Efficient Boundary Detection and Transfer Function Generation in Direct Volume Rendering

Jörg-Stefan Praßni, Timo Ropinski, Klaus H. Hinrichs

Visualization and Computer Graphics Working Group (VisCG),
Department of Computer Science, University of Münster, Germany
Email: {j-s.prassni,ropinski,khh}@math.uni-muenster.de

Abstract

In this paper we present an efficient technique for the construction of LH histograms which, in contrast to previous work, does not require an expensive tracking of intensity profiles across boundaries and therefore allows an LH classification in real time. We propose a volume exploration system for the semi-automatic generation of LH transfer functions, which does not require any user interaction within the transfer function domain. During an iterative process the user extracts features by marking them directly in the volume rendered image. The system automatically detects a marked feature's boundary by exploiting our novel LH technique and generates a suitable component transfer function that associates user-specified optical properties with the region representing the boundary in the transfer function space. The component functions thus generated are automatically combined to produce an LH transfer function to be used for rendering.

1 Introduction

In the past years volumetric data sets, which arise from simulation or acquisition, have been steadily increasing in size. Today, the sizes are usually beyond what could be examined efficiently slice-by-slice. As a consequence, the demand for sophisticated interactive exploration techniques is growing. Krueger et al. state that a large portion of users have difficulties in understanding non-trivial data sets and in finding what they are looking for in those data sets [KSW06]. Since many users of interactive volume rendering software do not have the necessary visualization skills or the time, intuitive and simple techniques for exploration are needed. The major interaction task to be performed during volume exploration is the interactive transfer function specification, which is necessary to identify the features of interest. According to Rezk-Salama and Kolb [RSK06], transfer function setup in general is a time-consuming and cumbersome process, which becomes even more intricate when specifying multidimensional transfer functions.

Šereda et al. [SBSG06] introduced the LH space as transfer function domain. They show that the LH histogram conveys information about a data set's boundaries in a more compact and robust way than common intensity-gradient histograms and therefore seems to be well-suited for volume exploration. Their technique, however, requires complex computations that do not allow LH post-classification but have to be performed in a preprocessing step.

The contributions of this paper are: first, an efficient technique for the computation of LH values, which is fast enough to allow post-classification at interactive frame rates. Second, an intuitive and efficient mechanism for specifying LH transfer functions that does not require any user interaction within the transfer function domain. We propose a sketching metaphor, which allows to mark the features of interest in image space in order to assign the desired optical properties. Thus the user is able to sequentially identify the features of interest and to make them visible by the matter of a mouse click later on. This approach also enables less experienced users to interactively explore complex volumetric data sets without requiring training.

2 Related Work

Volume Classification. König and Gröller [KG01] propose an image-centric technique for transfer function specification. They treat each specification domain such as data value, opacity, and color separately and arrange thumbnail renderings according to the value ranges they represent. When the user has specified a value range for each domain, these ranges are combined to produce the final transfer function.

Wu and Qu [WQ07] presented a framework for combining and editing existing transfer functions based on genetic algorithms. The user operates on volume renderings, which are generated by applying the respective transfer functions, by sketching the features that are desired to be visible in the combined rendering. While achieving impressive results in combination with an intuitive user interaction, this technique heavily depends on the quality of pre-generated transfer functions. Ropinski et al. [RPSH08] proposed an interface for the design of 1D transfer functions that is based on direct interaction with the rendered volume. After the user has identified a feature of interest by drawing strokes close to its silhouette, a suitable component transfer function is generated based on a histogram analysis.

Kindlmann and Durkin [KD98] presented a semi-automatic data-centric technique for the generation of opacity transfer functions that aims at visualizing boundaries between materials. They incorporate a boundary model that enables them to map the combination of the data value and its first and second derivatives at a sample point to a position along a boundary. We exploit this boundary model for an efficient computation of LH values. Kniss et al. [KKH02] showed how a 2D histogram of the data values and the gradient magnitudes can be used as a 2D transfer function domain. In this histogram space boundaries appear as arcs, which they select by using interaction widgets. They also introduced a dual-domain interaction that eases the setup of higher-dimensional transfer functions by data probing within the volume. In contrast to our approach, however, the user is still required to interact within the transfer function domain. Further approaches for data probing-based volume classification were proposed by Tzeng et al. [TLM03] [TLM05]. They use neural networks for the generation of a transfer function based on samples the user has drawn onto slices of the volume. Rezk-Salama and Kolb [RSK06] introduce opacity peeling for the extraction of feature layers that allows the extraction of structures which are difficult to classify using conventional transfer functions.

LH Histograms. Šereda et al. [SBSG06] use so-called *LH histograms* for transfer function generation. They assume that every voxel lies either inside a material or on the boundary between two materials with lower intensity F_L and higher intensity F_H, respectively. The LH histogram is a 2D histogram whose axes correspond to F_L and F_H. It is built from the data set by accumulating boundary voxels with the same (F_L,F_H) coordinates, which are retrieved by analyzing the intensity profile across a boundary. The authors show that the LH histogram conveys information about a data set's boundaries in a more compact and robust way than common 2D histograms incorporating the intensity and gradient magnitude, because in LH histograms boundaries appear as blobs instead of arcs. A further significant advantage of the LH space is that it allows an unambiguous classification of boundaries with distinct LH values, which is not the case for the intensity-gradient space due to arch overlaps as shown by Kniss et al. [KKH02]. These properties make the LH space a very attractive transfer function domain, especially for the semi-automatic classification of boundaries. In [SGV06], Šereda et al. described a projection of the LH space to a 1D transfer function domain that allows the classification of complete objects instead of single boundaries.

A major drawback of the LH technique, however, are the complex computations required for the computation of LH values, since for each boundary voxel the intensity profile across the boundary has to be analyzed by integrating the gradient field until a constant area, a local extremum or an inflex point is reached. Therefore, the LH classification cannot be performed in real time during the rendering process. Instead, an additional volume storing the LH values of all voxels has to be generated. Besides the required pre-computation time, this approach triples the memory consumption of the volumetric data to display, since the LH volume contains two channels per voxel and must match the original volume in resolution and bit depth. This is especially problematic for GPU-based volume rendering.

One could argue that post-interpolative classification is still possible with pre-computed LH values, since the stored LH values can be interpolated before actually applying the transfer function. However, we consider the LH value computation to be part of the classification process. Therefore, in this work the term ”LH post-classification” de-

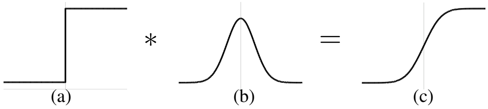


Figure 1: Ideal boundaries (c) in volume data sets are step functions (a) blurred by a Gaussian (b).

notes the computation of LH values from interpolated volume data.

3 Efficient Construction of LH Histograms

In this section, we present an efficient way to compute a boundary voxel's LH values by only considering its intensity, gradient magnitude and second directional derivative along the gradient direction. The calculation is fast enough to allow post-classification at interactive frame rates. In order to compute LH values without the expensive tracking of a path to the neighbored materials, it is necessary to make assumptions about the characteristics of boundaries in volume data sets. Since actual scanning devices are band-limited, they are unable to exactly reproduce point objects or sharp boundaries. In general, the band-limiting characteristics of an imaging system are described by its *point spread function (PSF)*, which specifies the system's impulse response to a point source. For the following considerations we employ the boundary model proposed by Kindlmann and Durkin [KD98]. They assume that real objects have sharp edges, i.e., discontinuous changes in the measured physical property, and model the PSF by a Gaussian function which is isotropic and constant over a data set. Although these assumptions might appear rather strong, the authors have shown that they match well the characteristics of CT and to some extent of MRI data sets. From a more intuitive point-of-view, one can think of boundaries as step functions that are blurred by a Gaussian as depicted in Figure 1.

3.1 Boundary Function

We start our derivation by considering a path that continuously follows the gradient direction through a boundary between the two materials with intensities F_L and F_H. Since the gradient vector at a position within a scalar field is always perpendicular to the isosurface through this point, this path also penetrates the boundary perpendicularly and thus constitutes the shortest path. In the following, we refer to this as the *boundary path.* For a mathematical description, we introduce the *boundary function* $f(x)$, which maps a position x along the boundary path to the intensity v at the respective sampling point. According to the boundary model, $f(x)$ is a result of the convolution of a step function describing the physical boundary and a Gaussian function with the standard deviation σ. Therefore, the boundary function can be defined as:

$$f(x) = F_L + (F_H - F_L)\ \Phi\left(\frac{x}{\sigma}\right) \qquad (1)$$

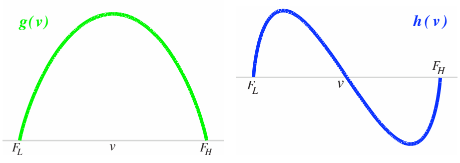
with $\Phi(x)$ being the cumulative distribution function (CDF) of the standard normal distribution. The center of the boundary is defined to be located at $x = 0$. Equation (1) tells us that each boundary features its own boundary function, which is parametrized by the intensities F_L and F_H of the neighboring materials as well as a parameter σ that specifies the amount of blurring that happened to the boundary. However, under the assumption that the boundary blurring is an attribute of the scanning device and is therefore uniform over a data set, the *blurring parameter* σ can be considered to be a constant. For the determination of σ as well as a derivation of the boundary function refer to Kindlmann and Durkin [KD98]. The first and second derivatives of $f(x)$ are as follows:

$$f'(x) = \frac{F_H - F_L}{\sigma\sqrt{2\pi}}\ e^{-\frac{x^2}{2\sigma^2}} \qquad (2)$$

$$f''(x) = -x\ \frac{F_H - F_L}{\sigma^3\sqrt{2\pi}}\ e^{-\frac{x^2}{2\sigma^2}} \qquad (3)$$

Apparently, $f'(x)$ has the form of a Gaussian function. Since the Gaussian function has inflection points at $x = \pm\sigma$, these are also the positions where $f''(x)$ attains its extrema.

Since the CDF of the standard normal distribution $\Phi(x)$ is bijective and thus invertible within its range, we can directly conclude from Equation (1) that the same property applies to $f(x)$. Therefore, we can define the inverse of the boundary function $f^{-1}(v)$, which maps an intensity $v \in\ ]F_L, F_H[$ at a sampling point within the boundary between two



(a) (b)

Figure 2: First (a) and second (b) derivative of the boundary function as functions of intensity.

materials F_L and F_H to a position x along the corresponding boundary path:

$$\begin{aligned} & f(x) \;=\; v \\ \Leftrightarrow\quad & F_L + (F_H - F_L)\;\Phi\left(\frac{x}{\sigma}\right) \;=\; v \\ \Leftrightarrow\quad & x \;=\; \sigma\;\Phi^{-1}\left(\frac{v - F_L}{F_H - F_L}\right) \\ \Leftrightarrow\quad & f^{-1}(v) \;=\; \sigma\;\Phi^{-1}\left(\frac{v - F_L}{F_H - F_L}\right) \qquad (4) \end{aligned}$$

$f^{-1}(v)$ enables us to express the boundary function's derivatives as functions of intensity instead of position:

$$g(v) \;:=\; f'\left(f^{-1}(v)\right) = \frac{F_H - F_L}{\sigma\sqrt{2\pi}}\exp\left(\bar{\Phi}\right) \quad (5)$$

$$h(v) \;:=\; f''\left(f^{-1}(v)\right) \quad (6)$$

$$= \;-\frac{F_H - F_L}{\sigma^2\sqrt{2\pi}}\;\Phi^{-1}\left(\bar{v}\right)\;\exp\left(\bar{\Phi}\right) \quad (7)$$

$$\bar{v} := \frac{v - F_L}{F_H - F_L} \qquad \bar{\Phi} := -\frac{1}{2}\Phi^{-1}\left(\frac{v - F_L}{F_H - F_L}\right)^2$$

The functions $g(v)$ and $h(v)$ specify the first and second derivative of the boundary function at the position x with $f(x) = v$. Plots of them are shown in Figure 2. $g(v)$ can be used to map the LH space to the conventional intensity-gradient transfer function space. Though this mapping is not injective and therefore suffers from information loss, it allows a basic integration of the proposed classification technique into volume rendering systems without the need for a LH post-classification as described in Section 4.3.

3.2 Computation of LH Values

By considering these assumptions, we now return to our goal of calculating the LH values at an arbitrary sampling point within a boundary. We consider the boundary path that runs through the sampling point. The position along this path that corresponds to the sampling point is labeled x_p. Furthermore, we assume the function values $f(x_p)$, $f'(x_p)$, and $f''(x_p)$ to be known. For clarity of presentation, these substitutes are used in the following derivation:

$$v := f(x_p) \qquad g := f'(x_p) \qquad h := f''(x_p)$$

First of all, we can recover x_p from the ratio of g and h. Dividing Equation (2) by Equation (3) yields:

$$\frac{g}{h} \;=\; -\frac{\sigma^2}{x_p} \quad\Leftrightarrow\quad x_p \;=\; -\sigma^2\,\frac{h}{g} \qquad (8)$$

By additionally considering Equation (1) it is now possible to determine F_L:

$$\begin{aligned} \textbf{I}\text{ (Eq. 1)}\qquad & v \;=\; F_L + (F_H - F_L)\,\Phi\left(\frac{x_p}{\sigma}\right) \\ \textbf{II}\text{ (Eq. 2)}\qquad & g \;=\; \frac{F_H - F_L}{\sigma\sqrt{2\pi}}\;\exp\left(-\frac{x_p^2}{2\sigma^2}\right) \\ \Leftrightarrow\quad & F_H \;=\; \frac{\sigma\sqrt{2\pi}\,g}{\exp\left(-\frac{x_p^2}{2\sigma^2}\right)} + F_L \end{aligned}$$

$$\begin{aligned} \textbf{II in I}\qquad & v \;=\; F_L + \frac{\sigma\sqrt{2\pi}\,g}{\exp\left(-\frac{x_p^2}{2\sigma^2}\right)}\;\Phi\left(\frac{x_p}{\sigma}\right) \\ \Leftrightarrow\quad & F_L \;=\; v - \frac{\sigma\sqrt{2\pi}\,g}{\exp\left(-\frac{x_p^2}{2\sigma^2}\right)}\;\Phi\left(\frac{x_p}{\sigma}\right) \\ \underset{(8)}{\Rightarrow}\quad & F_L \;= v - \frac{\sigma\sqrt{2\pi}\,g}{\exp\left(-\frac{h^2}{g^2}\frac{\sigma^2}{2}\right)}\;\Phi\left(-\sigma\frac{h}{g}\right) \end{aligned} \qquad (9)$$

F_H can be derived by inserting F_L and x_p into Equation (1):

$$F_H \;=\; \frac{v - F_L}{\Phi\left(-\sigma\,\frac{h}{g}\right)} + F_L \qquad (10)$$

The preceding analysis is based on the assumption that the boundary function $f(x)$ and its derivatives are known at all positions within boundaries. This is true for $f(x)$ itself, as it simply equals the intensity value at a sampling point; the determination of the boundary function's derivatives, however, needs further consideration. Recalling that the boundary path is defined to continuously follow the gradient direction, $f'(x)$ and $f''(x)$ turn out to be the data set's first and second directional derivatives along the gradient direction. According to vector calculus [MT96], the directional derivative $\mathbf{D}_{\vec{v}}$ along the direction $\vec{v}$ of a scalar field $s(\vec{r})$ is the scalar product of the gradient of s and the vector $\vec{v}$ in normalized form:

$$\mathbf{D}_{\vec{v}}\, s = \nabla s \cdot \frac{\vec{v}}{||\vec{v}||}$$

Hence, the first derivative along the gradient direction is just the gradient magnitude:

$$\mathbf{D}_{\nabla s}\, s \;=\; \nabla s \cdot \frac{\nabla s}{||\nabla s||} = ||\nabla s||$$

In a similar way, we obtain the second directional derivative along the gradient direction:

$$\begin{aligned} \mathbf{D}^2_{\nabla s}\, s \quad &= \quad \mathbf{D}_{\nabla s}\, (\mathbf{D}_{\nabla s}\, s) = \mathbf{D}_{\nabla s}\, (||\nabla s||) \\ &= \quad \frac{\nabla\, (||\nabla s||) \cdot \nabla s}{||\nabla s||} \end{aligned}$$

3.3 Implementation

We integrated the proposed LH classification technique into an existing GPU-based volume raycaster. For the calculation of LH values by formulas (9) and (10) the gradient magnitude g and the second derivative h have to be known at each sampling point. We compute approximations of these quantities by central differences and the discrete Laplacian operator, respectively, either on-the-fly or in a preprocessing step. Since the CDF of the standard normal distribution Φ is not available on the GPU, we sample it into a one-dimensional texture that can be accessed by the shader. Due to the low-frequency character of Φ a resolution between 32 and 64 samples seems to be sufficient.

3.4 Comparison with Šereda's Method

In order to compare the proposed technique to the original one in terms of speed and accuracy, we applied it to some of the data sets used by Šereda et al. We did not implement their method but took the results presented in [SBSG06].

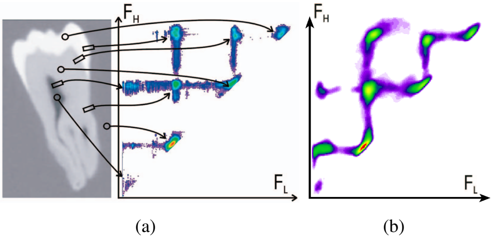


Figure 3: LH histograms of the tooth data set generated with the original method (a) (courtesy of Šereda et al. [SBSG06]) and with our technique (b). The contribution is shown in logarithmic scale: red is highest, magenta is lowest.

3.4.1 Computation Time

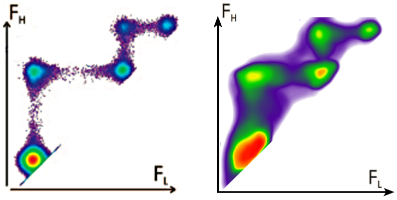
Table 1 shows performance comparisons for the tooth and the hand data set, both CT scans. The measurements were conducted on an Intel Core 2 Duo 2.2 GHZ machine with an NVIDIA GeForce 8800 GTX graphics board. We determined the construction time of an LH histogram of the entire data set on the CPU with a single-threaded implementation as well as the rendering speed when performing an LH post-classification in combination with Phong shading in a GPU-based raycaster. Both measurements were performed with and without pre-computed gradient magnitudes and second derivatives. Šereda et al. used pre-computed derivatives for their benchmark.

For the single-threaded LH histogram generation on the CPU we achieved a significant speed up of about one order of magnitude with pre-calculated derivatives and still a speed up of about factor five when generating the derivatives on the fly. Since the LH values can be computed independently for each voxel position, a nearly linear speedup can be expected for a multi-threaded implementation. It should be considered, however, that Šereda et al. did not specify their hardware configuration, which hampers the comparability of the results.

As the authors admitted, the original LH technique is not fast enough to allow post-classification, and therefore they had to store the LH classification in an additional pre-computed volume. In contrast, our method allows LH post-classification at interactive frame rates, as can be seen from Table 1. For an estimation of the additional effort caused by the LH value calculation in the shader we also

		LH Hist. Construction on CPU		Rendering Speed (FPS)	
data set	size	Šereda et al.	our technique	2D TF	LH TF
tooth	256^2 x 161	1:12 min	7.2 / 15.3 s	35.9	26.9 / 20.5
hand	256^2 x 232	1:36 min	10.7 / 19.4 s	32.2	24.5 / 17.3

Table 1: Performance of the LH classification of two CT data sets. Columns 3 and 4 compare the construction time of LH histograms on the CPU by the original technique and by ours. For our technique, the first figure indicates the performance with pre-computed derivatives, the second without. Columns 5 and 6 specify frame rates for the LH post-classification with and without pre-computed derivatives compared to the application of a conventional 2D transfer function with a viewport size of 512^2.



(a) (b)

Figure 4: LH histograms of an artificial data set of two noisy spheres computed with Šereda's (a) and our technique (b).

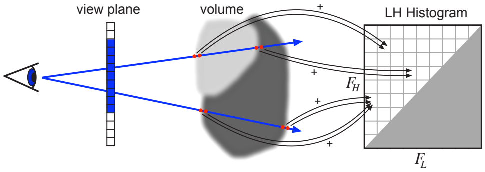


Figure 5: Generation of the LH histogram. A ray is cast through each view plane pixel covered by the user-drawn patch (blue). Each visible boundary voxel hit by one of these rays is mapped to the histogram bin (F_L, F_H) representing its boundary.

determined frame rates for a classification with a conventional intensity-gradient based transfer function, which was set up to produce a rendering result resembling the one achieved with the LH classification as closely as possible: with pre-computed derivatives we noticed a frame rate drop of only about 30 %, while an on-the-fly calculation of the derivatives slows down the rendering by about 40 % compared to the 2D transfer function.

3.4.2 Accuracy

The ability to calculate sufficiently accurate LH values is crucial for the proposed technique. Figure 3 compares an LH histogram of the tooth data set computed by our technique to the result presented in [SBSG06]. Although our LH histogram appears to be slightly more blurry and especially the horizontal and vertical bars between the boundary blobs are more pronounced, it clearly exhibits the same structure as the original LH histogram. Note that not only the blobs representing boundaries are existent but also the blobs on the diagonal of the histogram that represent homogeneous regions.

In order to investigate to what extent our LH calculation is prone to noise, we generated an artificial data set of two spheres blurred by a Gaussian and added 0.1 percent of Gaussian noise to it, similar to the data set used by Šereda et al. The LH histogram we derived from the volume (Figure 4) still shows the expected blobs, but also exhibits a broader distribution of misclassified voxels around these blobs than in the original histogram. As this seems to be a consequence of the noise sensitivity of the standard derivative estimators, we believe that an application of more advanced derivative reconstruction schemes could significantly improve the noise robustness of our method.

4 Volume Exploration

In this section, we describe a semi-automatic volume exploration system that exploits the LH technique in order to enable the user to interactively classify features of interest. In an iterative process the user extracts boundaries by directly marking them in the volume rendered image. The system then detects LH values of the respective boundaries and generates a suitable LH transfer function incorporating user-specified optical properties.

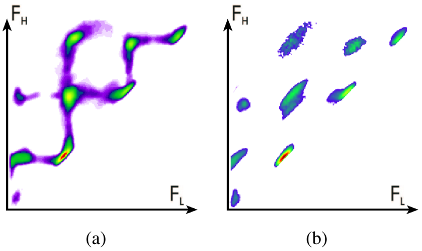


Figure 6: LH histograms of the tooth data set generated by our technique without weighting (a) and with distance weighting (b).

4.1 User Interaction

In our system, the user is expected to mark a desired feature of interest by drawing a free-form patch onto the region covered by the respective feature in image space. It is neither necessary to precisely sketch a feature's silhouette nor has the user to mark the whole object. Furthermore, during the exploration no interaction within the transfer function domain is required, rather the user can apply the commonly used windowing approach to change the visibility of parts of the data set.

In order to offer the user the possibility to assign optical properties to features independently, we use a layer interface similar in spirit to the one presented by Ropinski et al. [RPSH08]. Each extracted boundary is represented by a layer through which the user can specify the boundary's color and opacity.

4.2 Boundary Extraction

After the user has drawn a patch, the system analyzes the subvolume determined by the extrusion of the patch in the viewing direction in order to detect all visible boundaries contained within this subvolume. The boundary detection is performed by an analysis of the LH histogram of this subvolume. We create the LH histogram by casting a ray through each pixel of the user-drawn patch in the view plane. At each visible boundary voxel that is hit by one of these rays, the F_L and F_H values are calculated, and the corresponding histogram cell with coordinates (F_L, F_H) is incremented. Figure 5 depicts this proceeding.

As we are interested in the detection of features that have been marked by the user, it is necessary to consider only those voxels for the construction of the LH histogram that are visible in the current rendering. For that purpose, we weight each voxel's contribution by its opacity, which is determined by the current windowing. In order to prevent opaque but completely occluded voxels from contributing to the histogram, we stop the traversal of a ray when it is saturated, i.e., when the ray has reached an alpha value of 1.0 during the front-to-back compositing. This allows an depth-selection of features.

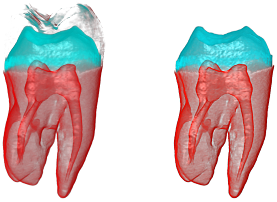
Since we aim at the extraction of boundaries, it is necessary to consider only boundary voxels for the LH histogram. The straightforward approach to incorporate only voxels with a gradient magnitude above a certain threshold, however, has some disadvantages:

- The gradient magnitude is boundary-dependent, i.e., the gradient magnitude distribution within a boundary is proportional to the intensity range that is spanned by the boundary. This makes it difficult to define an adequate threshold for the entire data set.
- Noisy regions exhibit a significantly high gradient magnitude and may thus accidentally contribute to the histogram.

Instead, we use a voxel's position x_p along its boundary path (see Eq. (8)) for weighting its contribution. The weighting factor w is given by:

$$w := \exp\left(-x_p^2\right) = \exp\left(-\sigma^4 \frac{h^2}{g^2}\right) \qquad (11)$$

As we defined voxels with $x_p = 0$ to be located at the center of a boundary, the contribution of these center voxels is maximal while an increasing distance to the center reduces the influence on the LH histogram. Furthermore, this weighting diminishes the influence of noisy regions, since such regions usually exhibit high second order derivatives in relation to the gradient magnitude. Figure 6 (b) demonstrates the effect of the distance weighting for the tooth data set: The boundary as well as the material blobs are significantly more localized and the connecting bars almost vanish. We also noticed a substantial reduction of rendering artifacts when applying the distance weighting. Figure 7 illustrates this for the tooth data set. Both renderings have been generated with the same LH transfer function consisting of two blobs that are located at the respective coordinates of the dentin-background (red) and dentin-enamel (blue) boundaries in LH



(a) (b)

Figure 7: LH classified renderings without weighting (a) and with distance weighting (b).

space. Without distance weighting, there are several misclassified voxels at the enamel-background boundary visible, whereas distance weighting eliminates these artifacts without perceptibly effecting the boundaries themselves. Apparently, the LH classification is most reliable at the center of a boundary. Therefore, we use the distance weighting not only for the boundary extraction but also as part of the classification.

After the construction of the LH histogram, the detection of boundaries is now reduced to a local maxima search in the histogram space, as each blob in the LH histogram represents a boundary within the analyzed subvolume. In order to cope with noise an discretization artifacts, we apply a slight blurring before the maxima detection, e.g. by a 3x3 Gaussian kernel for an 8 bit histogram.

4.3 Transfer Function Generation

The feature extraction yields a list of tuples (F_L, F_H) representing the extracted boundaries. This boundary information can be used in a straightforward way to generate a suitable LH transfer function for the visualization of these boundaries. In our setup, each boundary is represented by a single layer in the user interface and associated with an LH component function that contains a Gaussian bell curve centered around the boundary's (F_L, F_H) coordinates. Besides a boundary's optical properties, the user can control its bell curve's variance. We call this parameter ”fuzziness” as it specifies the size of the bell curve in LH space and therefore determines to what extent voxels with slightly deviating LH values are incorporated by the component function. In order to produce an LH transfer function that can actually be used during the rendering process, the boundaries' component functions are combined by calculating their weighted average based on opacity. This yields a transfer function containing the LH blobs of all classified boundaries.

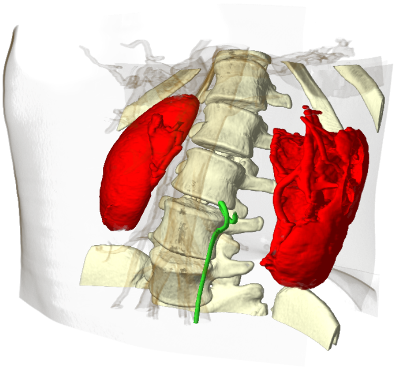


Figure 8: Application of the proposed technique to a renal angio CT data set. We extracted the bone, the kidneys, the blood vessels, the skin, and a stick.

We want to stress that the proposed volume exploration technique does not necessarily require an LH classification during the rendering process. Instead, the extracted boundary information can be used to generate a 2D transfer function based on the intensity and gradient magnitude, which is nowadays widely used by volume rendering systems. The region that represents a boundary in this transfer function space can be determined by evaluating Equation (5), which provides us with the gradient magnitude as a function of intensity. This mapping, however, is not injective since distinct LH coordinates may be mapped to intersecting arcs in the intensity-gradient space.

5 Results

We applied the proposed volume exploration technique to three medical CT scans as well as an electron microscopy scan. All renderings have been generated by interaction with the described user interface only, without any direct manipulation of the LH transfer function.

Figure 8 shows results of the application of the proposed volume exploration technique to a renal angio CT scan. We extracted five features: the bone,

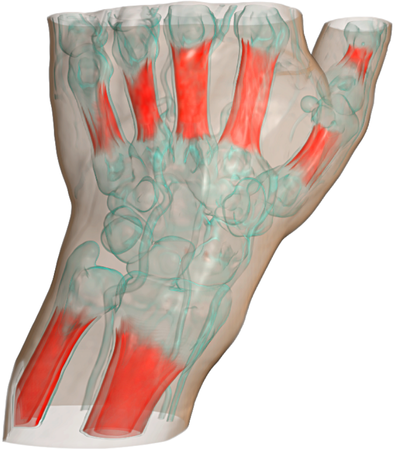


Figure 9: Classification of the hand data set (CT). The tissue-background boundary (skin), the bone-tissue (blue), and the bone-marrow boundaries (red) are visible.

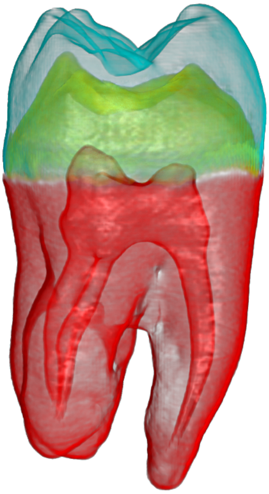


Figure 10: Rendering of the tooth data set (CT). The dentin-background (red), the dentin-enamel (yellow), and the enamel-background (blue) boundaries have been classified.

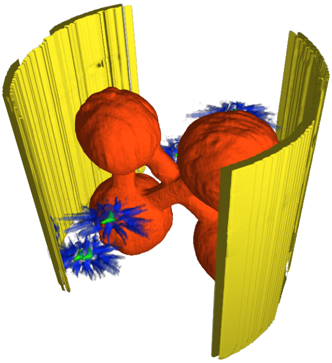


Figure 11: Volume exploration of an electron microscopy scan with four extracted features.

the kidneys, the blood vessels (semi-transparent), the skin, and an apparently synthetic stick (green). The exploration of this data set took about five minutes, including the assignment of optical properties fine-tuning the layers' fuzziness parameters.

Figure 9 presents a classification of the hand CT data set. Besides the tissue-background boundary, which appears as the skin in the rendering, we extracted the bone-tissue boundary (blue) as well as the bone-marrow boundary (red). The extraction process took just about one minute. However, we were not able to separate the blood vessels from the bone as they share a common footprint in the LH space. Figure 10 shows the result of the application of the proposed technique to the tooth data set. In Figure 11 an electron microscopy scan is explored. We managed to classify four features in this data set with relatively little effort. This indicates that the assumed boundary model works well for this type of data. We could not further investigate that, because the data set was the only electron microscopy scan we had access to.

6 Conclusions and Future Work

We have presented an efficient method for the calculation of LH values that does not require a tracking of boundary intensity profiles but is based on local measures. We have shown that it allows post-classification at interactive frame rates whereby the pre-computation of an LH volume is not necessary. By comparing our results to the work of Šereda et al. [SBSG06], we have demonstrated that the proposed technique is sufficiently robust, at least when applied to CT data. Also due to the relatively easy implementation, we believe that our novel LH classification has the potential to boost the use of the LH space as transfer function domain.

Moreover, we have proposed a system for the semi-automatic design of LH transfer functions, which completely shields the user from the transfer function domain and allows him/her to extract features of interest by direct interaction with the rendered volume and to conveniently assign optical properties to these features. Furthermore, we have pointed out a possibility to exploit our system for the generation of conventional 2D transfer functions based on the intensity and gradient magnitude, which eases the integration into existing volume rendering systems.

In the future, we would like to improve the applicability of our LH classification to MRI data. We believe that this could be achieved by exploiting more advanced derivative reconstruction schemes or an adaption of the boundary model.

Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partly supported by grants from Deutsche Forschungsgemeinschaft, SFB 656 MoBil Münster (project Z1). The presented concepts have been integrated into the Voreen volume rendering engine (http://www.voreen.org).

References

[KD98] KINDLMANN G., DURKIN J. W.: Semi-automatic generation of transfer functions for direct volume rendering. In *VVS '98: Proceedings of the 1998 IEEE symposium on Volume visualization* (New York, NY, USA, 1998), ACM, pp. 79–86.

[KG01] KÖNIG A., GRÖLLER E.: Mastering transfer function specification by using volumepro technology. In *Proceedings of the 17th Spring Conference on Computer Graphics 2001* (2001), pp. 279–286.

[KKH02] KNISS J., KINDLMANN G., HANSEN C.: Multidimensional transfer functions for interactive volume rendering. 270–285.

[KSW06] KRUGER J., SCHNEIDER J., WESTERMANN R.: Clearview: An interactive context preserving hotspot visualization technique. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 941–948.

[MT96] MARSDEN J. E., TROMBA A. J.: *Vector Calculus*. W.H. Freeman and Company, New York, 1996.

[RPSH08] ROPINSKI T., PRASSNI J.-S., STEINICKE F., HINRICHS K. H.: Stroke-based transfer function design. In *IEEE/EG International Symposium on Volume and Point-Based Graphics* (2008), IEEE, pp. 41–48.

[RSK06] REZK-SALAMA C., KOLB A.: Opacity peeling for direct volume rendering. *Comput. Graph. Forum 25*, 3 (2006), 597–606.

[SBSG06] SEREDA P., BARTROLI A. V., SERLIE I. W. O., GERRITSEN F. A.: Visualization of boundaries in volumetric data sets using lh histograms. *IEEE Transactions on Visualization and Computer Graphics 12*, 2 (2006), 208–218.

[SGV06] SEREDA P., GERRITSEN F. A., VILANOVA A.: Mirrored lh histograms for the visualization of material boundaries. *Proceedings of Vision, Modeling, and Visualization* (2006), 237–244.

[TLM03] TZENG F.-Y., LUM E. B., MA K.-L.: A novel interface for higher-dimensional classification of volume data. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)* (Washington, DC, USA, 2003), IEEE Computer Society, p. 66.

[TLM05] TZENG F.-Y., LUM E. B., MA K.-L.: An intelligent system approach to higher-dimensional classification of volume data. *IEEE Transactions on Visualization and Computer Graphics 11*, 3 (2005), 273–284.

[WQ07] WU Y., QU H.: Interactive transfer function design based on editing direct volume rendered images. *IEEE Transactions on Visualization and Computer Graphics 13*, 5 (2007), 1027–1040.