches in the domain of intensity and gradient magnitude. We have shown cases where because of a large overlap (due to disturbing phenomena such as noise and partial volume effect) one is not able to select a boundary in the arches while this is possible in the LH histogram. The compactness of the boundaries in the LH histogram may become even more relevant if we are to select the boundaries automatically. However, the LH histogram requires more complex computations.

We presented a region growing technique that uses information of the boundary in each voxel. The cost function we proposed allows the growing of boundaries distorted by several types of phenomena. Approaches that use only intensity and gradient magnitude can be easily misleading in cases where the combination of $f(\vec{x})$ and $|\nabla f|$ belongs to more boundaries. Our approach has the potential of resolving such ambiguities.

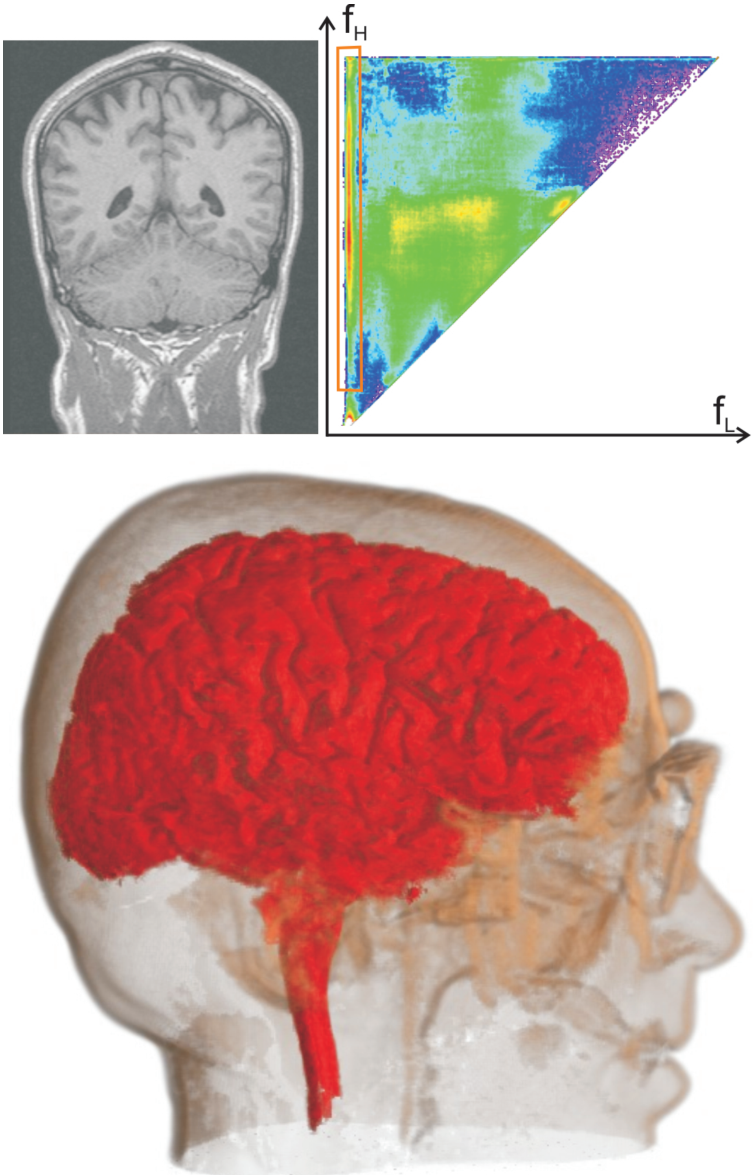


Figure 3.26: *MRI scan of a head (187x236x253). The air-tissue boundary was selected in the LH histogram. The brain was selected by region growing which took 34 seconds. The cost coefficients were set to* $k_{OD} = 1$, $k_{ND} = 2$, $k_{NC} = 2$ *and* $k_{BS} = 3$.

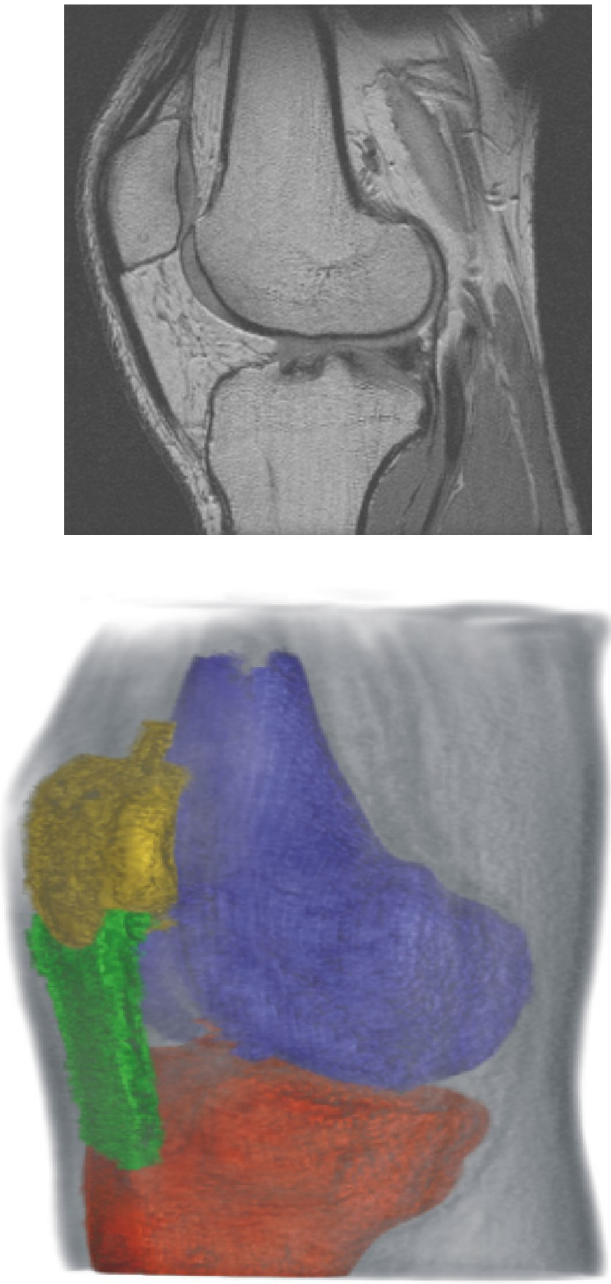


Figure 3.27: *MRI scan of a knee (256x256x87). The colored segments were obtained by using region growing. The cost coefficients were* $k_{OD} = 1$, $k_{ND} = 2$, $k_{NC} = 4$ *and* $k_{BS} = 1$.

Moreover, in conventional techniques the user is often asked to enter the range of possible intensities and gradient magnitudes that can be grown. In our approach the boundary is specified by only two values – the intensities of materials that form the boundary. This approach is much more intuitive. The presented methods could be further extended by defining the cost function in the mirrored LH histogram which would enable an independent selection of either side of the boundary.

A common drawback of region growing methods is that they are often too slow to support fluent user interaction. Similarly, our method would need to be simplified and optimized to allow interactivity.

The model used here only considers boundaries of two materials. However there are often places where three materials meet. Such three-material boundaries are recognized by our method as boundaries of materials of the highest and lowest value. Serlie et al. [43] used a three-material boundary model to classify materials for colon cleansing. This method would have to be extended to include boundaries of any three materials.

An important issue in the presented methods is the robustness for estimating the F_L and F_H values. A more robust estimation could possibly improve compactness of the LH histogram, especially in the presence of noise. A faster estimation of F_L and F_H would allow a post-classification rendering.

To conclude, the LH histogram is a promising domain that has the potential of facilitating the visualization of volume data. The next chapter shows how the LH histogram can be used to automate the TF design.

Chapter 4

Automating TF Design Using Hierarchical Clustering

4.1 Introduction

TF design is an important issue in volume rendering. Medical workstations usually tackle the problem of intuitive TF design by having several preset transfer functions. The user, similarly to the approach of the design galleries [29], picks the best preview. This approach is, however, not flexible. If there is no appropriate preset, the TF has to be again tuned manually. There have been several approaches that automate and facilitate the TF design using clustering techniques [38, 36, 15]. Interaction with the clusters seems to be more intuitive than a direct interaction with the TF.

The overall performance of the clustering techniques is highly dependent on the separability and compactness of the clusters in feature space. For the visualization of material boundaries, the space of the intensity and the gradient magnitude is commonly used. However, in that space the boundaries appear as arches that frequently overlap causing classification ambiguities. Furthermore, it is rather difficult to take into account the shape of such clusters. For clustering, it is a great advantage if the clusters have a simpler shape. It has been shown in the previous chapter that in the so-called LH space the boundaries between materials have substantially less overlap compared to the space of arches. In addition, boundaries appear as blobs instead of arches. This space shows more suitable characteristics for developing a clustering technique.

The usual drawback of clustering techniques is their sensitivity to clustering criteria or parameters. Tuning of such criteria is often a difficult task. More-

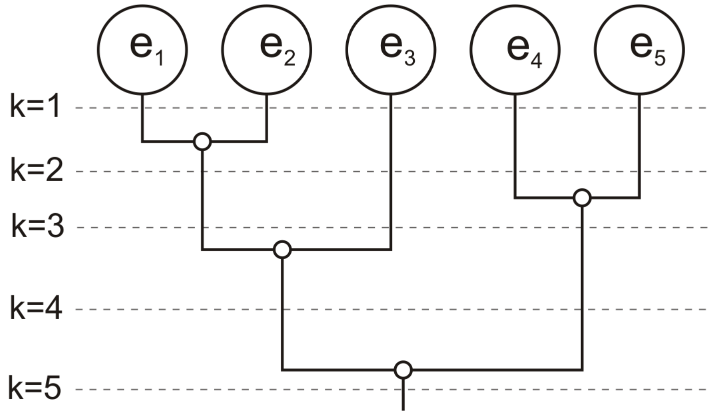


Figure 4.1: *Example of hierarchical clustering of five initial clusters visualized using a dendrogram.*

over, in order to combine several criteria, one needs to choose proper weighting which is even more difficult. In this chapter, we show how hierarchical clustering methods can facilitate the design of transfer functions. Our first contribution is a framework in which the user interacts with a hierarchy of clusters. No predefined clustering threshold is needed. The user interacts with the hierarchy. At any time the user can pick another similarity measure (clustering criterion) and the hierarchy is automatically adjusted. Our framework enables an easy combination of similarity measures. We define the clusters in the LH space that serves as the TF domain. The second contribution of this chapter are two intuitive similarity measures for the clusters. One evaluates the similarity of the boundaries in the LH space and the second evaluates their spatial relation in the volume. We include both similarity measures into the framework.

4.2 Hierarchical clustering

Having an initial set of n elements $e_1, ..., e_n$, agglomerative hierarchical clustering [49] describes the order in which the elements join into clusters. In the initialization step each element e_i is in its own cluster C_i. The hierarchy has n levels $k = 1, ..., n$. In level $k = 1$ there are n clusters, i.e., no elements are joined. In every following level the two most similar clusters join. The maximal similarity s at level k between clusters is found as

$$s_k = max\{s(C_i, Cj)\}; i, j = 1, ..., n - k + 1; i \neq j$$

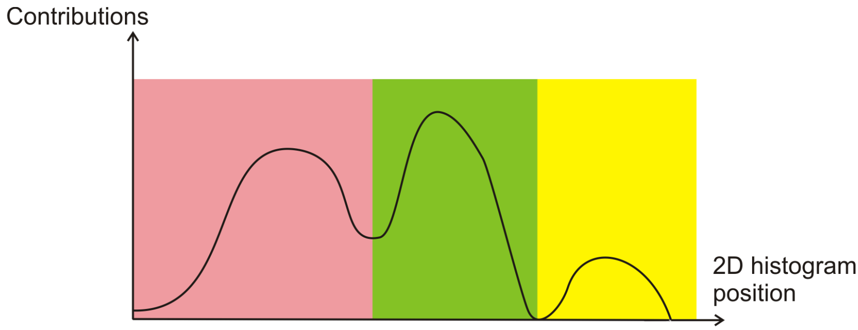


Figure 4.2: *A 1D illustration of the initial labeling of the 2D peaks in the LH histogram. Every peak becomes one initial cluster.*

Finally, at level $k = n$ only one cluster remains. The resulting hierarchy can be intuitively represented by a binary tree (supposing no two pairs of clusters have the same similarity) called dendrogram. Figure 4.1 shows an example of a dendrogram.

In the following text we describe our approach to find the initial set of elements $e_1, \ldots, e_n$. Further we describe two different similarity measures developed in order to evaluate the similarity between clusters.

4.3 Similarity measures

We have developed two similarity measures that intuitively evaluate the correspondence between clusters in the LH space. The first measure is designed to group similar boundaries, i.e., boundaries that lie close to each other in the LH space (see Chapter 3). The second measure evaluates the spatial connectivity of clusters. We first describe how the initial elements e_i are obtained and then how the similarity measures are evaluated.

4.3.1 Initial clustering

It has been shown in the previous chapter that the boundaries between tissues form blob-like clusters in the LH space. We use this property to define meaningful initial clustering elements e_i. In the LH histogram, every boundary can be localized as a local peak. Therefore, we label every local maximum of the 2D histogram, see Figure 4.2 for an 1D illustration. All histogram bins that belong to the same peak then form one of the initial clustering elements, e_i.

Having too many initial clusters is not beneficial for the performance of the clustering technique. The complexity of generating the hierarchy is $O(n^3)$ where n is the number of initial clustering elements e_i. Depending on the dataset and the resolution of the histogram (in this thesis we used 256^2 bins), in some cases there might be several thousands of local peaks. Large numbers of them would typically represent only very small initial clusters consisting of a few voxels. From our observation, reducing the number of initial clusters to several hundreds, allows an interactive generation of the hierarchy without much effect on the results.

In order to get rid of very small clusters that contain only few voxels, we perform the following two steps. Firstly, before the local peak detection, the histogram is blurred using a 2D Gaussian kernel of a small σ equal to the bin size. This both removes small noisy local maxima and establishes a direct neighborhood relation between peaks that were separated. Secondly, after the initial clusters have been generated, those consisting of only few voxels (in our method less than 10, note that this is not a critical threshold, other values could also have been used) are joined with their direct neighbors. We can reason that such small clusters have little effect on the visualization and therefore assigning them to their neighbors will not introduce any visible artifacts.

4.3.2 Similarity in LH space

This similarity measure is designed to enable grouping of similar boundaries. In order to achieve this, we use the positions, sizes, and shapes of initial elements e_i in the LH space.

In order to evaluate the similarity of two elements, we want to take into account the following criteria:

- *Distance.* Close elements have similar L and H values (see Figure 4.4a), i.e., they correspond to boundaries between similar tissues.
- *Separation.* A deep valley between two peaks yields a good separation. If there is little evidence of boundary profiles that project in between the two peaks, it is likely that the peaks correspond to different boundaries.
- *Direction of elongation.* Due to changes of the tissue intensities that form the boundaries, the corresponding peaks may be horizontally and/or vertically elongated. Therefore, we preferably want to join the elements in the direction indicated by their elongation.

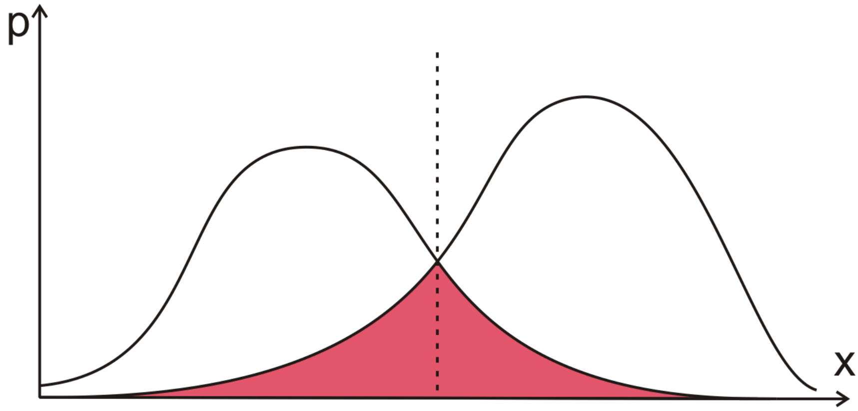


Figure 4.3: *Two probability distributions. The dashed line is the decision boundary made by a Bayesian classifier. The red area represents the probability of a wrong decision. We use an estimate of the overlap of two 2D distributions as our similarity measure.*

All these criteria can be elegantly combined by applying known techniques from Bayesian decision theory [49]. If we look at the clustering elements as bivariate (2D) probability density functions (PDF), we can define the similarity between two elements as the overlap of their PDFs. An exact computation of this overlap is not trivial. However, assuming the elements can be approximated by bivariate normal density functions, we can estimate the upper bound of the overlap by using the so-called Bhattacharyya bound. It is the upper estimate of the probability of a wrong decision made by a Bayesian classifier given the PDFs and the priors [49] (Figure 4.3).

We define the size of an element as the number of voxels that belong to the element, i.e.,

$$|e_i| = \sum_{b_{xy} \in e_i} |b_{xy}|$$

where b_{xy} is the bin at discrete histogram position (x, y) and $|b_{xy}|$ the number of contributions in the bin. Then, the priors $P(e_i)$ and $P(e_j)$ can be obtained as

$$P(e_i) = \frac{|e_i|}{|e_i| + |e_j|}; \quad P(e_j) = \frac{|e_j|}{|e_i| + |e_j|}$$

The similarity of two elements e_i and e_j is computed using the Bhattacharyya bound

$$s(e_i, e_j) = 2\sqrt{P(e_i)P(e_j)}\, e^{-k}$$

where the factor 2 scales the values into the range $< 0, 1 >$ and

$$k = \frac{1}{8}(\mu_j - \mu_i)^t \left[\frac{\Sigma_i + \Sigma_j}{2}\right]^{-1} (\mu_j - \mu_i) + \frac{1}{2} ln \frac{\left|\frac{\Sigma_i + \Sigma_j}{2}\right|}{\sqrt{|\Sigma_i||\Sigma_j|}}$$

where the mean vector μ_i and covariance matrix Σ_i describe the bivariate normal distribution of element e_i. The mean vector of element e_i is defined as

$$\mu_\mathbf{i} = (\mu_{ix}, \mu_{iy})^t = \left(\frac{1}{|e_i|} \sum_{b_{xy} \in e_i} (x|b_{xy}|), \frac{1}{|e_i|} \sum_{b_{xy} \in e_i} (y|b_{xy}|) \right)^t$$

$$\text{and the covariance matrix is} \Sigma_i = \begin{pmatrix} \sigma_{i\,xx} & \sigma_{i\,yx} \\ \sigma_{i\,xy} & \sigma_{i\,yy} \end{pmatrix}$$

In order to avoid Σ_i to be singular due to clustering elements that are only one bin wide, we split every bin into four

$$b_{xy} \rightarrow \{b_{x \pm \frac{1}{4}\, y \pm \frac{1}{4}}\}; \quad |b_{x \pm \frac{1}{4}\, y \pm \frac{1}{4}}| = \frac{|b_{xy}|}{4}$$

which is equivalent to adding small regularization terms to the diagonal elements of Σ_i, i.e.,

$$\sigma_{i\,xx} = \frac{1}{|e_i|} \sum_{b_{xy} \in e_i} \left(|b_{xy}|(x - \mu_{ix})^2\right) + \frac{1}{16}$$

$$\sigma_{i\,yy} = \frac{1}{|e_i|} \sum_{b_{xy} \in e_i} \left(|b_{xy}|(y - \mu_{iy})^2\right) + \frac{1}{16}$$

$$\sigma_{i\,xy} = \sigma_{i\,yx} = \frac{1}{|e_i|} \sum_{b_{xy} \in e_i} (|b_{xy}|(x - \mu_{ix})(y - \mu_{iy}))$$

We compute the similarity of two clusters C_i, C_j using the single-link (nearest neighbor) method, i.e.,

$$s(C_i, C_j) = max\{s(e_i, e_j)\}; \quad e_i \in C_i, e_j \in C_j$$

The joined cluster $C_{i \cup j}$ contains all elements from both clusters.

The presented similarity measure is a semi-metric, since it is positive, symmetric and reflexive but the triangle inequality does not hold. For our method these properties are sufficient. The triangle inequality is not required since we do not perform arithmetic operations on the similarities, we only compare their values.

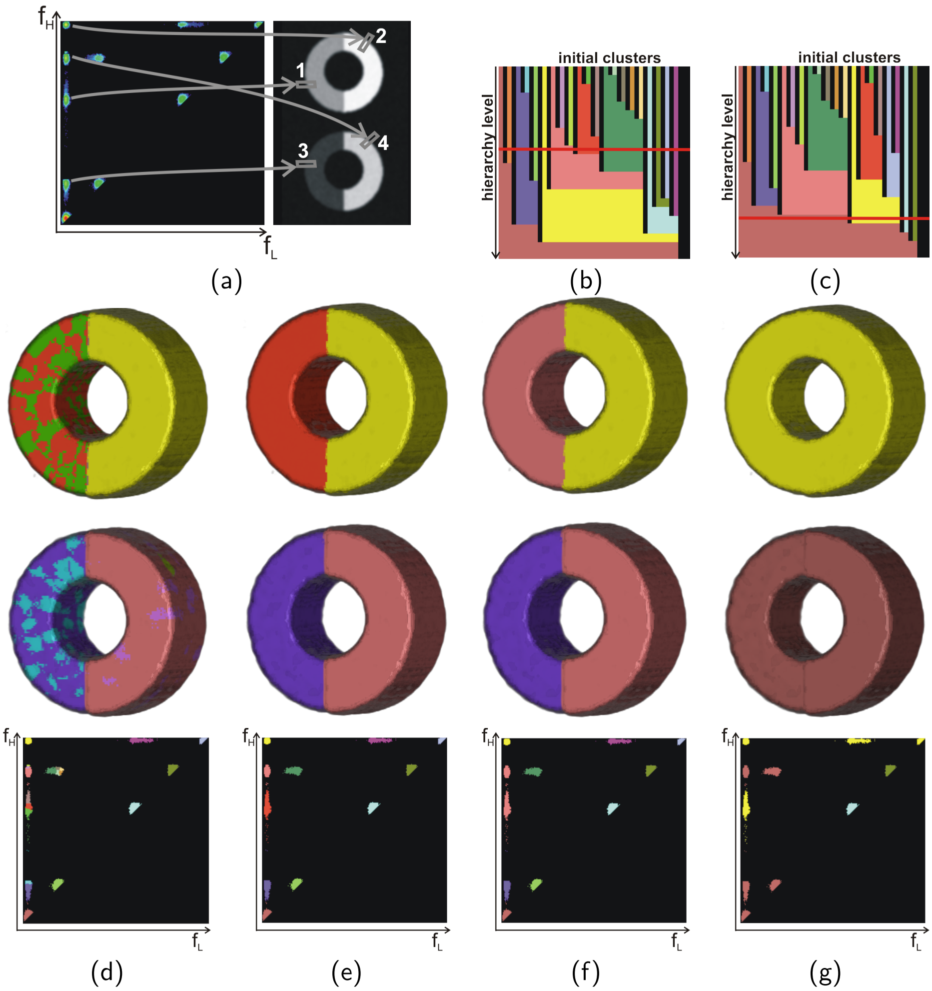


Figure 4.4: *Phantom dataset. (a) shows a slice and the correspondence of the four most important boundaries in the LH histogram. (d-g) show renderings at different hierarchy levels. (d) shows the initial clustering. In (e) and (f) the hierarchy based on the LH similarity was used (see dendrogram (b)). The level chosen in (e) yields each of the four boundaries in an own cluster. In (f) boundaries 1 and 4 are grouped since they are close in the LH space. At the hierarchy level highlighted in (b) the spatial similarity was chosen. A hierarchy that combines both similarity measures (c) results in the desired clustering (g). The dendrograms (b, c) show the order of grouping. The right child inherits the color from the left child.*

4.3.3 Similarity in the volume

This second similarity measure is designed to group boundaries that are connected in the volume. It is an alternative to the similarity in the LH space. Figure 4.4 illustrates the use of the spatial similarity on a phantom consisting of four materials. Figures 4.4d-4.4f were generated using the similarity in the LH space. In 4.4f boundaries 1 and 4 are grouped since they are close in the LH space. Combining the LH similarity with the volume similarity helped to group boundary 1 with 2 and 3 with 4 (see Figure 4.4g). We evaluate the spatial similarity as the number of direct neighborhood relations between clusters. Having the initial clustering elements, we evaluate the number of neighborhood relations for each pair of elements $NR(e_i, e_j)$. We look at all 26 neighbors of every voxel $v_i \in e_i$ and count how many of them belong to element e_j, i.e.,

$$NR(e_i, e_j) = NR(e_j, e_i) = \sum_{v_i \in e_i} \sum_{v_j \in e_j} N_{26}(v_i, v_j); e_i \neq e_j$$

where $N_{26}(v_i, v_j)$ is 1 if v_i and v_j are neighbors and 0 otherwise. Since a cluster cannot neighbor with itself we define $NR(e_i, e_i) = 0$. The total number of neighborhood relations of the element is

$$NR(e_i) = \sum_{e_j} NR(e_i, ej)$$

In order to group clusters that belong to the same boundary, we weight the relations between voxels. Then the sum of weighted relations between two elements

$$R(e_i, e_j) = R(e_j, e_i) = \sum_{v_i \in e_i} \sum_{v_j \in e_j} N_{26}(v_i, v_j) r(v_i, v_j); e_i \neq e_j$$

where $r(v_i, v_j) \in [0, 1]$ reflects the directional coherence of boundaries in voxel v_i and v_j (similarly as used for C_{NC} in section 3.5).

Finally, to make the similarity measure independent on the size of the cluster, we normalize $R(e_i, e_j)$ by $NR(e_i)$ and $R(e_j, e_i)$ by $NR(e_j)$. Since generally $NR(e_i) \neq NR(e_j)$ we choose the maximum of the two relations to make the similarity measure symmetric

$$s(e_i, e_j) = s(e_j, e_i) = max \left\{ \frac{R(e_i, e_j)}{NR(e_i)}, \frac{R(e_j, e_i)}{NR(e_j)} \right\}$$

Note that $NR(e_i)$ cannot be zero as long as there are at least two clusters in the data. Further we define $s(e_i, e_i) = 1$ in order to obtain a similarity measure that is semi-metric.

The presented relationships between initial clustering elements are evaluated in one pass through the volume. The initial clusters C_i contain each one element e_i. Therefore $NR(C_i) = NR(e_i)$, $NR(C_i, C_j) = NR(e_i, e_j)$, $R(C_i, C_j) = R(e_i, e_j)$ and $s(C_i, C_j) = s(e_i, e_j)$.

When clusters C_i and C_j join, we have to recalculate the number of relations of the joint cluster $C_{i \cup j}$

$$NR(C_{i \cup j}) = NR(C_i) + NR(C_j) - 2NR(C_i, Cj)$$

and update its relations with all other clusters C_k; $C_k \neq C_i$, $C_k \neq C_j$

$$NR(C_{i \cup j}, C_k) = NR(C_i, C_k) + NR(C_j, Ck)$$

$$R(C_{i \cup j}, C_k) = R(C_i, C_k) + R(C_j, Ck)$$

$$s(C_{i \cup j}, C_k) = max \left\{ \frac{R(C_{i \cup j}, C_k)}{NR(C_{i \cup j})}, \frac{R(C_{i \cup j}, C_k)}{NR(C_k)} \right\}$$

For both similarity measures the complexity of joining two clusters is $O(K)$, where K is the number of clusters C_k, i.e., clusters remaining in the hierarchy.

4.4 Hierarchy interaction framework

A major disadvantage of clustering techniques is the need to define a similarity threshold at which the desired partitioning of the elements should exist. Tuning such a threshold by changing its value and restarting the clustering algorithm is a rather cumbersome process. Furthermore, this would be assuming that at one level all desired objects can be perfectly partitioned. Our framework enables a real-time interaction with the cluster hierarchy. This approach is much more intuitive, since the user does not have to deal with threshold values and gets a better insight into the dataset. Moreover, in our method the objects can be selected at different hierarchy levels.

The second weakness of clustering techniques lies in the similarity measure. It is often the case that more criteria need to be combined to cover the crucial aspects of the data. However, the criteria are usually of different nature and their simple combination (such as the most often used weighted sum) have little meaning. In practice these weights are usually tuned for certain dataset(s) and their performance for other data is questionable. Again, tuning of these

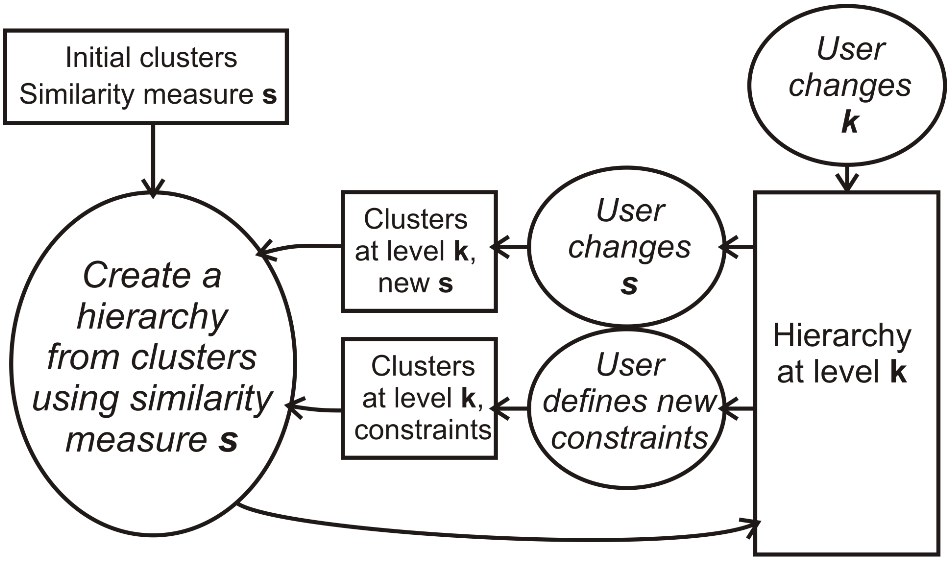


Figure 4.5: *The hierarchy interaction framework. First, the hierarchy is generated for the initial clusters using a default similarity measure* **s**. *Then the user can interact with the hierarchy by changing the level* **k** *and (de)selecting clusters. The user can also change the similarity measure or apply constraints to selected clusters.*

weights is difficult, since different datasets may have distinct sensitivity to certain criteria. In our framework, no weights need to be defined, since only one similarity measure is used at a time. However, the user can pick different similarity measures while moving in the hierarchy. This means that different parts of the hierarchy can be generated by different similarity measures.

We illustrate the use of the framework in the LH space using the two similarity measures described in sections 4.3.2 and 4.3.3. However, this is just one implementation of the framework. Using other space and similarity measures or their combination would be possible.

The input to the interaction process (see Figure 4.5) are the initial clustering elements and a default similarity measure **s**. After the hierarchy is generated, the user can interactively change the level **k** which results in a new partition of the elements. If moving to the next hierarchy level does not yield the expected clustering, the user might want to change the similarity criterion **s**. In that case a new hierarchy is generated on the clusters from level **k** using another similarity measure (see Figure 4.6).

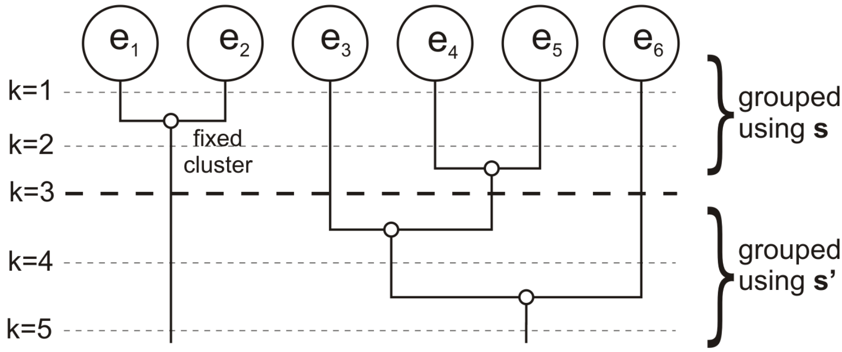


Figure 4.6: *A hierarchy cut at level* $\mathbf{k} = 3$ *yields clusters that are further grouped using another similarity measure. The hierarchy can be, therefore, a combination of several similarity measures. The cluster consisting of elements* e_1 *and* e_2 *was fixed on level* 2 *and taken out of the hierarchy.*

In general, there might be no single hierarchy level at which the clusters are grouped according to the user's expectations. In order to handle such situations, several constraints can be introduced into the hierarchy:

- *Fixing a cluster.* If a cluster is fixed, it is taken out of the hierarchy so that it is not split or joined with other clusters when the hierarchy level changes. This can be used for situations where certain objects are well selected at different levels.

- *Explicit splitting(joining) of selected cluster(s).* Selected clusters are manually joined or split and the hierarchy is adjusted accordingly.

For more complex datasets, the user might find it difficult to observe and evaluate the completeness of all objects of interest at the same time. Since it is much easier to interact only with one object at a time, we allow the possibility to visualize only one branch of the hierarchy. The users can then concentrate on one object of interest and after fixing the object at a certain level, they can focus on another object.

The hierarchy interaction allows a large interaction freedom based on intuitive actions. However, if an application-specific knowledge is available (approximate intensities of tissues, number ad sizes of clusters, etc.), it is possible to automatically include certain constraints of the hierarchy without compromising the generality of the framework.

4.5 Transfer functions from clustering

The result of a clustering is a labeling of the LH histogram. Since the color and opacity are defined for every label, we obtain a 2D transfer function. By default, the clusters are automatically assigned random colors and full opacities. When two clusters join they should appear in the same way, i.e., have the same color and opacity. In our implementation the smaller cluster inherits the color and opacity from the bigger (dominant) cluster. Naturally, the user is free to change colors and opacities of any cluster.

Since our rendering is implemented using the VolumePro1000 board [46] that allows only 1D TFs, we label the volume and perform a 1D TF on the labels. The labeling, however, needs to be done only once for the initial clusters, therefore the speed of interaction with the hierarchy is not compromised. One of the advantages of using the labels is the possibility to combine the TF with a segmentation (as shown in section 3.5). The disadvantage is that techniques such as fuzzy borders between clusters are not possible. However, the fuzzy borders could be implemented, e.g., in a GPU-based renderer.

4.6 Results

We demonstrate the functionality of our methods on four different datasets. The first two are the CT datasets of the engine and the carp. The clusters in the LH histogram and corresponding renderings are shown in Figure 4.7. These datasets contain only few important boundaries. We have only used the LH-based similarity. Visualizing these datasets was fairly easy. All major boundaries can be seen at the same time. Therefore, one can easily see if the cluster should grow more or be fixed. In the tooth dataset (Figure 4.8) most of the boundaries could be completely visualized at one hierarchy level. In order to completely select the enamel-air boundary, we had to first fix the dentin-air boundary. In Figure 4.8d we switched to the spatial-based similarity in order to group the enamel-air and dentin-air boundaries. In this dataset we had to make certain boundaries (semi-)transparent in order to evaluate the quality of grouping in the underlying boundaries.

In Figure 4.9, we show several combinations of important boundaries in the CTA dataset of a head. All the shown boundaries could be selected at the same level in the hierarchy. In Figure 4.9d we further increased the hierarchy level in order to select and easily remove the remaining boundaries inside the skull.

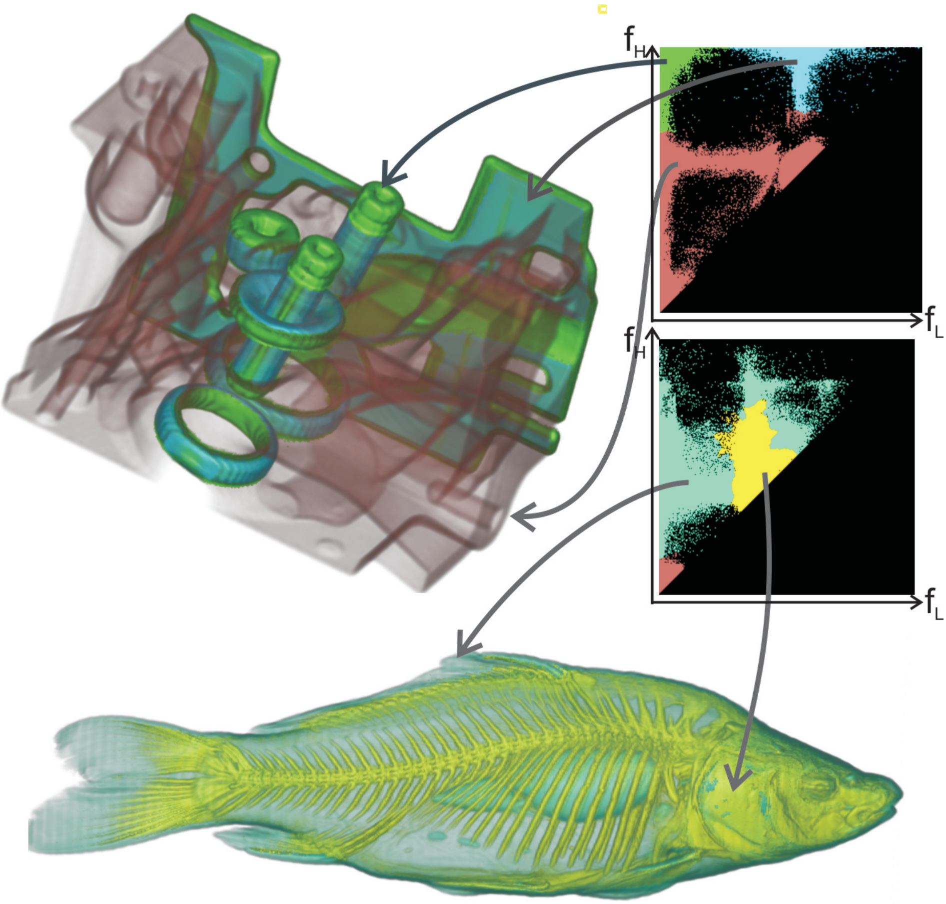


Figure 4.7: *Renderings of the carp (256x256x512) and engine (256x256x110) datasets. The green and blue boundaries of the engine were visualized at a certain hierarchy level. After fixing them at that level, a further change of the level grouped the brown boundary. For the carp we first fixed the bone boundaries at the level where all of them were grouped and then continued joining the remaining air boundaries.*

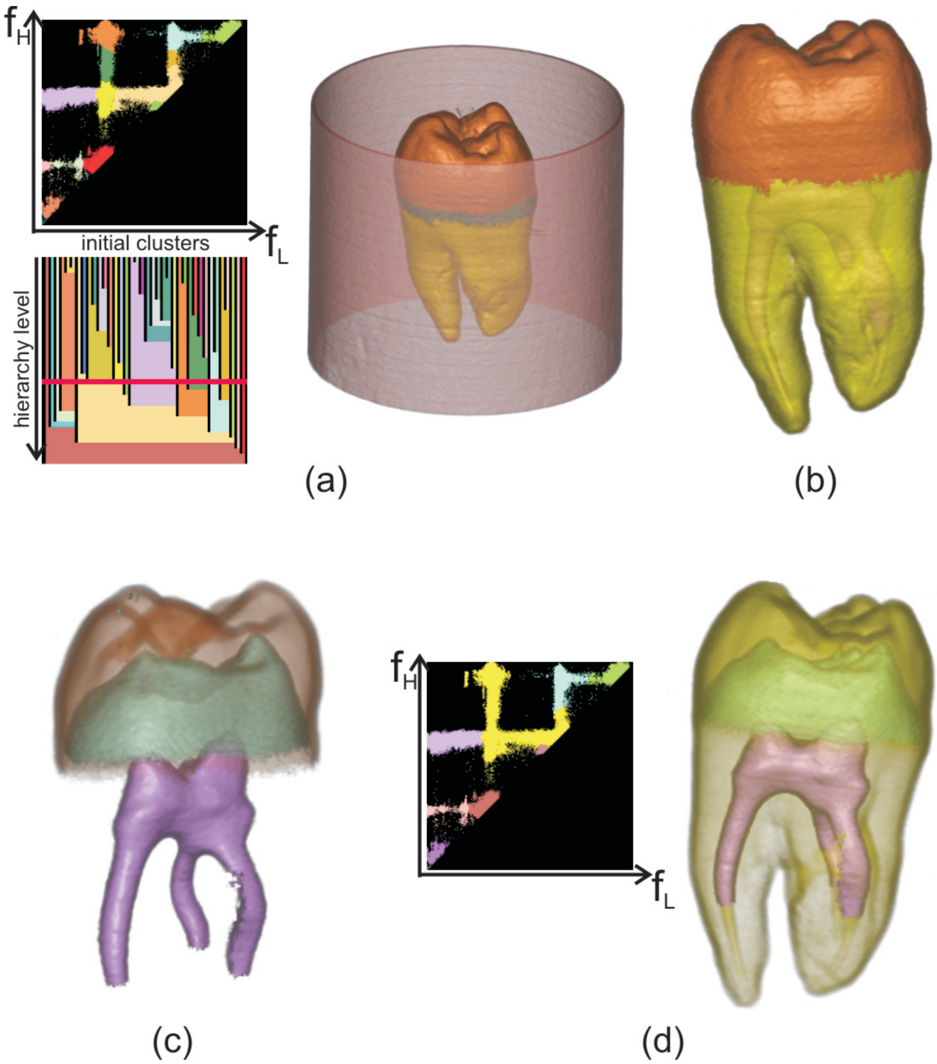


Figure 4.8: *Tooth dataset (256x256x161). In (a) the major boundaries were selected by an easy choice of the hierarchy level (see the red line in the dendrogram). The LH histogram shows the corresponding selection. Then the surrounding (pink) boundary was removed (b) and after fixing the dentin-air (yellow) boundary the rest of the enamel-air (orange) boundary was selected. In (c) the inside boundaries are shown after setting the outside boundaries (semi-)transparent. In (d) we illustrate the possibility of grouping the outer boundaries (yellow). We achieved this by choosing the spatial-based similarity measure.*

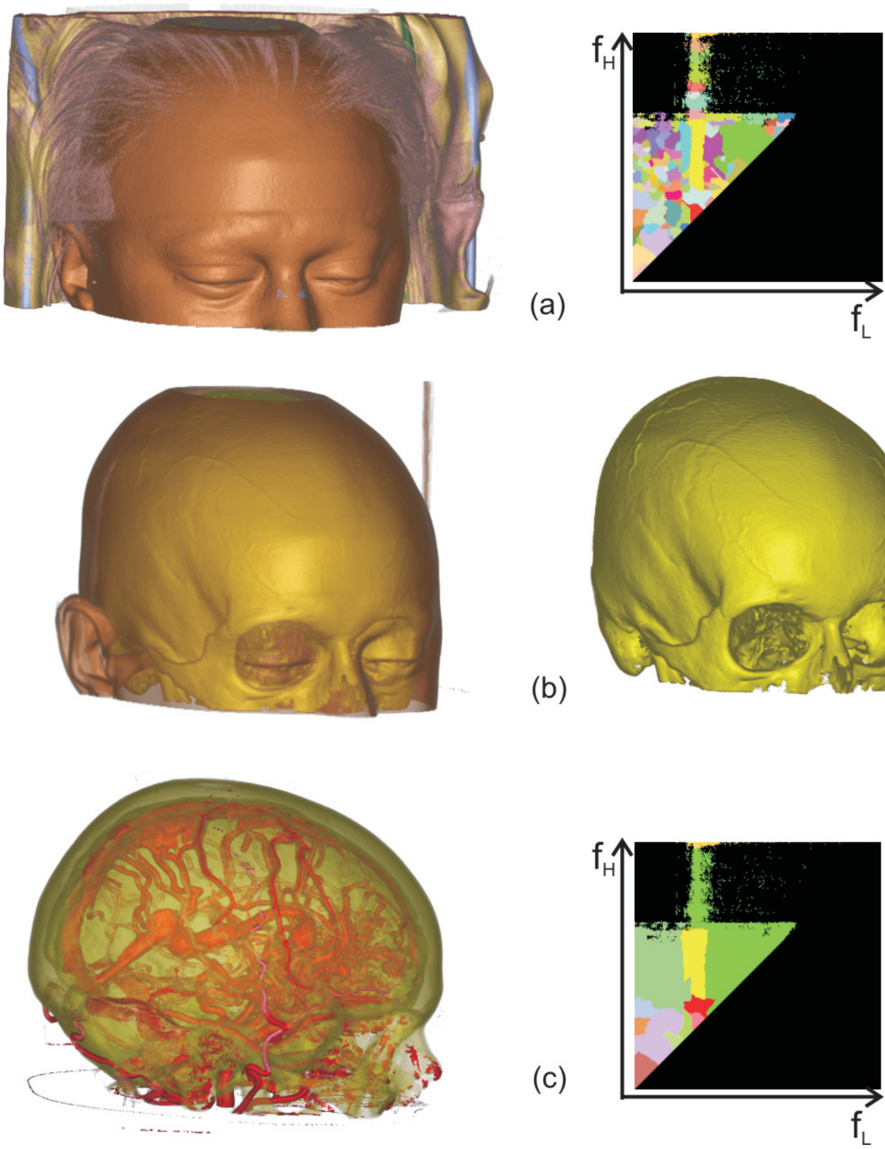


Figure 4.9: *CTA dataset of a head (512x512x286). One level in the hierarchy and corresponding rendering is shown in (a). In (b) only two of the clusters were chosen: the skin and the skull. In (c) the skull and the vessels were selected, fixed and the color of the vessels was changed. Then by changing the level all other clusters inside the head (green) were grouped and removed from the visualization.*

Note that in the presented results we only used moving in the hierarchy and one constraint, i.e., fixing certain clusters after they were well selected. We did not use other constraints such as the explicit joining/splitting of the clusters. Because of the fast hierarchy interaction, it was easy to select the proper hierarchy level or to discover that no such level exists. If no such level existed, the constraints had to be introduced. For the tested datasets we obtained a maximum of around 200 initial clustering elements. With such a low number, the hierarchy and its modifications could be generated in real time without any noticeable delays.

4.7 Summary and conclusions

In this chapter we have presented a flexible framework that enables an intuitive interaction with the hierarchy of clusters. The clusters serve as the transfer function for the direct volume rendering. The user works with objects and therefore does not need to understand the underlying transfer function domain. Moreover interacting with the hierarchy replaces the commonly used parameters such as clustering thresholds and weighting constants for the similarity measures. Our framework enables an easy combination of different similarity measures.

We have presented a clustering method based on the LH space. This space seems more suitable for clustering of material boundaries than the commonly used space of intensity and gradient magnitude. We have introduced two measures that evaluate similarity of clusters in the LH space and in the volume space.

However, the current method is sensitive to the initial clustering that is given by the peaks in the LH histogram. If there is more than one boundary in one initial clustering element, in our implementation they cannot be separated. It would be interesting to split that element and refine the selection down to the level of single bins. Such operation would fully fit into the framework, since it would only mean replacing the element in the hierarchy by a sub-tree that represents its division.

The presented implementation of the framework used the LH space as the domain for the TF. However, the framework is general and could be applied to any TF domain. A trivial extension is to perform clustering in the mirrored LH space (see section 3.4) as shown in Figure 4.10. The clustering could then, e.g., help to select boundaries of one material by automatically selecting all clusters lying on a horizontal line.

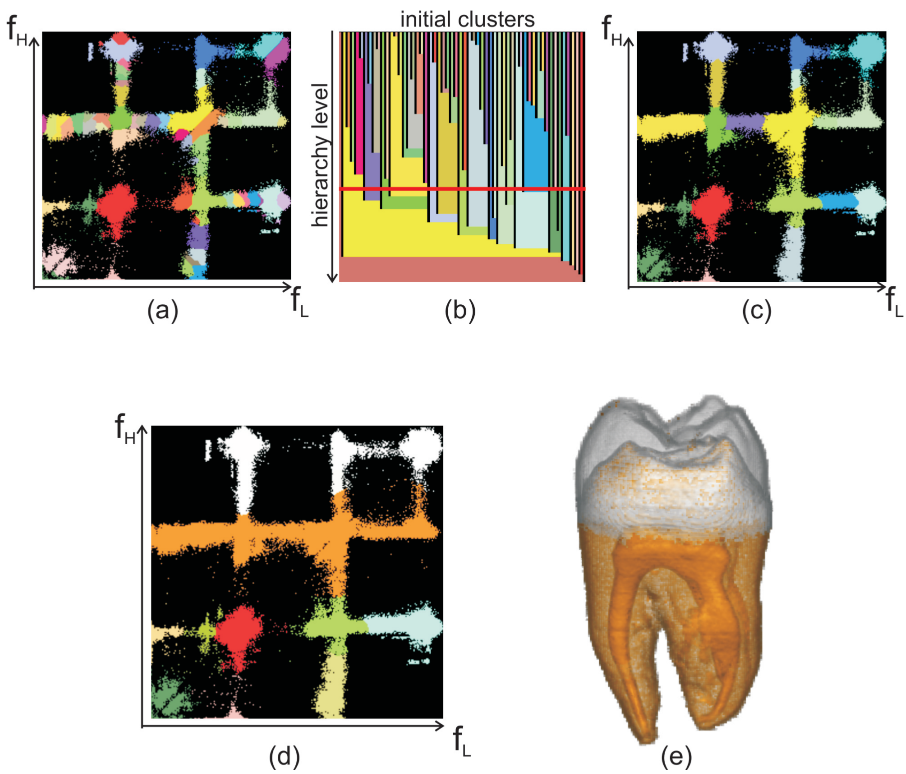


Figure 4.10: *Clustering of the mirrored LH histogram of the tooth dataset. (a) shows the initial clustering and (c) clustering at a level set in the dendrogram (b). In (d) all clusters belonging to the enamel and dentine were set to white and orange respectively. The rendering is shown in (e).*

Having the hierarchy of clusters, a number of additional automation steps can be performed. Instead of introducing all hierarchy levels to the user, an analysis of the clusters can be performed based on a priori knowledge. Information such as the position, size or compactness of the clusters can be taken into account. We have performed a study showing that, in similar types of datasets, one can assume the optimal clusters to occur at similar hierarchy levels and have similar properties. These typical values can then be used as a preset for the first rendering.

In our current implementation the user interacts with objects by pointing at corresponding clusters in the TF domain. An improvement in the intuitiveness would be a 3D interaction with the rendering since the user would be completely shielded from the transfer function.

Chapter 5

Local Transfer Functions

5.1 Introduction

Conventional global transfer functions are sensitive to changes of data properties through the dataset. In this chapter we illustrate weaknesses of the global transfer functions. Then we introduce a new concept of so-called local transfer functions that aims to overcome those weaknesses. We discuss a framework of local transfer functions and illustrate its use on several examples.

5.1.1 Global transfer functions

Typically, the domain of the TF consists of one or more data properties that can be evaluated at every point of the data. The TF maps the data properties to optical properties (Figure 5.1). In the common visualization scenario, there is one (global) TF that is applied to every point of the data. The basic assumption behind global TFs is that the properties of each object lie within a certain range in the TF domain. When using the TF we assume to be able to define the range such that the object (see section 2.1) of interest is completely selected without selecting other objects (good sensitivity and specificity of the TF-based classification). If the data properties (typically intensities) of different tissues have a high contrast and are stable over the dataset, one can easily visualize them by selecting corresponding ranges in the global TF.

The assumption is often made that points belonging to the same material appear having the same intensity (data property, in general) in the whole dataset. However, this assumption is the weak point of the global TFs. There are several scenarios in which this assumption does not hold and therefore the global TF may encounter problems. A TF that is tuned for a certain region of the dataset

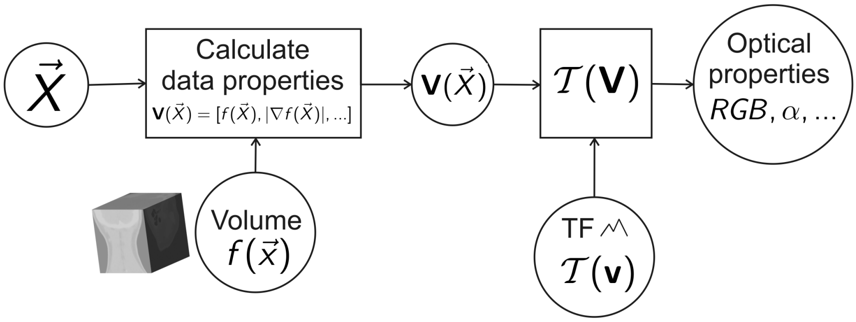


Figure 5.1: *The pipeline of global transfer functions.*

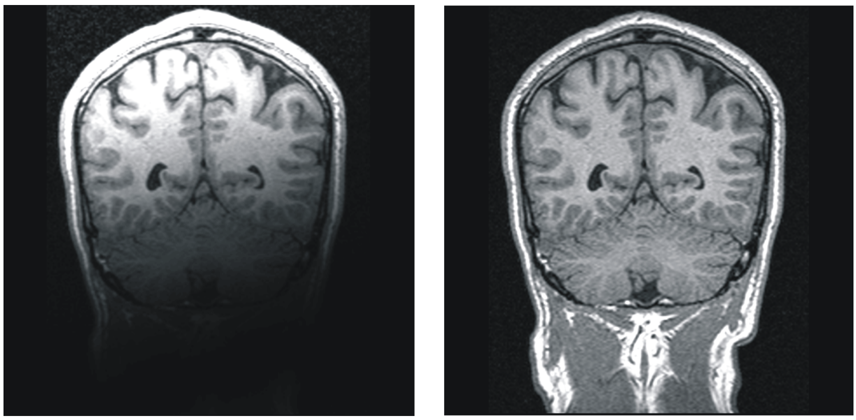


Figure 5.2: *An example of a strongly biased (left) and unbiased (right) MR dataset of a human head. Note that some tissues, although in reality having constant properties, may yield a wide range of values. For the purpose of illustration a multiplicative linear bias field was simulated.*

could then yield incorrect results in other regions. Some examples where this assumption is violated:

- **Variations in MR data:** Smooth variations of image intensities, such as the bias (see Figure 5.2), are commonly present in MR datasets. The cause of these changes are factors such as inhomogeneities of the magnetic field and the coil, interferences with other fields or the magnetic properties of the materials being scanned. These factors result in unstable measurements of material properties through the data volume. Therefore a TF defined for visualizing a certain material in a part of the volume may not be valid for other parts where the data values have changed.

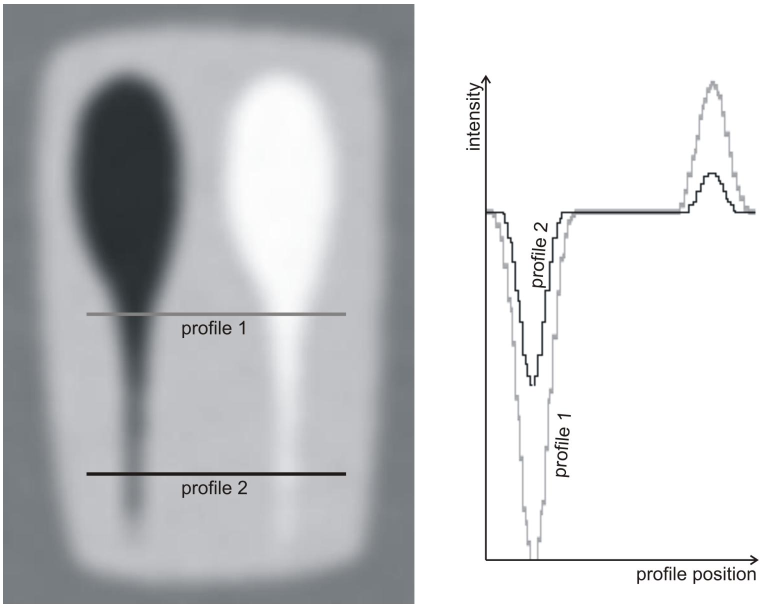


Figure 5.3: *An artificial dataset that simulates the intensities of thin structures. Note that due to the blurring by the point spread function of the scanner the intensities of thin materials resemble to those of the background (i.e., light materials become darker and dark become lighter)*

- **Thin objects:** The variance in the measured intensities can also be caused by changes of the size of the objects. If the size of the object becomes small relative to the size of the point spread function (PSF) of the scanner, the intensity of the material appears to be closer to the intensity of the background material (see Figure 5.3). This happens due to the blurring by the PSF.

- **Changes of the materials:** In some situations the object of interest may not consist of homogenous material(s), but of materials that gradually change through the volume. Typical examples of this are objects of changing densities, such as bones or vessels containing non-uniformly distributed contrast agent (see Figure 5.4).

- **Ambiguity of the objects:** If more objects have the same intensity (data properties, in general) one cannot show them in contrast using a global transfer function. The intensities might be the same either because the objects are made of materials that appear with the same intensities or the intensities may overlap because of the changes mentioned above.

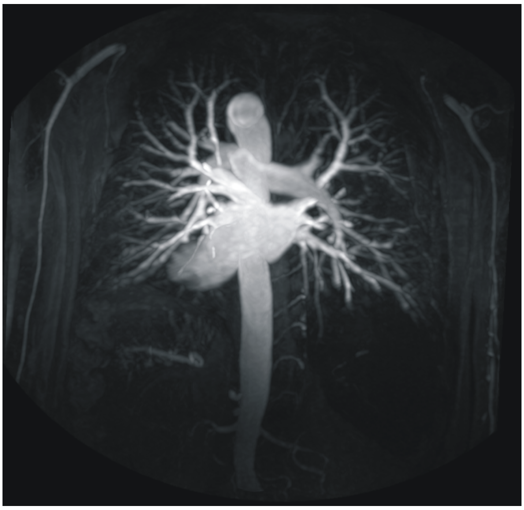


Figure 5.4: *A maximum intensity projection (MIP) of a MR cardiac dataset. The contrast agent is not distributed uniformly which reflects as change in the intensity of the arteries. The intensity variations are probably also influenced by the bias field and thin structures.*

5.1.2 Related work

In problematic situations, such as those mentioned above, the global TF cannot be used effectively. A common solution is to either remove the problematic artifacts or to use a segmentation. If the variation, such as the bias caused by the inhomogeneity of the magnetic field, is correctly estimated, it can be removed from the data [50], e.g., by fitting a simplified model. However, some influences are difficult to model causing the related problems to remain. Thin parts of the material might be addressed by specially designed methods [51] or by enhancing them in order to obtain intensity values similar to those of the rest of the material [52]. For this purpose special local filters detecting and enhancing the structure of interest might be used. The limitation of the filters is the need to adapt them to a large variety of shapes that an object of interest may have.

Segmentation techniques may use a wide range of models and algorithms [53], often tailored to a given application [54, 55]. Instead of visualizing objects by selecting corresponding data ranges in the TF, the objects are segmented and each segment is assigned a different optical property. By using segmentation, however, we give away some advantages of TFs. One of the most important strengths of the TF is the ability to show directly the properties of the data (i.e., the TF domain), not just the result of a segmentation. Furthermore, the TF provides the possibility of flexible modifications of the visualization settings. An important property of segmentation is that every object that is supposed to be shown in contrast has to be segmented. For the purpose of visualization, however, the need to assign all voxels to a (fuzzy) label can be rather cumbersome.

The work of Correa et al. [56] is probably the most related to the work presented in this chapter. Their goal is to enable locally modified TFs for the purpose of time evolving illustration and focus-context visualizations. We generalize the idea of locally changing TFs by introducing a general framework. We discuss several possible approaches towards the definition of local TFs. Our main motivation is the use of local TFs for situations where the global TF fails or is difficult to tune. We further suggest and discuss an alternative interpolation between multiple TFs that is better suited for gradual changes of data properties. We want to handle the TFs locally in order to adapt to variations of data properties (due to acquisition inaccuracies or inhomogeneities of the material) as well as variations of the visualization criteria that the user may have.

5.2 Local transfer functions (LTF)

The idea of LTFs aims at compensating for some of the possible changes of data properties that would hamper the global TF. The LTFs would keep the advantages of TFs while adding spatial information.

One of the main motivations for using LTFs is that they are often easier to define than a global TF. Figure 5.5 illustrates this in a phantom with three materials that overlap due to bias and noise. It is impossible to distinguish the materials in the global histogram. That is often an indication that there might not be any global TF that can show the materials in contrast (as seen in the rendering that attempts to use a global TF). If we, however, take into account only a certain neighborhood (sub-volume), the variance of the data values decreases and the materials appear as clear peaks. This is an indication that there might be a locally valid TF. If the peaks in the histogram are clearly visible, the definition of a TF becomes relatively easy. The user can position various interaction widgets to highlight data ranges corresponding to some of the peaks. Clarity of the peaks also facilitates the use of automatic methods that are based on detection of peaks in the histograms, such as the one described in Chapter 4. In order to further increase the clarity of the histogram peaks used in this chapter, we have applied the LH method for deblurring the partial volume voxels at material boundaries from Section 3.4.

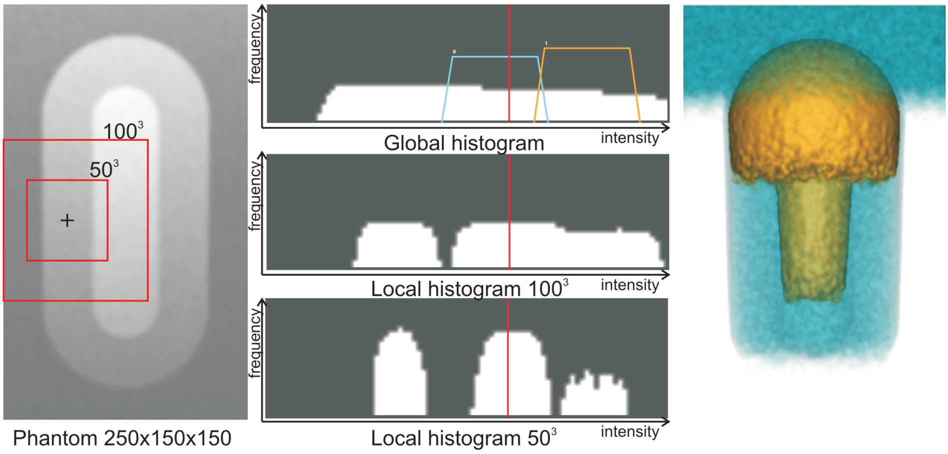


Figure 5.5: *A comparison of global and local histograms for a phantom dataset containing strong bias and noise. In the global histogram materials cannot be distinguished. An attempt to show the materials in contrast using a global TF fails (right). In local histograms, however, the peaks become clear. The center of the cubic neighborhood is marked by the cross. The red marker in the histograms represents the data value at the cross location.*

In general, the idea of LTFs is to use several locally defined TFs, or ultimately a different TF in every point. In other words, the transfer function can be continuously modified through the volume. We can define the TF field as a mapping $\mathcal{L}$ that maps the spatial position to a local transfer function valid at that position $\mathcal{L} : \mathbb{R}^3 \rightarrow \mathcal{T}$.

Figure 5.6 shows the setup for using local transfer functions. The local TF to be used is determined by the TF field. This TF is then used in the same manner as in the case of a global TF. In the next section, we describe some approaches to generate the TF field.

5.3 The TF field

In this section, we discuss several possible solutions for constructing the TF field if one or more LTFs are known (Figure 5.7). Defining a TF locally is usually easier than globally, however, more TFs need to be defined. If too many LTFs had to be defined their advantages would disappear. Therefore we assume that a set of N sparsely defined, locally valid TFs are available at locations $\vec{X}_i$. The aim of our investigation is to reconstruct the TF field $\mathcal{L}(\vec{x})$ knowing the set $\mathcal{L}(\vec{X}_i)$.

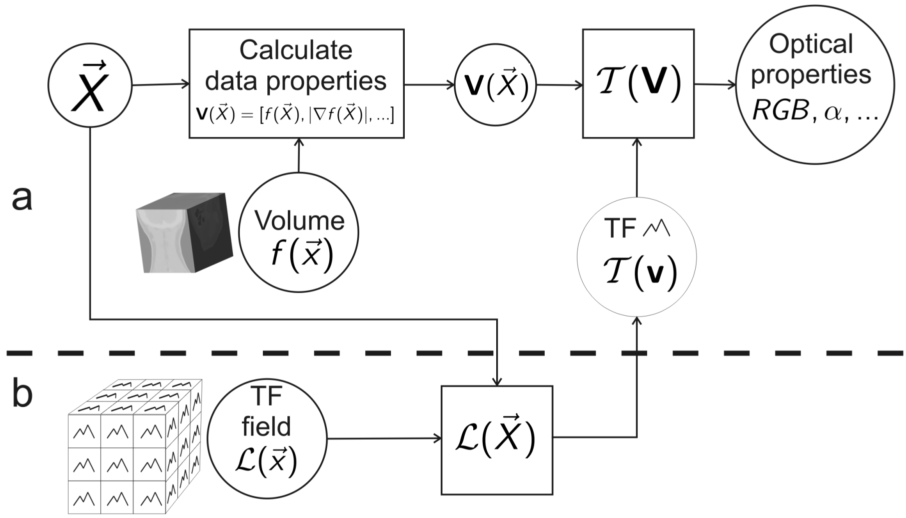


Figure 5.6: *The framework for local transfer functions. (a) The pipeline of global transfer functions. (b) In the local TF approach the TF that will be at the current position $\vec{X}$ is determined from the TF field $\mathcal{L}(\vec{x})$.*

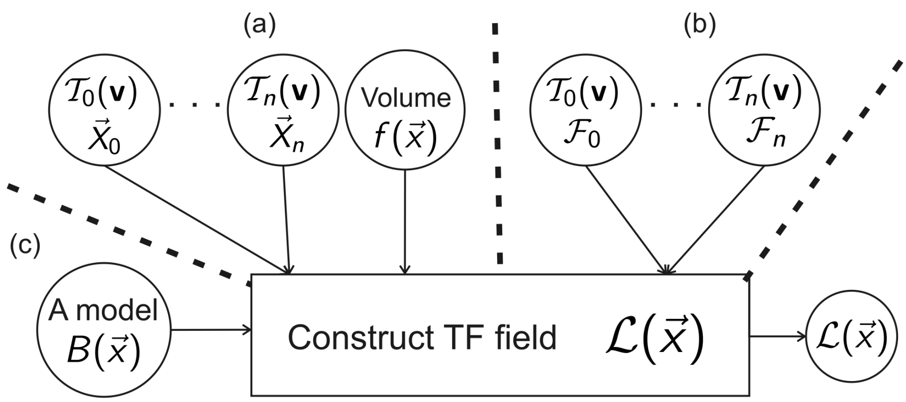


Figure 5.7: *Several options to reconstruct the TF field: (a) By using a set of TFs defined in reference positions, (b) By using a set of TFs with fuzzy-segmented regions of influence, (c) Another model describing, e.g., the intensity changes, can be added.*

5.3.1 Combining several local TFs

We reconstruct $\mathcal{L}(\vec{x})$ using a combination of known local TFs $\mathcal{L}(\vec{X}_i)$ (Figure 5.7a). Note that $\mathcal{L}(\vec{X}_i) = \mathcal{T}_i(v)$. We combine them using a weighted sum

$$\mathcal{L}(\vec{x}) = \sum_{i=1}^{N} w_i(\vec{x}) \cdot \mathcal{L}(\vec{X}_i) \tag{5.1}$$

where $w_i(\vec{x})$ is the weight of the local transfer function $\mathcal{L}(\vec{X}_i)$ at point $\vec{x}$ and $\sum_{i=1}^{N} w_i(\vec{x}) = 1$.

In order to evaluate equation 5.1 the weights w_i have to be available at every point. The weight w_i is a normalized influence I_i at point $\vec{x}$

$$w_i(\vec{x}) = \frac{I_i(\vec{x})}{\sum_{j=1}^{N} I_j(\vec{x})} \tag{5.2}$$

The influence specifies how the local TF $\mathcal{L}(\vec{X}_i)$ relates to point $\vec{x}$. One can consider different criteria for determining its influence at any other point. We have experimented with four approaches. Each of the locally valid transfer functions $\mathcal{L}(\vec{X}_i)$ is coupled with a reference point $\vec{X}_i$ at which its influence is maximal. Assuming that the difference between TFs can be based on their spatial distance in the volume we might use one of the following approaches.

Using spatial distance

The easiest approach that takes into account spatial location is the nearest neighbor decision, i.e., the TF with the closest reference point is used, the other TFs have no influence.

$$I_i(\vec{x}) = \begin{cases} 1 & \|\vec{x} - \vec{X}_i\| < \|\vec{x} - \vec{X}_j\|, \forall j \neq i \\ 0 & \text{otherwise} \end{cases}$$

This is a simple approach whose implementation does not require any real combination of the TFs. Figure 5.8a illustrates the tree different TFs combined by the nearest neighbor decision. The medium TF is set to highlight that part of the phantom. Similarly to the work by Correa et al.[56], we might want to remove the discontinuities between the TFs by interpolating the TFs (see Figure 5.8b). We make the influence of the reference point inversely proportional to the Euclidian distance d

$$I_i(\vec{x}) = \left(\frac{1}{d(\vec{x}, \vec{X}_i)} \right)^p \tag{5.3}$$

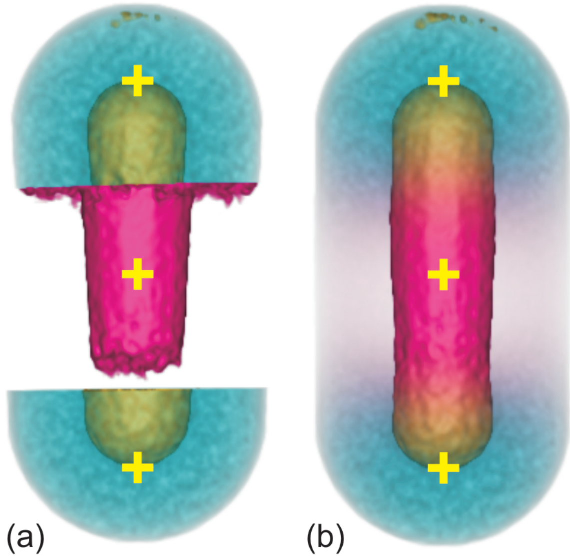


Figure 5.8: *Volume rendering of the phantom dataset that uses three LTFs with reference points positioned as marked by the crosses. The middle TF is set to visualize the middle part of the phantom in a different way. In (a) the nearest neighbor combination is used. In (b) the LTFs are continuously combined by using the spatial distance.*

where the exponent p controls the sharpness of the boundary between the influences of transfer functions, i.e., as p increases the smoothly changing influence becomes nearest neighbor influence. The inverse relation ensures reference points with distance close to zero to prevail over remaining points. The combination of equations 5.2 and 5.3 represents the so-called Shepard method (Amidror [57]).

Using similarity of local histograms

The results of the spatial distance-based combination are highly dependent on the placement of the reference points. This is useful for position-dependent visualizations, such as the focus plus context [58, 59]. However, if we want to adapt to the data changes, which is also the goal of this chapter, another approach is necessary. Such a data-driven approach that adapts to data changes is expected to be less sensitive to the placement of the reference points. The assumption we make here is that similar TFs can be used for similar neighborhoods in the data. In order to illustrate this approach we assume that local neighborhoods can be compared by comparing their histograms.

For every reference point $\vec{X}_i$, we construct a reference histogram $H_{\vec{X}_i}$. For every point $\vec{x}$ in the volume its neighborhood histogram $H_{\vec{x}}$ is calculated. The influence of a transfer function $\mathcal{T}_i$ is evaluated with a similarity measure that compares $H_{\vec{X}_i}$ and $H_{\vec{x}}$. We propose to use a normalized similarity between two histograms G and H

$$S(G,H) = \frac{\sum_{i=1}^{K} G(i)H(i)}{\sqrt{\sum_{i=1}^{K} G(i)^2 \sum_{i=1}^{K} H(i)^2}} \tag{5.4}$$

where K is number of bins of the histograms. The size of the neighborhood from which the histograms are constructed is crucial for this approach. The comparison of the histograms performs the best when both histograms contain the representative intensities in the neighborhood. It is therefore dependent on the data and needs to be defined for each case. Similarly to Eq. 5.3 the influence is computed as

$$I_i(\vec{x}) = \left(\frac{1}{1 - S(H_{\vec{x}}, H_{\vec{X}_i}))}\right)^p \tag{5.5}$$

Assuming the similarity of local histograms corresponds to the changes in the data properties this method adapts to those changes. In Figure 5.9 the histogram-based influence yields almost identical results for different positions of the reference points.

The above approach compares neighborhoods of $\vec{X}_i$ and $\vec{x}$ without taking into account their spatial relation in the volume and the continuity of the changes between the points. One could include those criteria, e.g., by computing the accumulative cost of arrival from $\vec{X}_i$ to $\vec{x}$ using region-growing. The cost function could be evaluated as the similarity of the histograms used in Eq. 5.4.

Another possible approach that we introduce is to evaluate similarity of local neighborhoods from Eq. 5.4 using cross-correlation of local histograms

$$C(G,H,t) = \frac{\sum_{i=1}^{K} G(i)H(i+t)}{\sqrt{\sum_{i=1}^{K} G(i)^2 \sum_{i=1}^{K} H(i)^2}} \tag{5.6}$$

where $t = arg_t max\{C(G,H,t)\}$ is the shift between histograms. This shift allows to match histograms of similar shapes that have been shifted due to some of the phenomena that may occur in the data as illustrated in section 5.1.1. This measure yields the maximal correlation and the shift t at which it occurs.

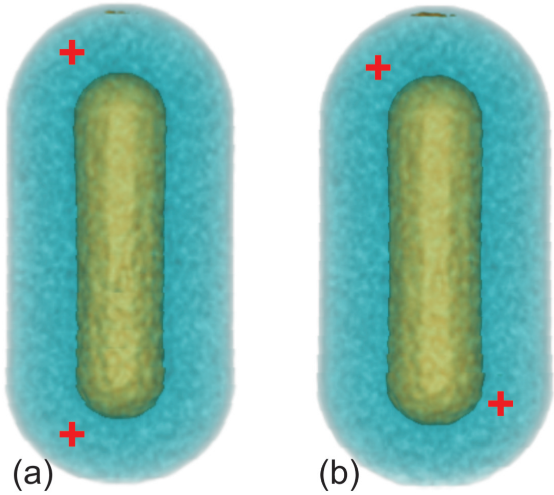


Figure 5.9: *The histogram-based influence does not show much sensitivity to the placement of the LTFs. The renderings in (a) and (b) are almost identical. The neighborhood size was* 100^3.

Similarly one could allow for scaling of properties between neighborhoods by using

$$C(G, H, s) = \frac{\sum_{i=1}^{K} G(i)H(i \cdot s)}{\sqrt{\sum_{i=1}^{K} G(i)^2 \sum_{i=1}^{K} H(i)^2}} \tag{5.7}$$

which yields the maximal correlation and the scale $s = arg_s max\{C(G, H, s)\}$ at which it occurs. Scaled neighborhoods occur, e.g, when the intensity variations are caused by an MR bias field. The common model for the intensity distortion by a bias field is a multiplicative model [54]

$$I'(\vec{x}) = I(\vec{x}) \cdot B(\vec{x}) + N(\vec{x}) \tag{5.8}$$

where I' is the measured data intensity at point $\vec{x}$, I is the original intensity without distortion, B is the bias field and N the noise.

Using a segmentation

In order to illustrate the generality of the framework of LTFs we can also obtain the influence for each of the local TFs by means of a specific (model-based) segmentation method (see Figure 5.7b). For each label L a different transfer function $\mathcal{T}_L$ can be defined and its influence then depends on $\mathcal{F}_L$. An example of this approach is shown in Figure 5.19.

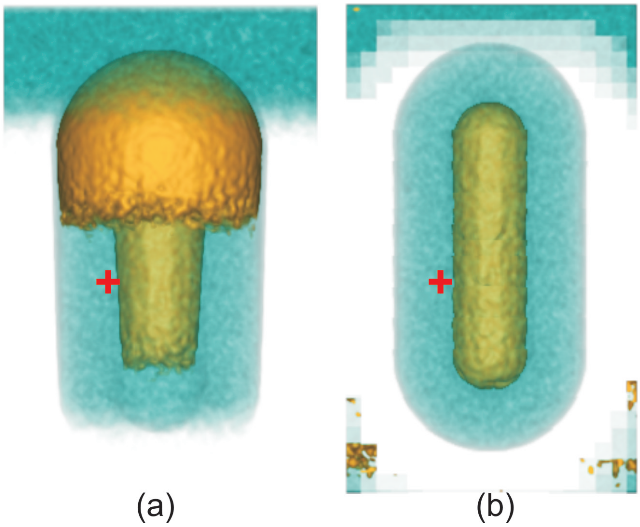


Figure 5.10: *(a) A global TF. (b) A TF defined in the center of the image and adapted according to the scaling of local histograms.*

5.3.2 The TF field as an adaptation of one local TF

Instead of combining multiple TFs, one might want to use an adaptation (based on the similarity of histograms) of one, locally easy to define, reference TF in order to define the TF field (Figure 5.10). Assuming there is one locally suitable transfer function $\mathcal{L}(\vec{X}_R)$ and that the changes in the intensities (data properties, in general) are continuous, we can continuously adapt it at other positions.

In order to detect the intensity changes, a model for possible changes is needed. For example, we suppose that intensity variations of the material caused by an MR bias field are present. The variations caused by the multiplicative bias field (Eq. 5.8) can then be detected using the scaling-based correlation from Eq. 5.7. $B(\vec{x})$ is then equal to the scaling factor s that yields the best match between $H_{\vec{x}}$ and $H_{\vec{X}_R}$.

Knowing $B(\vec{x})$ at every position, we can use it to adapt the reference local transfer function $\mathcal{T}_{\vec{X}_R}$. We adapt the TF by scaling its domain according to

$$\mathcal{L}(\vec{x}) = \mathcal{T}_{\vec{X}_R}\left(\frac{f(\vec{x})}{B(\vec{x})}\right)$$

Such an automatic adaptation to the intensity variations is just one of the possible ways to define changes of the TF. One could, e.g., adapt the TF along a given path or consider other criteria than the histogram correlation mentioned above.

Both approaches, i.e., the combination of multiple TFs and adaptation of one TF, could be, in principle, combined. The TF field would then be constructed by combining multiple TFs that have been adapted. However, one would have to carefully define how to specify such an operation and in which situations it would be useful.

5.3.3 Local TF as a generalization of common methods

Note that the framework for local transfer functions generalizes and extends several commonly used visualization pipelines. Rendering using a *global TF* is a special case of the TF field with the same $\mathcal{T}$ at every point

$$\mathcal{L}(\vec{x}) = \mathcal{T}, \forall \vec{x}$$

Visualizing a *binary segmentation* $\mathcal{S}$ can be achieved using a TF field defined as

$$\mathcal{L}(\vec{x}) = \mathcal{T}_L \text{ where } L = \mathcal{S}(\vec{x})$$

and where $\mathcal{T}_i$ yield constant optical properties (i.e., each segment has constant color and opacity).

Visualizing a *fuzzy segmentation* can be done using a TF field

$$\mathcal{L}(\vec{x}) = \sum_{i=1}^{K} \mathcal{F}_i(\vec{x}) \cdot \mathcal{T}_i$$

where again $\mathcal{T}_i$ yield constant optical properties. It is basically an analogy of Eq. 5.1 where the weights are given by the fuzzy segmentation $\mathcal{F}_i$ (see Figure 5.7b).

A more elaborated study on visualizing fuzzy-segmented data was done by Kniss et al. [60] where the mixtures of optical properties between classes are determined using the probabilities of belonging to certain classes.

5.4 The weighted sum of local transfer functions

So far we have discussed the weights in equation 5.1. Further, we need to specify how to evaluate the weighted combination of the functions. We have been using addition and multiplication but we have not described how to do these operations. We have investigated two approaches for these operations:

5.4.1 Weighting the output of TFs

A straightforward way is to use the TF as a function and apply the common addition and scalar multiplication of the inputs

$$\mathcal{T}(\mathbf{a}) = w_1\mathcal{T}_1(\mathbf{a}) + w_2\mathcal{T}_2(\mathbf{a}),$$

where $\mathbf{a}$ is a value of the data property. This yields, e.g., for the combined opacity α and the color $\mathbf{c}$:

$$\alpha = w_1\alpha_1 + w_2\alpha_2,\ c = w_1\alpha_1\mathbf{c}_1 + w_2\alpha_2\mathbf{c}_2$$

The combination (interpolation) of colors is not a trivial problem. It is discussed in section 5.4.3.

The main advantage of this method is that it can be used independently of the shape of the TF. The transition from one TF to the other is basically achieved by making one selection more transparent and the other more opaque (see the left column of Figure 5.11). Note that one does not need to evaluate the whole TF, but only one value, i.e., $\mathcal{T}(f(\vec{X}))$. The visual result is a fading effect between TFs. This approach might be useful for illustration purposes (Correa et al. [56], Viola et al. [59]) in which several TFs are used to highlight different regions in a specific way. An example of this technique is shown in Figure 5.15.

5.4.2 Weighting parameters of the TF primitives

In some situations, such as when the TF has to adapt to the gradual changes of data properties, one would prefer to smoothly shift or scale certain selections in the TF domain. We can achieve this by using a parameterized shape model of the local TFs. A commonly used shape is the trapezoid (see Figure 5.12) because of its versatility (can represent shapes such as constant, threshold, ramp, tent) and a relatively small number of parameters.

TFs defined with trapezoids can smoothly change from one to the other by interpolating the parameters. This approach allows for gradual changes in the

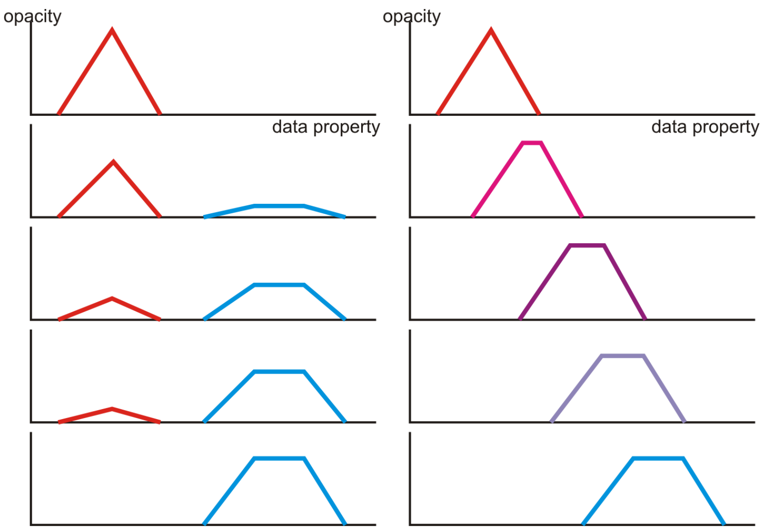


Figure 5.11: *An illustration of both interpolation methods. In the top and bottom are both transfer functions that need to be interpolated. In the left column the interpolation is done by interpolating the corresponding values. On the right the parameters of the trapezoids are interpolated.*

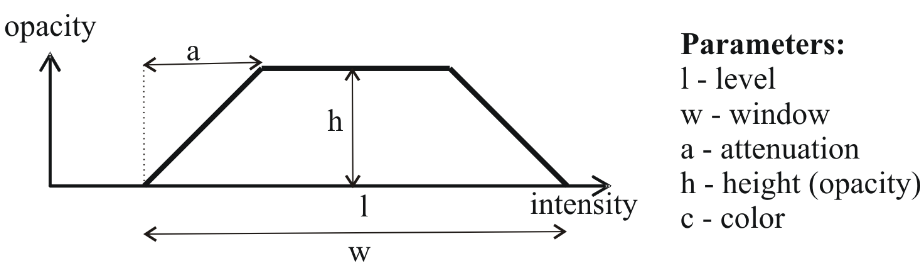


Figure 5.12: *The parameterized trapezoid primitive. Its position and shape are defined by 4 parameters. The fifth parameter is the color. The height of the trapezoid corresponds to the opacity.*

height, level and attenuation of the selected ranges. The basic assumption here is that each primitive is associated with a corresponding primitive in every other definition of the local TF. Figure 5.11 illustrates the difference between both interpolation approaches. The TF can be composed of multiple possibly overlapping trapezoids. We need to combine them in order to get the actual TF. We can look at each of these trapezoid primitives as a simplified transfer function $\mathcal{P}$ that has constant color. The combination of multiple trapezoids could be done in different ways. In our method we implement the combination as follows:

If we denote that trapezoid $\mathcal{P}_i(a)$ for intensity value a yields opacity α_i and color c_i and the resulting transfer function $\mathcal{T}(a)$ yields α and c then

- The opacity $\alpha = \max_{i=1}^{K} \alpha_i$, where K is the number of trapezoids.
- The color c of the combination will be a sum of colors of the primitives weighted by their opacity at value a
  $c = \frac{\sum_{i=1}^{K} \alpha_i c_i}{\sum_{i=1}^{K} \alpha_i}$
  Again, the color operations are not trivial and will be further discussed in the next section.

Other primitives, such as the Gaussians, could also be used. The advantage of the Gaussian would be its relatively easy extensibility to multiple dimensions.

The drawback of interpolating the primitives is a relatively slower evaluation of the interpolated TF and the need of having correspondence between primitives in different TFs. The interpolation of primitives was used in all renderings in this chapter that combine multiple LTFs, except of Figures 5.15 and 5.19, where the values were interpolated (see section 5.4.1).

5.4.3 Color interpolation

Interpolating (combining) two or more colors is not a straightforward task. Colors can be represented as vectors c in a color space. Operations such as $c = \sum w_i c_i$, where $\sum w_i = 1$ are weighting factors, can then be performed by simple vector operations. The interpolated color c is then located between the colors c_i in the color space. It is desirable, however, that the interpolated color is also perceived by the user as "a color between the colors c_i". Such a judgement might be user and task dependent. However, rather then using the traditional RGB color system, we can interpolate the color in the CIE Lab color system. The CIE Lab color space has been constructed as a perceptually

Figure 5.13: *Interpolating colors in different color systems yields visible differences. On the left the RGB system was used. On the right the interpolation was done in the CIE Lab system.*

uniform system. In other words, the spatial relation between colors in that system is related to the way we humans perceive them. In our implementation, we have included the option of using both the RGB and CIE LAB systems.

5.5 Defining local TFs

The local TF can be defined similarly to the global TF. The main difference is that only a local region is taken into account when investigating data properties and evaluating the quality of the TF. It is sufficient that the TF performs well in one part of the volume. In the part where it does not perform well a different TF can be defined. This makes the definition of local TFs, in principle, easier as long as not too many definitions are needed.

There are numerous ways to define TFs (see, e.g., Chapter 4 or [12, 18, 27, 28, 29, 36, 39]). We have used a manual definition for local TFs based on the local histogram of intensities in order to guide the user. The peaks of the histogram are assumed to give a visual hint on the scalar ranges of tissues. In local regions the intensities of tissues usually have a smaller variance and therefore the peaks in local histograms are clearer. The user first picks a reference point $\vec{X}_R$ in the 3D volume. The reference point is the center of a cubic sub-volume on which the local histogram is constructed. We use trapezoid primitives to define the TF (see Figures 5.12 and 5.14). The user can place the trapezoid over a certain range of values and adjust its parameters, including the color. If the TF does not perform well in all regions (as seen in Figure 5.14a), then the user can define different LTFs, typically in those regions where the result is the worst (see Figure 5.14b). If the histogram-based adaptation is used, the size of the neighborhood should be set large enough to always contain those representative materials that describe the behavior of the data properties.

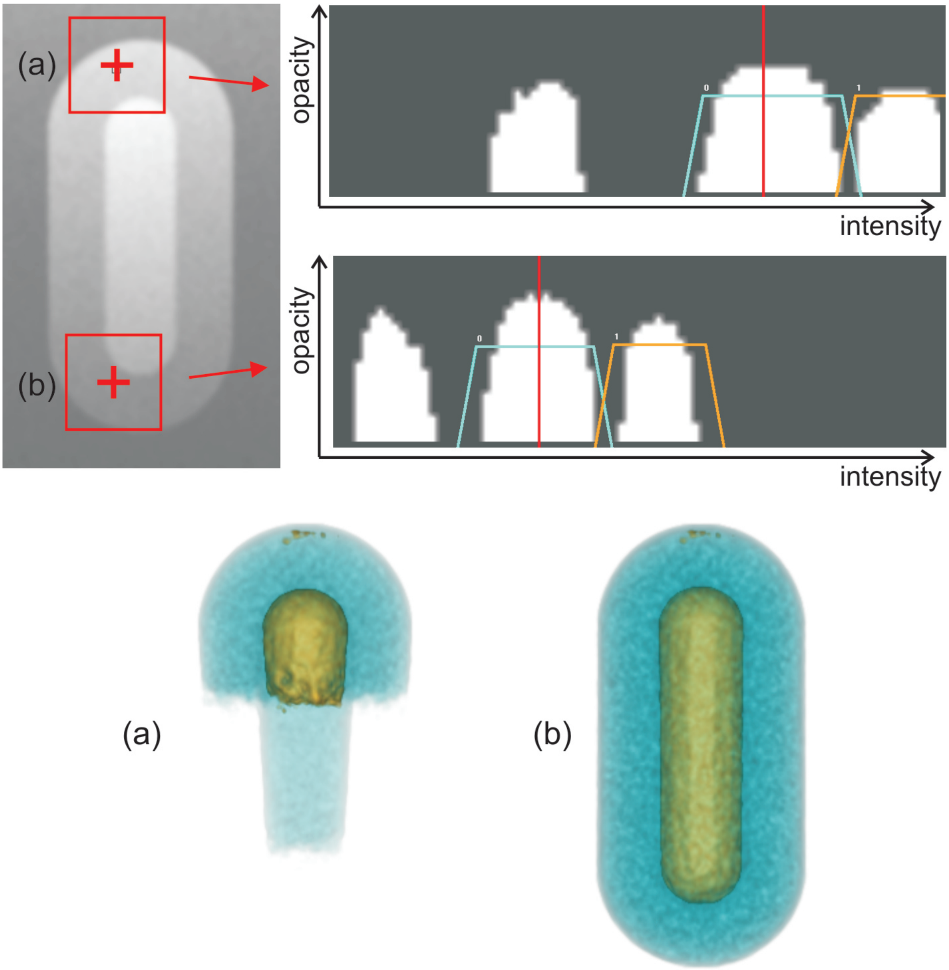


Figure 5.14: *User interaction by defining local transfer functions based on trapezoids. In (a) a TF is defined that visualizes the top of the volume. In (b) a second LTF is added in the bottom of the volume (i.e., in the region where the first TF failed). The similarity of local histograms is used to combine both LTFs. In this example a neighborhood of size* 50^3 *is used, yielding clear peaks in the local histograms.*

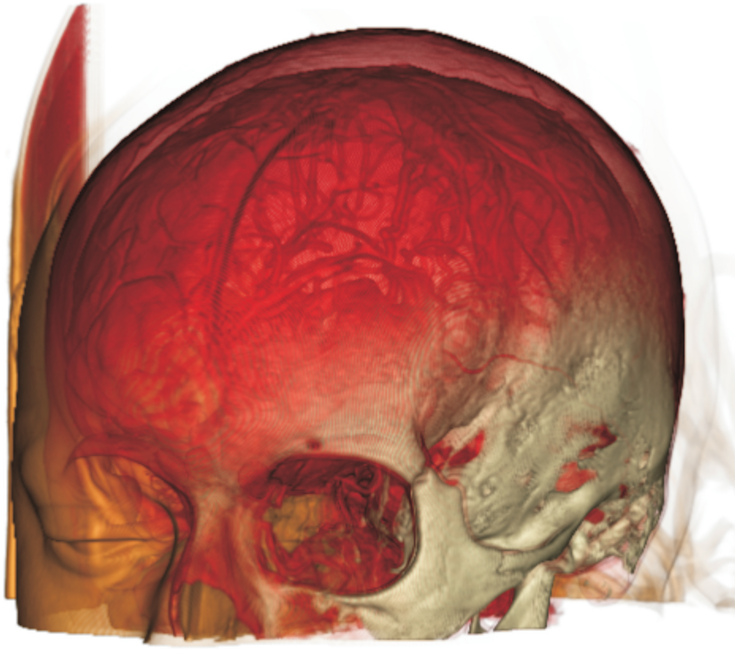


Figure 5.15: *CT dataset of a head (512x512x286). Combination of three LTFs with reference points positioned on both sides of the head and in the top part. The TFs are set in order to show the skin, the bone and to reveal interior vessel structures respectively. The combination is based on the distance to the reference points and on the interpolation of values.*

5.6 Results and discussion

By using several implementations of the local transfer functions, we illustrate the advantages of this framework.

In Figure 5.15 different TFs were designed on both sides of the head and in the top part. The first highlights the bone, the second the skin and the third makes the the bone and skin semi-transparent in order to reveal the underlying vessels. In this case interpolating values is more appropriate since, as the influence decreases, it shows the tissues as semi-transparent. This example illustrates that the concept of local TFs could be used for the purpose of illustration [56] or focus-context visualizations [58, 59].

Figure 5.16 illustrates the advantage of using multiple LTFs. In the low-quality CT image of a foot, it is not possible to show the bones opaque without showing the background tissue (see Figure 5.16a,b). The thin bone in the toes

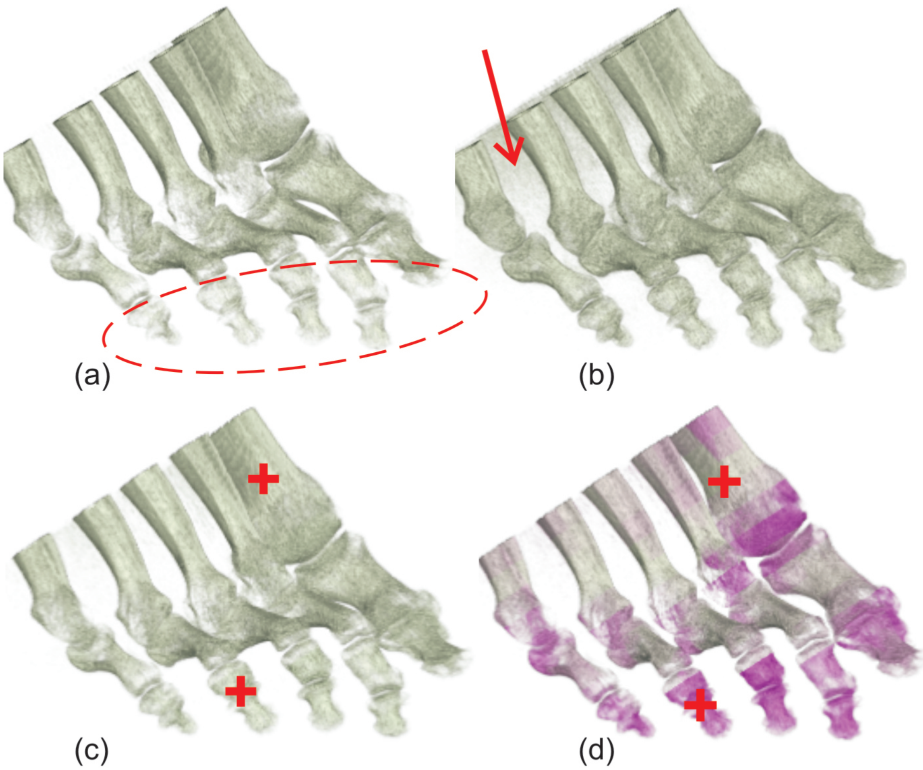


Figure 5.16: *A noisy CT dataset of a foot* (256^3)*. In (a) and (b) two different global TFs are depicted. (c) shows two LTFs combined using the similarity of local histograms with a sub-volume of* 50^3*. For illustration, (d) shows the influence region from (c) in a different color.*

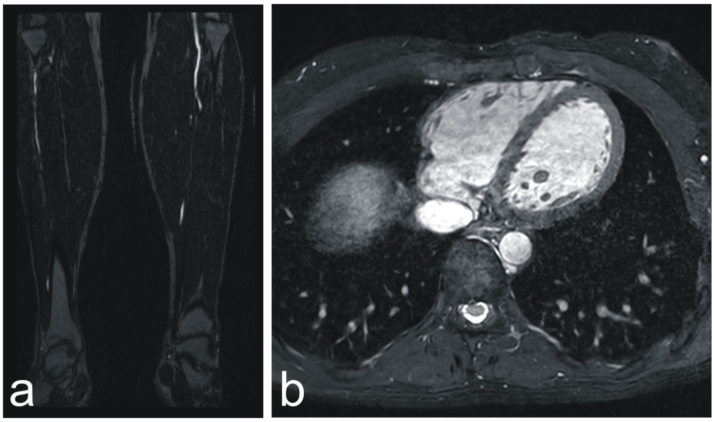


Figure 5.17: *2D slices of the datasets from (a) Figure 5.18, (b) Figure 5.19.*

and the softer tissue in the joints have lower intensity which resembles that of the muscle tissue marked by the arrow in Figure 5.16b. In Figure 5.16c two LTFs are defined and combined by the weighting based on histogram similarity. The two LTFs were positioned at the positions of the crosses: one in the region with hard bone and background tissue, the other in a region with soft bone. Figure 5.16d shows the influence from Figure 5.16c generated by the similarity of local histograms. We show the influence by assigning another color to the bone in the second LTF. It can be seen that the histogram similarity also managed to detect the joints as regions with low intensity of bone. The histogram similarity makes the method less sensitive to the placement of the reference points. Similar results to Figure 5.16c would be obtained if the reference points were placed in other hard/soft bone regions.

Figures 5.17a and 5.18 show a data slice and renderings of an MR dataset of human legs. The vessels contain contrast agent. Although some bias corrections were applied as part of the scanning protocol, it is visible, e.g., looking at the global TF in Figure 5.18a, that the upper right part is brighter. The intensity values are hampered by both bias and decay of contrast agent concentration. There is a trade-off between showing only vessels in the top right and still showing the less bright vessels in the bottom. Using two TFs already improves the visualization (Figure 5.18b). Figures 5.18c and 5.18d compare the spatial distance and histogram similarity based influences. Red and purple denote the amount of influence received from the reference points. Note that the histogram similarity automatically reveals the horizontal inhomogeneities as well. Figure 5.18e shows the results of one TF (defined in the location of the cross) that adapts to the data. The size of the neighborhood should be

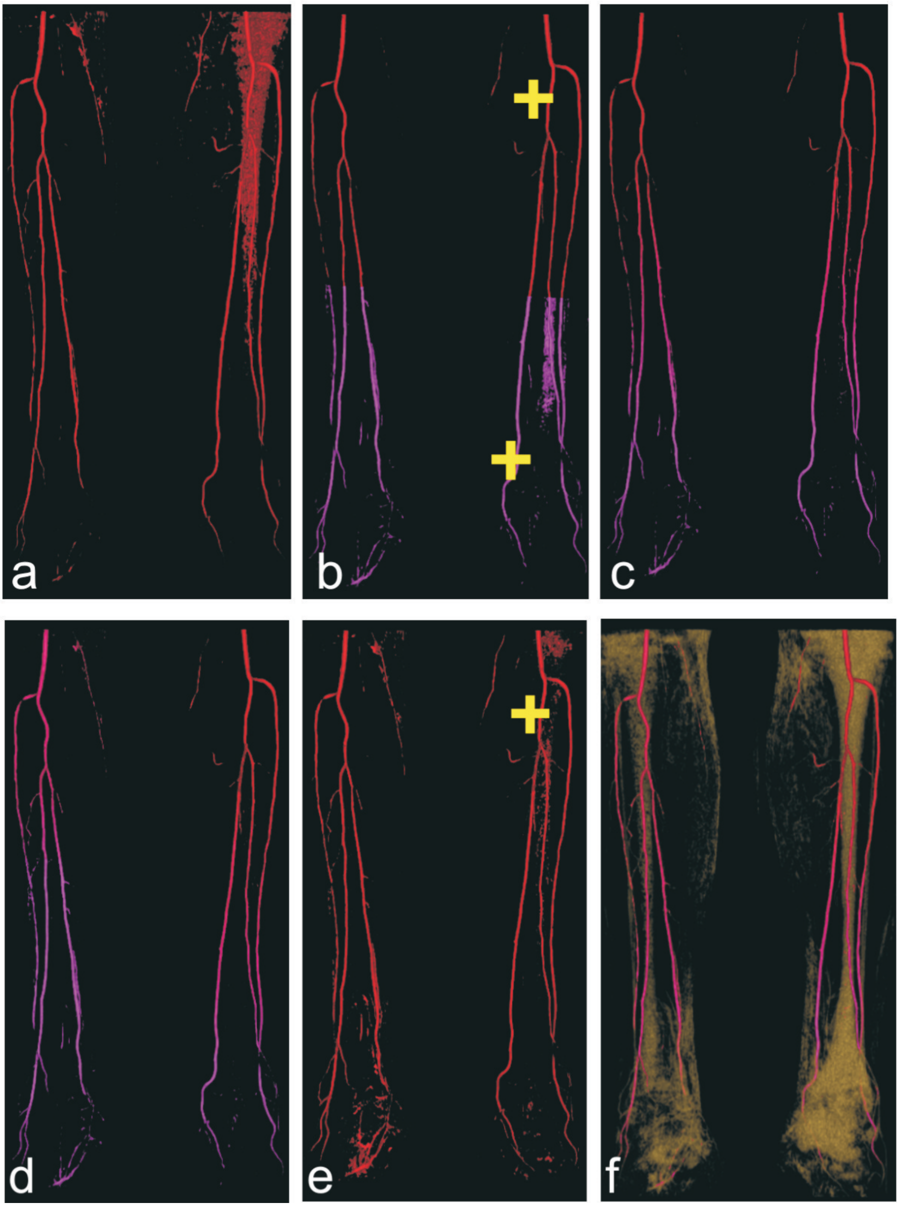


Figure 5.18: *An MR dataset (512x512x140) of human legs with contrast media in the vessels. (a) A global TF. (b) Two TFs with nearest neighbor. (c) Distance and (d) local histogram similarity correlation with neighborhood size 150. (e) Scaling adaptation of one TF based on optimal scaling of local histograms with neighborhood size 250. (f) Surrounding tissue can be added in order to show the context of the vessels.*

large enough to contain all representative tissues that would reveal the intensity changes in the neighborhood. On the other hand a too large neighborhood would average over a too large area and would not detect local changes. The selection of surrounding tissue in Figure 5.18f is not modified through the data.

Figures 5.17b and 5.19 show a data slice and renderings of an MR cardiac dataset. We used a model-based segmentation (Figure 5.19a) [55] in order to determine the influences of TFs. The myocardium segment (red) was assigned a TF highlighting the arteries. Further, a segmented neighborhood around the heart was assigned a TF showing arteries as a context. Note that fuzzy borders between the TFs (Figure 5.19d) can code the uncertainty of the segmentation and the context information may reveal possible segmentation errors.

The concept of locally modified TFs is not standard in volume visualization. Therefore we had to implement our own rendering pipeline in which the TFs are interpolated at every point of the volume. The goal of this chapter is to present a concept and the implementation was used to proof the usefulness of this concept. Therefore we opted for a flexible software solution. In order to enable interactive rendering we first pre-classify the volume and then render the RGBA volume. The drawback, however, of this approach is a reduced image quality and the lack of interactivity in modifying the transfer functions. The pre-classification for the rendering in Figure 5.15 took around 60 seconds. In case of interpolation based on primitives the time would be approximately double since the parameters of the primitives need to be interpolated. One possible solution would be to implement an optimized GPU-based pipeline for local TFs where the TFs are interactively interpolated at every sample point.

5.7 Conclusions and future work

We have shown that the concept of local transfer functions can be used to adapt to locally modified data parameters or user requirements. In situations where the data properties change in different parts of the volume, one can define locally good transfer functions. This is easier than tuning a global TF and helps in situations when no good global TF exists.

In general, the LTFs are suited for slow gradual changes of the TF through the dataset. Such a behavior of the TF field could often be achieved by using only few reference points with either distance or histogram-based weighting. However, in situations when the LTFs would have to adapt to structures changing their properties fast (such as identical neighboring voxels belonging to different objects), the use of LTFs becomes cumbersome.

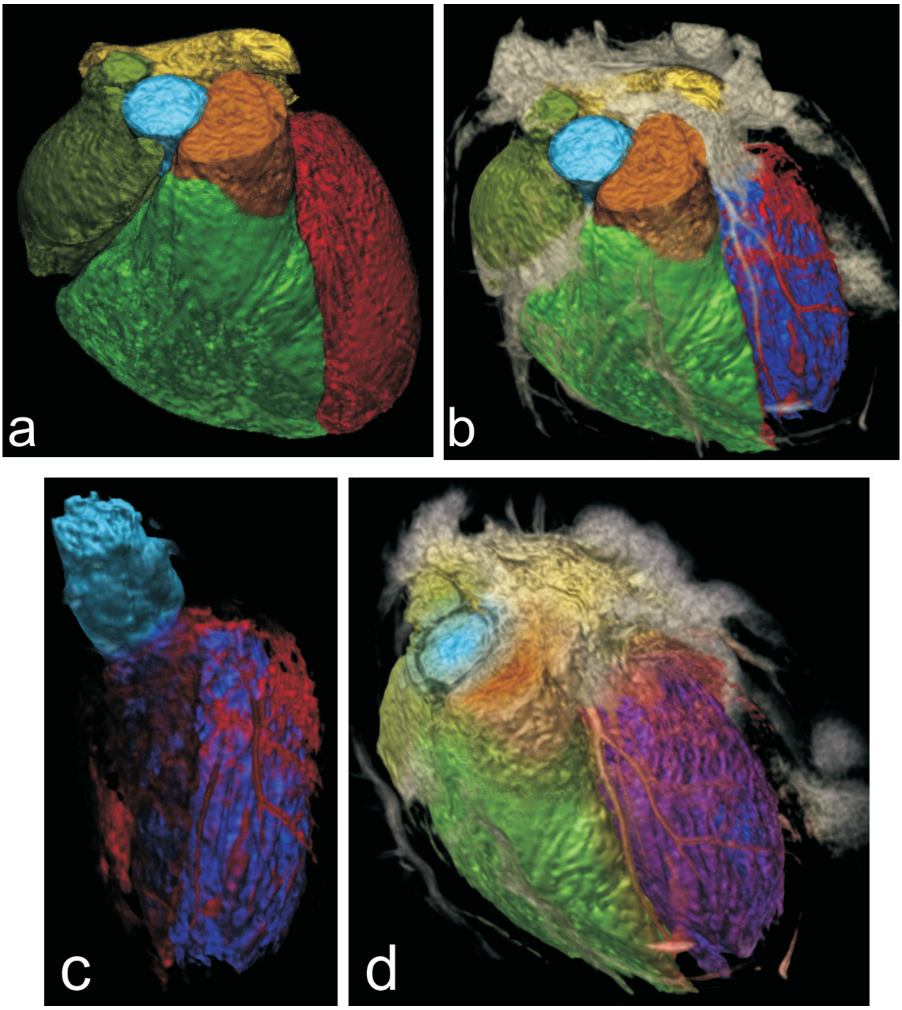


Figure 5.19: *An MR cardiac dataset (512x512x150). (a) A model-based segmentation. (b) Arteries in the myocardium (red) and in the neighborhood around the heart (white). (c) Arteries inside the myocardium segmentation (red), left ventricle (blue) and aorta (light blue). (d) Fuzzy borders between segments with interpolated transfer functions.*

We have illustrated several approaches to define the TF field, such as adapting the TF or combining existing TFs based on distance measures or model-based segmentation.

The combination of equations 5.2 and, e.g., 5.3 represents the so-called Shepard method (Amidror [57]) of weighting scattered points based on the distance. This method takes into account all reference points. Although weighting all points might be desired in some situations, one can think of using, e.g., the Delaunay triangulation in order to restrict the number of reference points taken into account. The triangulation (tetrahedralization in the 3D case) could be based either on the spatial distance or adapted to the space of histogram similarities. In general, scattered interpolation is a large research topic by itself [61].

What we presented in this chapter is basically a proof of concept. One could think of optimizing several parts of the pipeline. The TF field could be, e.g., optimized by the graph-based approach by Kniss et al. [60] that used graphs to reduce the dimensionality of storing the classification result.

For the sake of simplicity we have shown the local TFs for the case of 1D TFs. The state of the art are, however, multidimensional TFs (see Chapter 3 or [18, 62] that may yield a better image quality and allow for advanced classification. These can be as well combined and adapted using our framework.

Local transfer functions seem as a promising tool that might help to improve the visualization of data (e.g., MR) which are very difficult to visualize with global TFs.

Summary and Conclusions

Facilitating the Design of Multidimensional and Local Transfer Functions for Volume Visualization

Summary

The importance of volume visualization is increasing since the sizes of the datasets that need to be inspected grow with every new version of medical scanners (e.g., CT and MR). Direct volume rendering is a 3D visualization technique that has, in many cases, clear benefits over 2D views. It is able to show 3D information, facilitating mental reconstruction of the 3D shape of objects and their spatial relation. The complexity of the settings required in order to generate a 3D rendering is, however, one of the main reasons for this technique not being used more widely in practice. Transfer functions play an important role in the appearance of volume rendered images by determining the optical properties of each piece of the data. The transfer function determines what will be seen and how. The goal of the project on which this PhD thesis reports was to develop and investigate new approaches that would facilitate the setting of transfer functions.

As shown in the state of the art overview in Chapter 2, there are two main aspects that influence the effectiveness of a TF: the choice of the TF domain and the process of defining the shape of the TF. The choice of a TF domain, i.e., the choice of the data properties used, directly determines which aspects of the volume data can be visualized. In many approaches, special attention is given to TF domains that would enable an easier selection and visualization of boundaries between materials. The boundaries are an important aspect of the volume data since they reveal the shapes and sizes of objects. Our research in improving the TF definition focused on introducing new user interaction methods and automation techniques that shield the user from the complex process of manually defining the shape and color properties of TFs. Our research dealt with both the TF domain and the TF definition since they are closely related. A suitable TF domain cannot only greatly improve the manual definition, but also, more importantly, increases the possibilities of using automated techniques.

Chapter 3 presents a new TF domain. We have used the LH space and the associated LH histogram for TFs based on material boundaries. We showed that the LH space reduces the ambiguity when selecting boundaries compared to the commonly used space of the data value and gradient magnitude. Fur-

thermore, boundaries appear as blobs in the LH histogram that make them easier to select. Its compactness and easier selectivity of the boundaries makes the LH histogram suitable for the introduction of clustering-based automation. The mirrored extension of the LH space differentiates between both sides of the boundary. The mirrored LH histogram shows interesting properties of this space, allowing the selection of all boundaries belonging to one material in an easy way. We have also shown that segmentation techniques, such as region growing methods, can benefit from the properties of LH space. Standard cost functions based on the data value and/or the gradient magnitude may experience problems at the boundaries due to the partial volume effect. However, our cost function that is based on the LH space is, however, capable of handling the region growing of boundaries better.

Chapter 4 presents an interaction framework for the TF definition based on hierarchical clustering of material boundaries. Our framework aims at an easy combination of various similarity measures that reflect requirements of the user. One of the main benefits of the framework is the absence of similarity-weighting coefficients that are usually hard to define. Further, the framework enables the user to visualize objects that may exist at different levels of the hierarchy. We also introduced two similarity measures that illustrate the functionality of the framework. The main contribution is the first similarity measure that takes advantage of properties of the LH histogram from Chapter 3. We assumed that the shapes of the peaks in the LH histogram can guide the grouping of clusters. The second similarity measure is based on the spatial relationships of clusters.

In Chapter 5, we presented part of our research that focused on one of the main issues encountered in the TFs in general. Standard TFs, as they are applied everywhere in the volume in the same way, become difficult to use when the data properties (measurements) of the same material vary over the volume, for example, due to the acquisition inaccuracies. We address this problem by introducing the concept and framework of local transfer functions (LTFs). Local transfer functions are based on using locally applicable TFs in cases where it might be difficult or impossible to define a globally applicable TF. We discussed a number of reasons that hamper the global TF and illustrated how the LTFs may help to alleviate these problems. We have also discussed how multiple TFs can be combined and automatically adapted. One of our contributions is the use of the similarity of local histograms and their correlation for the combination and adaptation of LTFs.

Conclusions and future work

In addition to the conclusions made at the end of each chapter, we can draw some general conclusions. The use of the LH space and LH histograms for TFs is the main contribution of this thesis. We showed that the LH space has clear advantages compared to the TF domains used in the boundary-oriented state of the art approaches. The automation based on the hierarchical clustering framework proved to be capable of shielding the user from the TF domain, yet allowing manual adjustments of the TF.

The idea of local transfer functions seems very promising. Although we have tried to keep our approach as general as possible, it proved difficult to grasp all possible phenomena that could hamper the global effect of transfer functions and handle them with one general approach of local TFs. Our work has shown some of the possibilities of LTFs and pointed out the issues that need to be considered for their adoption. For their practical use, the freedom of possibilities offered by the concept of LTFs would need to be restricted. Each application deals with a certain specific type of data and encounters only a limited set of problems, therefore the LTF method could be very well tuned to those problems. For example, an application focused on visualization of thin vessels may not profit from a method that attempts to detect global bias in order to adapt the TF. Furthermore, if the typically problematic areas in the dataset are known beforehand, the method can be adapted while taking that knowledge into account.

The presented idea of TFs that use the LH histogram, including the clustering method, is based on the assumption that the boundaries in the datasets can be clearly differentiated by looking at the combination of f_L and f_H. This assumption might, however, not be valid in case of noisy and/or biased datasets such as MR or low-dose CT. One possible approach may be to use the concept of LTFs for the LH methods developed in Chapters 3 and 4. Both the manual and clustering techniques could be used on local LH histograms where the peaks are clearer than on the global histogram. In general, various enhancing / denoising techniques might be used as preprocessing to our methods in order to improve the clarity of the histograms.

The work presented in this thesis showed that good choices in the TF domains and using design methods based on analyzing the data can help to introduce different levels of abstraction into the user interface. In general, a promising future for TF design lies in using state of the art image analysis techniques that can help to translate commands that are easy to understand for the user into visualization settings.

Bibliography

[1] W. E. Lorensen and H. E. Cline, "Marching cubes: a high–resolution 3D surface construction algorithm," *Computer Graphics*, vol. 21, no. 3, pp. 163–169, 1987.

[2] M. Levoy, "Display of surfaces from volume data," *IEEE Computer Graphics and Applications*, vol. 8, no. 3, pp. 29–37, 1988.

[3] L. Westover, "Interactive volume rendering," *Proceedings Chapel Hill Workshop Volume Visualization*, pp. 9–16, 1989.

[4] L. Westover, "Footprint evaluation for volume rendering," *Computer Graphics, Proceedings SIGGRAPH 90*, pp. 367–376, 1990.

[5] M. Meissner, U. Hoffmann, and W. Strasser, "Enabling classification and shading for 3D texture mapping based volume rendering using opengl and extensions," *Proceedings IEEE Visualization*, pp. 207–214, 1999.

[6] B. Cabral, N. Cam, and J. Foran, "Accelerated volume rendering and tomographic reconstruction using texture mapping hardware," *Proceedings Symposium on Volume Visualization*, p. 9198, 1994.

[7] N. Max, "Optical models for direct volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 2, pp. 99–108, 1995.

[8] T. B. Phong, "Illumination for computer generated images," *Communications of the ACM*, vol. 18, pp. 311–317, 1975.

[9] W. Heidrich, M. McCool, and J. Stevens, "Interactive maximum projection volume rendering," *Proceedings IEEE Visualization*, pp. 11–18, 1995.

[10] K. Zuiderveld, "Visualization of multimodality medical volume data using object-oriented methods," *PhD thesis, Universiteit Utrecht, Netherlands*, 1995.

[11] Y. Sato, N. Shiraga, S. Nakajima, S. Tamura, and R. Kikinis, "LMIP: Local maximum intensity projection - a new rendering method for vascular visualization," *Journal of Computer Assisted Tomography*, vol. 22, no. 6, pp. 912–917, 1998.

[12] A. Kaufman and K. Mueller, "Overview of volume rendering," *chapter in The Visualization Handbook, Elsevier*, 2005.

[13] K. Engel, M. Hadwiger, J. M. Kniss, C. Rezk-Salama, and D. Weiskop, "Real-time volume graphics," *AK Peters*, 2006.

[14] M. S. Peercy, "Linear color representations for full speed spectral rendering," *Proceedings International Conference on Computer Graphics and Interactive Techniques*, pp. 191–198, 1993.

[15] S. Roettger, M. Bauer, and M. Stamminger, "Spatialized transfer functions," *Proceedings IEEE/EuroGraphics Symposium on Visualization*, pp. 271–278, 2005.

[16] J. B. A. Maintz and M. A. Viergever., "Survey of medical image registration," *Medical Image Analysis*, vol. 2, no. 1, pp. 1–36, 1998.

[17] E. L. Nickoloff and R. Riley, "A simplified approach for modulation transfer function determinations in computed tomography," *Medical Physics*, vol. 12, no. 4, pp. 437–442, 1985.

[18] G. Kindlmann and J. W. Durkin, "Semi-automatic generation of transfer functions for direct volume rendering," *Proceedings IEEE Symposium on Volume Visualization*, pp. 79–86, 1998.

[19] J. Kniss, G. Kindlmann, and C. Hansen, "Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets," *Proceedings IEEE Visualization*, pp. 255–262, 2001.

[20] J. Kniss, G. Kindlmann, and C. Hansen, "Multidimensional transfer functions for interactive volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 3, pp. 270–285, 2002.

[21] E. B. Lum and K. L. Ma, "Lighting transfer functions using gradient aligned sampling," *Proceedings IEEE Visualization*, pp. 289–296, 2004.

[22] B. M. ter Haar Romeny, "Front-end vision and multi-scale image analysis," *Kluwer Academic Publisher*, 2003.

[23] J. Hladůvka, A. König, and E. Gröller, "Curvature-based transfer functions for direct volume rendering," *Proceedings Spring Conference on Computer Graphics*, vol. 16, pp. 58–65, 2000.

[24] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Möller, "Curvature-based transfer functions for direct volume rendering: Methods and applications," *Proceedings IEEE Visualization*, pp. 513–520, October 2003.

[25] H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L. S. Avila, K. Martin, R. Machiraju, and J. Lee, "The transfer function bake-off," *IEEE Computer Graphics and Applications*, vol. 21, no. 3, pp. 16–22, 2001.

[26] C. Lundström, A. Ynnerman, P. Ljung, A. Persson, and H. Knutsson, "The α-histogram: Using spatial coherence to enhance histograms and transfer function design," *Proceedings IEEE/EuroGraphics Symposium on Visualization*, pp. 227–234, 2006.

[27] C. L. Bajaj, V. Pascucci, and D. Schikore, "The contour spectrum," *Proceedings IEEE Visualization*, pp. 167–174, 1997.

[28] V. Pekar, R. Wiemker, and D. Hempel, "Fast detection of meaningful isosurfaces for volume data visualization," *Proceedings IEEE Visualization*, pp. 223–230, 2001.

[29] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber, "Design galleries: a general approach to setting parameters for computer graphics and animation," *Proceedings SIGGRAPH Conference*, pp. 389–400, 1997.

[30] A. König and E. Gröller, "Mastering transfer function specification by using VolumePro technology," *Proceedings Spring Conference on Computer Graphics*, vol. 17, pp. 279–286, 2001.

[31] I. Fujishiro, T. Azuma, and Y. Takeshima, "Automating transfer function design for comprehensible volume rendering based on 3D field topology analysis," *Proceedings IEEE Visualization*, pp. 467–470, 1999.

[32] C. Lundström, P. Ljung, and A. Ynnerman, "Extending and simplifying transfer function design in medical volume rendering using local histograms," *Proceedings IEEE/EuroGraphics Symposium on Visualization*, pp. 263–270, 2005.

[33] C. Lundström, P. Ljung, and A. Ynnerman, "Local histograms for design of transfer functions in direct volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 6, pp. 1570–1579, 2006.

[34] D. H. Laidlaw, K. W. Fleischer, and A. H. Barr, "Partial-volume bayesian classification of material mixtures in MR volume data using voxel histograms," *IEEE Transactions on Medical Imaging*, vol. 17, no. 1, pp. 74–86, 1998.

[35] F. Y. Tzeng, E. B. Lum, and K. L. Ma, "A novel interface for higher-dimensional classification of volume data," *Proceedings IEEE Visualization*, pp. 505–512, 2003.

[36] F. Y. Tzeng and K. L. Ma, "A cluster-space visual interface for arbitrary dimensional classification of volume data," *Proceedings Eurographics/IEEE TCVG Visualization Symposium (VisSym)*, pp. 17–24, 2004.

[37] F. Y. Tzeng, E. B. Lum, and K. L. Ma, "An intelligent system approach to higher-dimensional classification of volume data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 3, pp. 273–284, 2005.

[38] R. Huang and K. L. Ma, "RGVis: Region growing based visualization techniques for volume visualization," *Proceedings Pacific Graphics Conference*, pp. 355–363, 2003.

[39] T. He, L. Hong, A. Kaufman, and H. Pfister, "Generation of transfer functions with stochastic search techniques," *Proceedings IEEE Visualization*, pp. 227–234, 1996.

[40] H. Pfister, "Hardware-accelerated volume rendering," *chapter in The Visualization Handbook, Elsevier*, 2005.

[41] K. Engel, M. Kraus, and T. Ertl, "High-quality pre-integrated volume rendering using hardware-accelerated pixel shading," *Eurographics / SIGGRAPH Workshop on Graphics Hardware01*, p. 916, 2001.

[42] M. Kraus and T. Ertl, "Pre-integrated volume rendering," *chapter in The Visualization Handbook, Elsevier*, 2005.

[43] I. W. O. Serlie, R. Truyen, J. Florie, F. Post, L. J. van Vliet, and F. M. Vos, "Computed cleansing for virtual colonoscopy using a three-material transition model," *Medical Image Computing and Computer-Assisted Intervention - MICCAI Proceedings, Part 2*, vol. 2897, pp. 175–183, 2003.

[44] H. W. Shen, C. Hansen, Y. Livnat, and C. R. Johnson, "Isosurfacing in Span Space with utmost efficiency (ISSUE)," *Proceedings IEEE Visualization*, pp. 287–294, 1996.

[45] M. T. Vlaardingerbroek and J. A. D. Boer, "Magnetic resonance imaging: Theory and practice," *Springer Verlag*, 1999.

[46] H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler, "The VolumePro real-time ray-casting system," *Proceedings 26th annual conference on Computer graphics and interactive techniques*, pp. 251–260, 1999.

[47] P. W. Verbeek and L. J. van Vliet, "On the location error of curved edges in low-pass filtered 2-D and 3-D images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 7, pp. 726–733, 1994.

[48] H. Bouma, A. Vilanova, L. J. van Vliet, and F. A. Gerritsen, "Correction for the dislocation of curved surfaces caused by the PSF in 2D and 3D CT images," *IEEE Transactions of Pattern Analysis and Machine Intelligence*, vol. 27, no. 9, pp. 1501–1507, 2005.

[49] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern classification," *2nd edition. Wiley. New York*, 2001.

[50] U. Vovk, F. Pernuš, and B. Likar, "MRI intensity inhomogeneity correction by combining intensity and spatial information," *Phys Med Biol*, vol. 49, pp. 4119–4133, 2004.

[51] M. Holtzman-Gazit, D. Goldsher, and R. Kimmel, "Hierarchical segmentation of thin structures in volumetric medical images," *MICCAI (2), Lecture Notes in Computer Science, Springer*, vol. 2879, pp. 562–569, 2003.

[52] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," *Lecture Notes in Computer Science*, vol. 1496, pp. 130–137, 1998.

[53] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.

[54] J. Rajapakse and F. Fruggel, "Segmentation of MR images with intensity inhomogeneities," *Image and Vision Computing*, vol. 16, no. 3, pp. 165–180, 1998.

[55] O. Ecabert, J. Peters, and J. Weese, "Modeling shape variability for full heart segmentation in cardiac computed-tomography images," *Medical Imaging 2006: Image Processing*, vol. 6144, pp. 1199–1210, 2006.

[56] C. D. Correa and D. Silver, "Dataset traversal with motion-controlled transfer functions," *Proceedings IEEE Visualization*, pp. 359 – 366, 2005.

[57] I. Admidror, "Scattered data interpolation methods for electronic imaging systems: a survey," *Journal of Electronic Imaging*, vol. 11, no. 2, pp. 157–176, 2002.

[58] J. Zhou, M. Hinz, and K. Tonnies, "Focal region-guided feature-based volume rendering," *Proceedings 1st International Symposium on 3D Data Processing Visualization and Transmission*, pp. 87–90, 2002.

[59] I. Viola, A. Kanitsar, and E. Gröller, "Importance-driven volume rendering," *Proceedings IEEE Visualization*, pp. 139–145, 2004.

[60] J. M. Kniss, R. V. Uitert, A. Stevens, G. S. Li, T. Tasdizen, and C. Hansen, "Statistically quantitative volume rendering," *Proceedings IEEE Visualization*, pp. 287–294, 2005.

[61] R. Franke and G. M. Nielson, "Scattered data interpolation and applications- a tutorial and survey," *Geometric Modelling: Methods and Their Application*, pp. 131–160, 1991.

[62] J. Kniss, P. McCormick, A. McPherson, J. Ahrens, J. Painter, A. Keahey, and C. Hansen, "Interactive texture-based volume rendering for large data sets," *IEEE Computer Graphics and Applications*, vol. 21, no. 4, pp. 52–61, 2001.

Publications

Refereed journal publications

P. Šereda, A. Vilanova, I. W. O. Serlie, and F. A. Gerritsen, "Visualization of Boundaries in Volumetric Datasets Using LH Histograms," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 2, pp. 208-218, 2006.

C. A. Guerrero Sanchez, T. Erdmenger, **P. Šereda**, D. Wouters, and U. S. Schubert, "Water-soluble ionic liquids as novel stabilizers in suspension polymerizations: engineering polymer beads," *Chemistry - A European Journal*, vol. 12, no. 35, pp. 9036-9045, 2006.

Refereed conference publications

P. Šereda, A. Vilanova, and F. A. Gerritsen, "Automating Transfer Function Design for Volume Rendering Using Hierarchical Clustering of Material Boundaries," *Proceedings IEEE/EuroGraphics Symposium on Visualization (EuroVis)*, pp. 243250, 2006.

P. Šereda, A. Vilanova, and F. A. Gerritsen, "Automating Transfer Function Design for Volume Rendering Using Hierarchical Clustering of Material Boundaries," *Proceedings 12th ASCI conference*, pp. 187-194, 2006.

P. Šereda, A. Vilanova ,and F. A. Gerritsen, "Mirrored LH Histograms for the Visualization of Material Boundaries," *Proceedings Visualization Modeling and Vision (VMV)*, pp. 237-244, 2006.

P. Šereda, A. Vilanova, S. Lobregt, and F. A. Gerritsen, "Local Transfer Functions," *Submitted to IEEE Visualization 2007*.

Acknowledgements

First of all, I would like to thank my direct supervisors Anna Vilanova i Bartrolí and Frans Gerritsen. Anna, it was great to work in your group. I liked our meetings, they helped me a lot in the research. I really appreciate that you would always find some time for discussion, solving problems or reviewing our paper, and all that with an infinite patience and friendly attitude. Frans, thanks for directing me in my research and helping me to realize that research and applications are two different worlds. Also thanks for introducing me to the group at Philips Medical Systems and pointing out people that could help me with my project. I also want to thank to Bart ter Haar Romeny, especially for his great scientific enthusiasm which, easily spreading among students, was a good source of motivation. Also, thanks Bart for the great social events!

Second, big thanks to all who contributed to this thesis with their comments. Apart from Anna, Frans and Bart it was Frits Post, Jack van Wijk, Meister Eduard Gröller and Steven Lobregt.

Then, I want to thank to all my TU/e colleagues during my PhD. Henri, thanks for numerous discussions and technical help, Tibor for many technically-philosophical discussions (so, do you already know why do mirrors switch left and right sides, but not up and down?). Thanks to the "A-team" for great times both inside and outside of the university: Paulo who becomes Paula during carnival to accompany Petra, Tim who dresses as an indian and claims it is a hippie, and the gaze Vesna. Hans, thanks a lot for your help with the template of this thesis! Thanks to all my other colleagues: Arjen who likes climbing and bananas a lot (but, from which side is it the best to open a banana?), Neda, Ralph, Alessandro, Remco, Erik, Frans K., Bram, Luc, Thorsten, Markus, Bart J., Evgeniya who from time to time changes the spelling of her name just to confuse us, Marieke, Laura and all the master students for their help and support. Especially thanks to The Matrin, for many productive evenings in the lab playing chess over few beers. Also thanks to Marijke for a cooperation on the clustering of the LH histogram. I cannot forget to thank to Margret for her great support in all the paperwork. Besides the people form our group, I want to thank to Larry Graaf and Jaap Jansen for contributing to the cover of this thesis by helping me to get an MR scan of my head.

Outside of TU/e, I want to thank to Philips Medical Systems for the financial support of my project, as well as the people from PMS: Steven and Roel for joining our meetings and their useful contributions to the discussions, Kees for a great help with EasyScil and technical support that I especially needed at the beginning of my project, Hubrecht for the VolumePro support, and other members of the group for their help. People from TU Delf, especially Frits, Iwo, and Charl, thanks a lot for our fruitful discussions. I cannot forget to mention people from Vienna. Edi, Matej, Ivan, Stefan, Alexandra, Markus and others, thanks guys for making my stay in Vienna so pleasant. Seeing your work and listening to your ideas helped me a lot.

In the non-scientific world, I want to, first of all, thank to my family. Actually, to both families: the Czech one and the group of friends who became my Dutch family. Especially thanks to both my moms (the Czech and the Serbian who adopted me in Holland with her warm hug). Thanks to all my friends for making the life enjoyable: dr. Yohan from the real center of France and Anabel - la chica de Valencia (it's true eh?) for their friendships and great enthusiasm in organizing parties and other activities. Francisco (el chico) and Laurène (la petite cerise), thanks to both for pulling me into the salsa world, I am enjoying it! :) dr.PD Olavio, dr.bic. Srdjan, Przemek, Francesca, Erik-Jan, Fabiola, Alberto from Monza, Bertrand, Dirk, Daniel, Vincent, Elodie who does not like to be touched, Séverine, met Ton, Hugo, The Martin and Natalia, and many others. I would also like to thank to our friends Martin I. and H. Gaarden, who became very popular and were present at almost every occasion. Special thanks to my friends and house mates at OUR place during last years (in chronological order): zammel zebi Issam, pinche Mexicano Carlito, mom Tamara, "andílek" Anka, and Amandine.

Curriculum vitae

Petr Šereda was born in Plzeň, Czech Republic, in 1979. He graduated from the grammar school in Plzeň in 1997. In 2002, he received his Ing. (M.Sc.) degree in computer graphics at the Faculty of Applied Sciences of the University of West Bohemia in Plzeň. His graduation project involved speed optimizations for ray-casting of volume data.

In December 2002, as a PhD student, he joined the Biomedical Image Analysis and Interpretation group at the Faculty of Biomedical Engineering of the Eindhoven University of Technology (TU/e), The Netherlands. His research project was performed in cooperation with Philips Medical Systems and focused on the design of transfer functions for direct volume rendering.