

Visualization of Boundaries in Volumetric Data Sets Using LH Histograms

Petr Šereda, Anna Vilanova Bartrolí, Iwo W.O. Serlie, and Frans A. Gerritsen

Abstract—A crucial step in volume rendering is the design of transfer functions that will highlight those aspects of the volume data that are of interest to the user. For many applications, boundaries carry most of the relevant information. Reliable detection of boundaries is often hampered by limitations of the imaging process, such as blurring and noise. We present a method to identify the materials that form the boundaries. These materials are then used in a new domain that facilitates interactive and semiautomatic design of appropriate transfer functions. We also show how the obtained boundary information can be used in region-growing-based segmentation.

Index Terms—Volume visualization, direct volume rendering, transfer functions, multidimensional transfer functions, region growing.

1 INTRODUCTION

DIRECT volume rendering is a powerful visualization technique to provide 3D views of volumetric data. In medical diagnostic imaging, typical sources of volumetric data are CT and MR scanners. The typical size of the data sets produced by these scanners grows with every new version of the equipment. However, the data sets need to be examined quickly. Two-dimensional slice-by-slice or slab-by-slab “stack viewing” is a time-consuming process that does not facilitate the mental reconstruction of the 3D shape of objects. However, it is still used by most radiologists. Volume rendering is capable of showing the data in 3D, where the sizes, shapes, and relations between objects are usually easier to observe.

In order to create a useful visualization, it is important to show the information that is relevant for the application at hand. To achieve this, a mapping from data properties (e.g., scalar value and gradient magnitude) to optical properties (e.g., color, opacity, and luminance) may be used to determine what should be visualized and in which way. Such a mapping is called a transfer function (TF). The opacity setting is used to select which parts of the volume will be visible. Color may help to visually distinguish between objects.

It is not trivial to choose the domain of the transfer function, i.e., to choose the data properties that enable a good distinction between objects of interest. The volume data is usually treated as a discrete representation of a

continuous scalar function $f(\vec{x})$, where $\vec{x} \in \mathbb{R}^3$. The most often used 1D transfer functions use the scalar values of the volume (i.e., the range of $f(\vec{x})$) as the domain. The scalar value is used in order to distinguish materials. In addition, as a second dimension, the gradient magnitude $|\nabla f|$ is often used to emphasize the strong boundaries between objects (see Levoy [1] and Kindlmann and Durkin [2]). Visualizing the boundaries is important for perceiving shapes of objects and their 3D configuration.

After choosing the domain comes the task of designing the transfer function itself. The TF is often 1D and defined interactively by placing peaks and ramps above ranges of values where a certain material is expected. This is often a cumbersome trial-and-error process since there may be no trivial relationship between the TF and the final result. In order to help the user focus on the most frequent values, a histogram of the occurrences of the scalar values is usually shown.

Transfer functions based only on scalar values often cannot distinguish between different boundaries. Kindlmann and Durkin [2] showed that important boundaries can be found by using a higher dimensional domain defined by the data value, gradient magnitude, and the second-order derivative along the gradient direction. Kniss et al. [3] introduced interaction widgets that allow visualization of boundaries by manual selection in that domain. This domain can be easily used for a small number of boundaries. However, with an increasing number of boundaries, their separation becomes more difficult due to intersections and overlaps.

In this paper, we present a novel multidimensional transfer function domain that is aimed at facilitating the selection of boundaries between materials. This is achieved by using the so-called LH Histogram (Serlie et al. [21]) that shows lower and higher intensities of materials that form the boundaries. We show properties of this domain under the influence of noise, bias, and partial volume effect. The LH Histogram represents more complex properties than the commonly used scalar value and its derivatives. This aspect makes our method different from traditional approaches

• P. Šereda and A. Vilanova Bartrolí are with the Eindhoven University of Technology, Department of Biomedical Engineering, Image Analysis Group, WH 3.2a, Postbus 513, 5600 MB Eindhoven, The Netherlands. Email: {p.sereda, a.vilanova}@tue.nl.

• I.W.O. Serlie is with the Delft University of Technology, Department of Applied Physics, Pattern Recognition Group, Lorenzweg 1, 2628 CJ Delft, The Netherlands. Email: iwo@ph.tn.tudelft.nl.

• F.A. Gerritsen is with the PHILIPS Medical Systems Nederland BV, Research and Advanced Development, P.O. Box. 10000, 5680 DA Best, The Netherlands. Email: frans.gerritsen@philips.com.

Manuscript received 17 Nov. 2004; revised 17 June 2005; accepted 5 July 2005; published online 10 Jan. 2005.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-0142-1104.

and similar to other approaches that use more complex operations for transfer functions (e.g., [4] or [5]). We also show that the LH information can be used for segmentation algorithms, such as region growing.

In the following section, we describe the relation of our work to a variety of existing methods. Then, in Section 3, we describe the construction and properties of the LH Histogram. In Section 4, we show how to use the LH Histogram for classification. In Section 5, we explain how LH values may be used to improve region-growing based segmentation.

2 RELATED WORK

In this section, we give a more detailed overview of some important work that has been done in the field of transfer functions. It is, however, not easy to make a fair comparison of different methods. An example of a head-to-head comparison of several approaches can be found in the Transfer Function Bake-Off [6]. In our description, we give special attention to those approaches that aim at visualization of boundaries, which is also the goal of our approach.

2.1 One-Dimensional Transfer Functions

The first transfer functions used for volume rendering were one-dimensional. In order to overcome the difficulties with the manual trial-and-error design of these functions, several automation methods have been proposed. These methods are either based on evaluation of the volume data (data-centric methods) or on evaluation of the rendered image (image-centric methods).

Bajaj et al. [7] introduced a data-centric method that generates the contour spectrum to help the user determine those gray levels that would correspond to important iso-surfaces. This approach facilitates automatic selection of iso-surfaces of interest. Pekar et al. [8] expanded on this principle and introduced an efficient construction of such spectra.

Another automated method, presented by Fujishiro et al. [9], aimed at finding important iso-surfaces by investigating their behavior while changing the iso value. At certain values, surfaces split or merge. Such surfaces were then emphasized in the visualization.

An image-centric type of automation was presented by He et al. [10]: A genetic algorithm was used to generate 1D TF. The successfulness of the “current 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 resulting visualization sounds very promising. However, choosing good criteria for an automatic assessment of the image quality is quite problematic.

Marks et al. [11] and König and Gröller [12] proposed design galleries as an interaction with the user. In design galleries, the space of all possible transfer functions is automatically sampled and a rendering is generated for each sample. In an iterative process, the user evaluates the quality of offered images and chooses the best one. The intuitive image-based user interaction is the main advantage of this approach. It is, however, difficult to sample the space so that all important combinations are depicted. Also, the number of samples needs to be small in order to allow

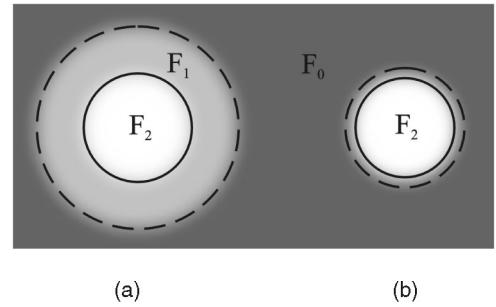


Fig. 1. (a) There is a sphere of material intensity F_2 placed inside another sphere of lower intensity F_1 such that $F_2 > F_1$. (b) There is a second sphere of intensity F_2 . The background has intensity F_0 such that $F_2 > F_1 > F_0$. The edges are blurred.

an interactive generation of sample renderings. The design galleries can also be used for higher dimensional transfer functions. However, with increasing dimensionality, it becomes more difficult to reasonably sample the transfer function space.

2.2 Multidimensional Transfer Functions

Curvatures have been used to help distinguish between objects according to their shapes. Hladuvka et al. [13] showed a 2D TF mapping of two principal curvatures to color. Further possibilities of using curvature transfer functions were shown by Kindlmann et al. [14].

Using a high-dimensional transfer function for volume rendering in the traditional way involves having a large look-up table. Kniss et al. [15] showed how separable transfer functions can be used to avoid these high-dimensional look-up tables.

Tzeng et al. [16] used a high-dimensional classification for the volume visualization. Instead of a multidimensional function, they use a learning classifier. The user interaction with the transfer function is done by painting into the data slices. The main difference from traditional transfer functions is that their classifier also uses the voxel position.

Another technique aiming on simplifying the user interaction with the transfer function was shown by Tzeng and Ma [17]. They cluster the voxels into material classes considering multiple material properties. The user then interacts directly with the clusters.

2.3 Multidimensional Transfer Functions Based on Boundaries

Using transfer functions based on scalar values has a serious drawback. We 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 or the assumption that $f(\vec{x})$ is a continuous function. Fig. 1 illustrates a situation when iso-surfacing is used in order to visualize materials F_1 and F_2 . This 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 Fig. 1b, the material of intensity F_1 is missing between materials F_0 and F_2 . The iso-surfaces of values F_i (full 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

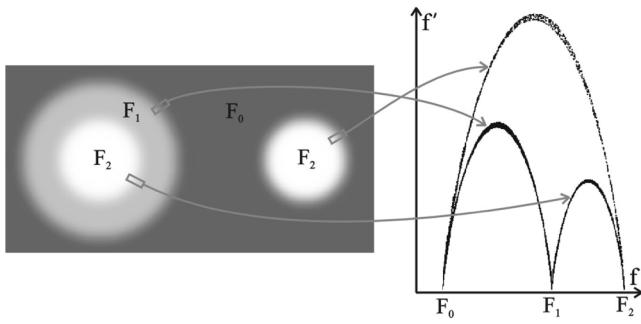


Fig. 2. Three different boundaries. Each appears as an arch.

Fig. 1b, the dashed iso-surface found due to the partial volume effect or interpolation is misleading.

In order to solve this problem, the boundary profile needs to be investigated. A common model for the boundaries in images acquired by CT and MR scanners is step edge blurred by a Gaussian, as shown by, e.g., Nickoloff and Riley [18].

Kindlmann and Durkin [2] showed that, instead of looking at how the gradient magnitude $|\nabla f|$ depends on the position across the edge, one can rather look at how it depends on the scalar value $f(\vec{x})$. The boundary appears as an arch.

They showed the arches for the volume data by creating a 2D histogram of the scalar value and the gradient magnitude. One can then distinguish between boundaries as they correspond to different arches. In Fig. 2, the arches are shown for the data set from Fig. 1. The gradient magnitude helps to distinguish between boundaries. Each boundary corresponds to a different arch.

Kniss et al. [3] used this histogram with arches as a 2D transfer function domain. They defined the transfer function by selecting peaks of the arches by using interaction widgets. An indirect selection in this space was presented by Huang and Ma [19]. They used a partial region growing in the volume. The selection was then defined by mapping the grown voxels onto the domain of arches.

Although using this domain substantially improves the selection of boundaries, it still suffers from several problems. In Fig. 3, an example is shown where two arches intersect. It is obvious that such intersections cannot be avoided. These overlaps cause ambiguities in classification of boundaries since the points at overlapping areas may belong to either of the arches. Because of this, Kniss et al. [3], [20] used, in addition, 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 effect, 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.

Recent work of Lum and Ma [4] used two samples lying in the gradient direction in order to correctly 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. The simplicity of this approach makes it suited for evaluation during

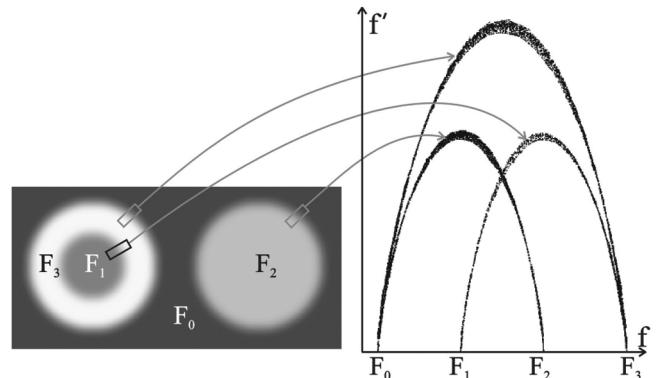


Fig. 3. Arches can cross due to the overlapping ranges of values.

rendering. However, the correct distance at which the samples should be taken is rather sensitive to the thickness of the boundary and thickness of the material. This drawback could cause misclassification of boundaries.

Our approach aims to improve the separability of the information shown by the arches. We detect the material values on both sides of the boundary. Knowing both values, we can construct a so-called LH Histogram (Serlie et al. [21]). However, it is not easy to find these values by detecting the start and the end of an arch (see Fig. 4). Serlie et al. [21] 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.

In Section 3 of this paper, we describe our method to find both boundary values. Further, in Section 4, we use these values in our transfer function.

3 THE LH HISTOGRAM

In this paper, we label the higher intensity of the two materials that form the boundary F_H and the lower intensity F_L (see Fig. 5). The LH Histogram is a 2D histogram whose axes correspond to F_L and F_H . The domain of the LH Histogram is the same as that of the Span Space [22]. However, in the Span Space, each point is

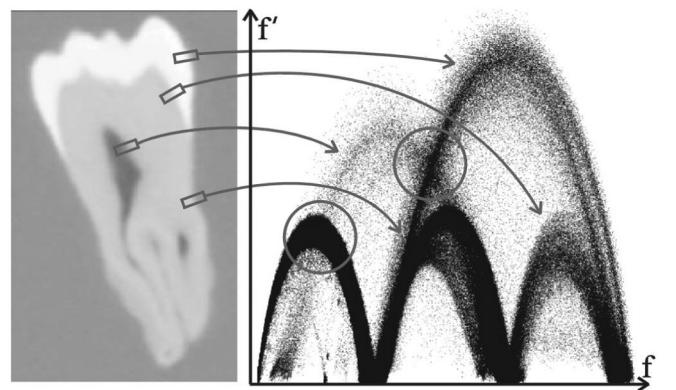
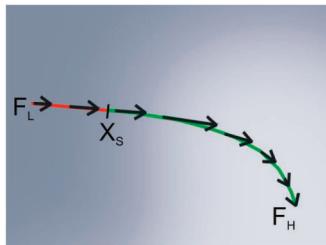
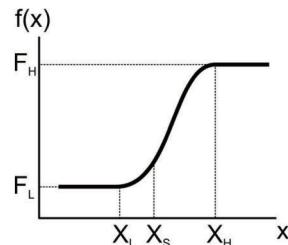


Fig. 4. CT data set of a tooth ($256 \times 256 \times 161$) with corresponding arches. The two most obvious overlaps of the arches are marked by circles. Several approaches to visualize this data set were shown in the Transfer Function Bake-Off [6].



(a)



(b)

Fig. 5. (a) Starting at position X_S , we generate a path across the boundary by integrating the gradient field. The green part of the path in the gradient direction leads to F_H . The red part generated in the opposite direction ends by finding F_L . (b) The intensity profile along the path.

indexed by the minimum and maximum values within a cell instead of the lower and higher value of materials that form the 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.

3.1 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 remainder are supposed to belong to a boundary and we continue to determine the intensities of both materials that form the boundary.

For a nonbiased 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 Fig. 5). We use 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 data sets used in this paper showed that choosing a smaller integration step does not add any observable improvements to the quality of the LH Histogram.

The integration stops when we reach F_L or F_H . The intensity profile along the path is examined while it is being constructed in order to evaluate the stopping criterion. Fig. 6a shows a common type of step edge boundary that can be easily detected by looking when the profile becomes constant. Fig. 6b and Fig. 6c 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 inflex point. In real data, we can find combinations of all these cases.

The first and second order derivatives needed in the process described above were precomputed by convolutions with Gaussian derivative kernels (see, e.g., ter Haar

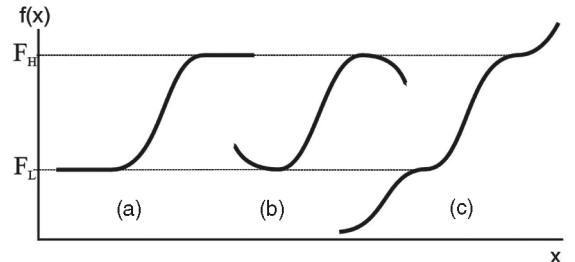


Fig. 6. Three types of intensity profiles across the boundary. Boundary ends by (a) large constant areas, (b) local extrema, or (c) inflex points.

Romeny [23]). To limit the amount of blurring introduced by the convolution, we used Gaussian with $\sigma = 1$ voxel. The typical size of our kernel was six voxels per dimension. Values between voxels were then obtained by using trilinear interpolation.

3.2 Properties

In this section, we will present some properties of the LH Histogram. In Fig. 7, an artificial data set with two concentric spheres is shown. The LH Histogram is shown in Fig. 7c. Each boundary appears as one point instead of an arch. This compact display of the boundaries allows an easier detection of boundaries. One advantage for the interactive or semiautomatic specification of the transfer function is that LH peaks are easier to select than arches.

Fig. 8 illustrates how the LH Histogram improves the separability of boundaries that were shown in Fig. 3. Whereas selecting the whole arch would take a lot of effort and would be impossible in the places where arches overlap, selecting 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 Fig. 7 is noise free. In order to simulate the behavior of the histogram in more realistic circumstances, we added 0.1 percent of Gaussian noise to the data. Fig. 9 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

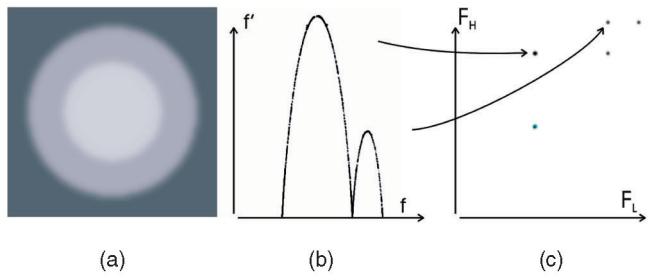


Fig. 7. Artificial data set of two spheres blurred with Gaussian. See the correspondence between the boundaries (a) in the slice, (b) in the arches, and (c) in the LH Histogram. Constant areas and boundaries in the LH Histogram appear as points. Constant areas of all three materials are projected onto the $F_L = F_H$ axis.

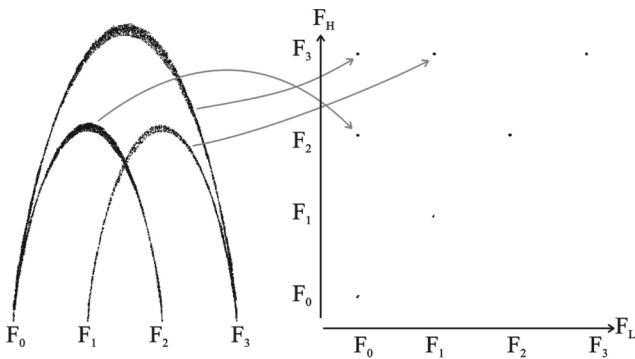


Fig. 8. Crossing arches appear in the LH Histogram as separated points.

is lowest, red highest. The amount of contributions in both histograms is shown in logarithmic scale.

2. **Bias.** Especially in MR data, we can often observe the presence of a bias field. In Fig. 10, the data from Fig. 7 is shown after applying a simple multiplicative bias field caused by one surface coil [24]. 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 very thin, we can observe that their intensity considerably changes. 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 Fig. 6b. 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.

In Fig. 11, the LH Histogram is shown of the same tooth data set as in Fig. 4. 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 of the 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.

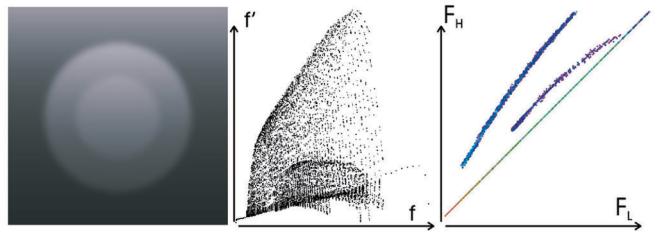


Fig. 10. Data after adding a rather strong bias. Notice that the information shown by the LH Histogram is much more compact than that shown by the arches.

4 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 (postclassification). However, we precompute them for the sake of speed (preclassification).

We can base a 2D transfer function on the LH Histogram by selecting relevant areas and by assigning them 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.

Fig. 12 shows the specification of a transfer function in the corresponding LH Histogram together with the resulting rendering. Both boundaries are selected and assigned colors. The boundary of the outer sphere is set to be semitransparent.

For achieving an interactive rendering speed, we used the VolumePro 1000 board [25]. 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. 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

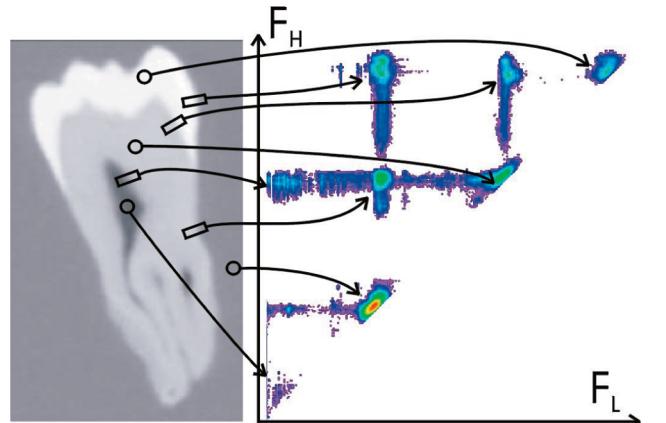


Fig. 11. LH Histogram constructed for the tooth CT data set from Fig. 4.

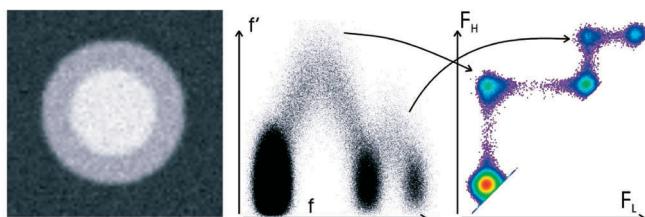


Fig. 9. Spheres after adding noise. The arches become blurred and project into the LH Histogram as blobs rather than points. The amount of contribution is shown in logarithmic scale: magenta is lowest, red highest.

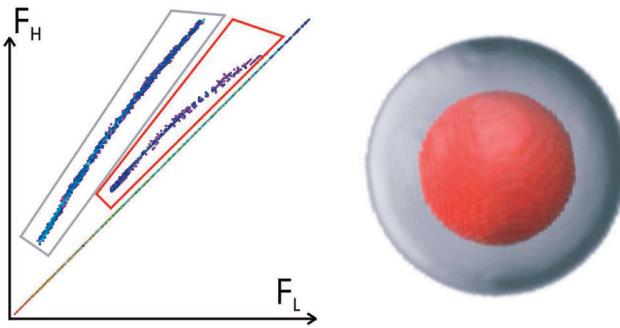


Fig. 12. The biased data set of spheres. Both elongated blobs that correspond to spheres were selected and assigned different colors for rendering.

original data for computing the gradients. The opacity is, in addition, modulated by the gradient magnitude.

5 REGION GROWING USING THE 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 [19] presented a region growing method that was aimed at facilitating good visualization. They based their cost function on the scalar value $f(\vec{x})$ 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 labeled is chosen by the user as a seed. From this seed voxel, the region is grown by repetitively labeling neighbors of already labeled voxels. In each step, a neighbor is selected that is most similar to already labeled voxels. To measure the similarity, a cost function may be 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.

5.1 Similarity Measure

We introduce a set of criteria that measure the cost of growing from a voxel to its neighbors. We evaluate 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 already labeled voxels. We use Euclidian distance in the LH space to measure the difference between two boundaries.

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

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

2. *Neighbor distance* reflects similarity between neighbors. Our experience has shown 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.

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

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 such 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 scalar value $f(\vec{x})$ 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 Fig. 4). 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 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 in the LH Histogram. However, the region growing as presented here is, rather, suited for selections of single boundaries. If the cost coefficient k_{ND} does not allow the growth of more different boundaries, the user needs to select them one by one.

6 RESULTS

In this section, we will show several visualizations by using selections in the LH Histogram or by using the region growing. For demonstration of our methods, we will show all three data sets used in the Transfer Function Bake-Off [6] (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 data set of a hand and an MRI data set of a human head.

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 is, however, not the point of this paper. For the region growing of a boundary, the user needs to point at the boundary in the data slice. Then, by using a slider, the extent of growth can be chosen.

Fig. 13 shows a visualization of a CT data set 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 the 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 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 would be rendered as transparent.

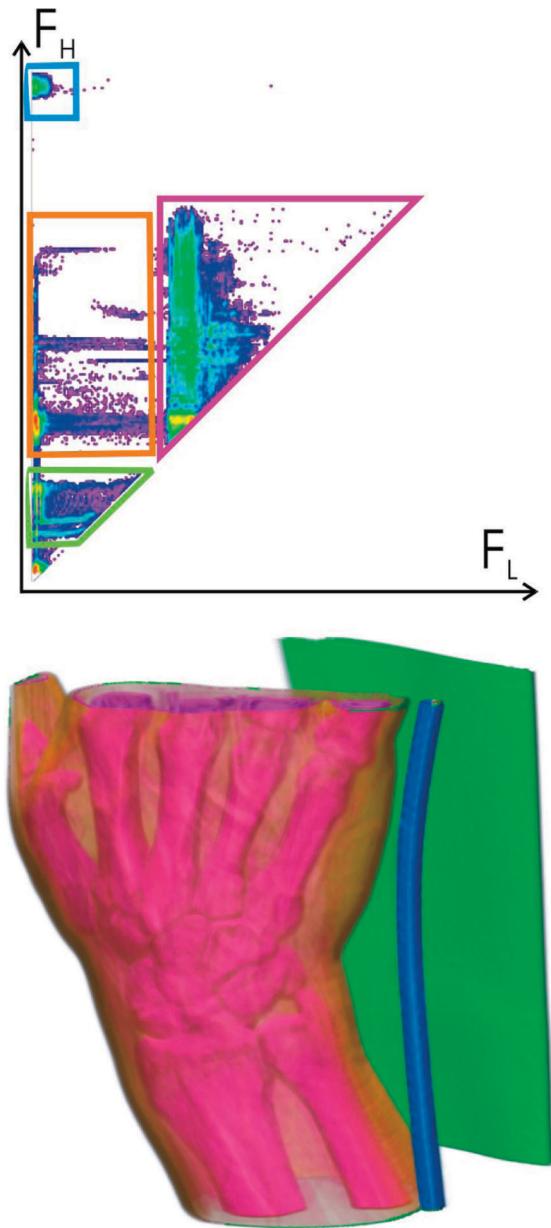


Fig. 13. Volume rendering of a CT scan of a hand ($256 \times 256 \times 232$) using a TF based on the LH Histogram.

Searching for the F_L and F_H values and constructing the LH Histogram is done in the preprocessing. Finding the F_L and F_H for the example shown in Fig. 13 took 1 minute 36 seconds. 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 Fig. 14, a 3D visualization of the tooth data set used in Fig. 4 and Fig. 11 is shown. Computation of the F_L and F_H took 1 minute 12 seconds. In 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

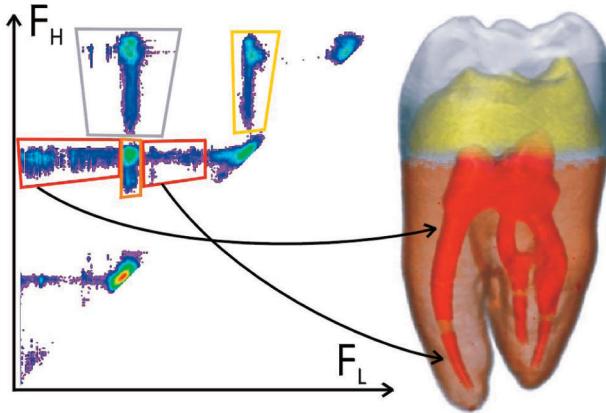


Fig. 14. CT of a tooth ($256 \times 256 \times 161$). The dentin-air (ochre) and enamel-air (white) boundaries are set to be semitransparent to reveal the inner boundaries. Note that part of the pulp-dentine boundary (red) is identical with the ochre boundary.

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. Fig. 15 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 Fig. 15b, 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



Fig. 16. The pulp boundary (red) was selected by region growing. Note that the ambiguous part of the boundary is now classified correctly. The cost coefficients were $k_{OD} = 2$, $k_{ND} = 1$, $k_{NC} = 2$, and $k_{BS} = 0$.

problem in the tooth renderings in the Transfer Function Bake-Off [6]). On the other hand, in the LH Histogram, this part is much easier to select.

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. Fig. 16 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. Also, in this case, the growth of coherent boundary was important because the dentine-air boundary appears very similar to the pulp-dentine and they get very close to each other. The Growing of the boundary took 3 seconds.

Fig. 17 shows a visualization of the MRI data set of the sheep heart. This data set 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 (Fig. 17b). 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 Fig. 17d indicate that it would not be possible to make a similar selection by using the arches.

In Fig. 18, an MRI data set 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 the 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 as good selection by using a cost function consisting of only scalar value and gradient magnitude.

Finally, we visualized the MRI data set of a knee (Fig. 19). The size of the original version was $512 \times 512 \times 87$. For the visualization we used a subsampled version of size

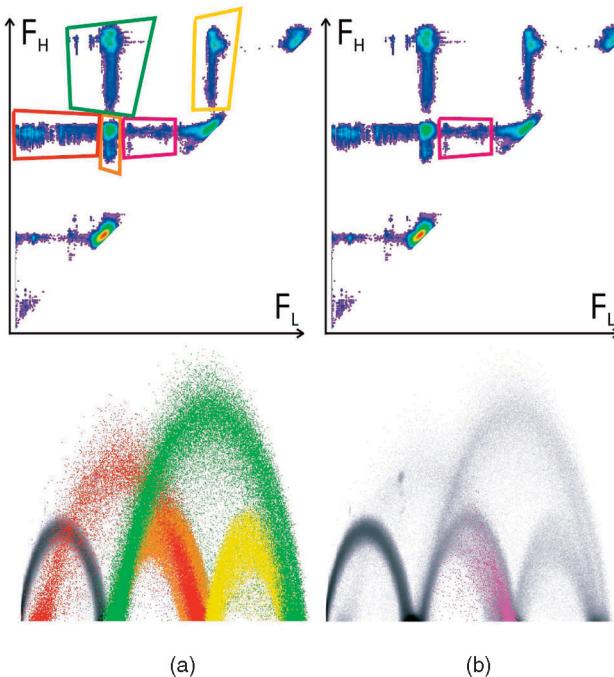


Fig. 15. (a) A selection in the LH Histogram and corresponding coloring of the arches. In (b), only the magenta selection is colored.

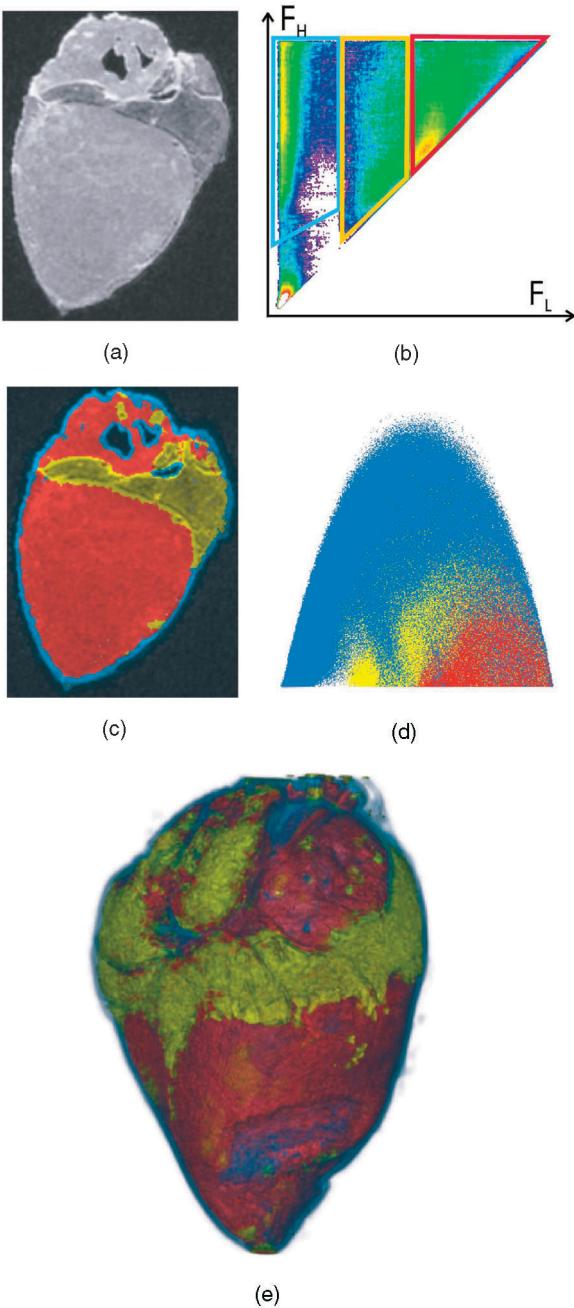


Fig. 17. MRI of a sheep heart ($352 \times 352 \times 256$). (a) The original slice, (b) the LH Histogram with three selections, (c) the slice from image (a) colored according to the selections made in the LH Histogram, (d) the corresponding coloring of arches, and (e) 3D rendering.

$256 \times 256 \times 87$. 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. Growth of each part took between 5 and 10 seconds.

7 CONCLUSIONS AND FUTURE WORK

We showed that the LH Histogram allows easier identification and selection of boundaries than we could obtain from the arches in the domain of scalar value and gradient magnitude. We have shown cases where, because of a large

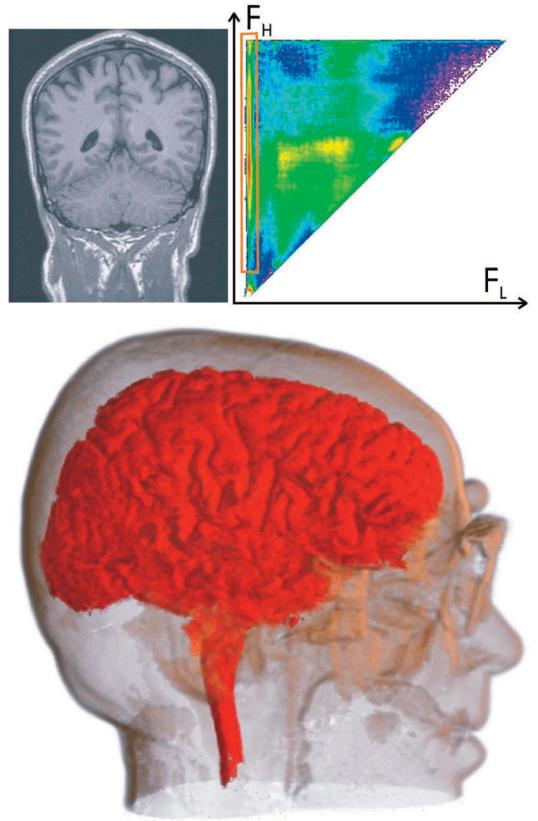


Fig. 18. MRI scan of a head ($187 \times 236 \times 253$). The air-tissue boundary was selected in the LH Histogram. The brain was selected by region growing which took 34 seconds. The cost coefficients were set to $k_{OD} = 1$, $k_{ND} = 2$, $k_{NC} = 2$, and $k_{BS} = 3$.

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 growth of boundaries distorted by several types of phenomena. Conventional approaches that use only scalar value and gradient magnitude can be easily misled in cases where the combination of $f(\vec{x})$ and $|\nabla f|$ belongs to more boundaries. Our approach has the potential to resolve such ambiguities.

Moreover, in conventional techniques, the user is often asked to enter the range of possible scalar values 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.

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.

In our future work, we would like to focus on the use of the LH method in the automated design of transfer functions. We expect that the LH Histogram will facilitate such automation. Further, we aim to combine selections in

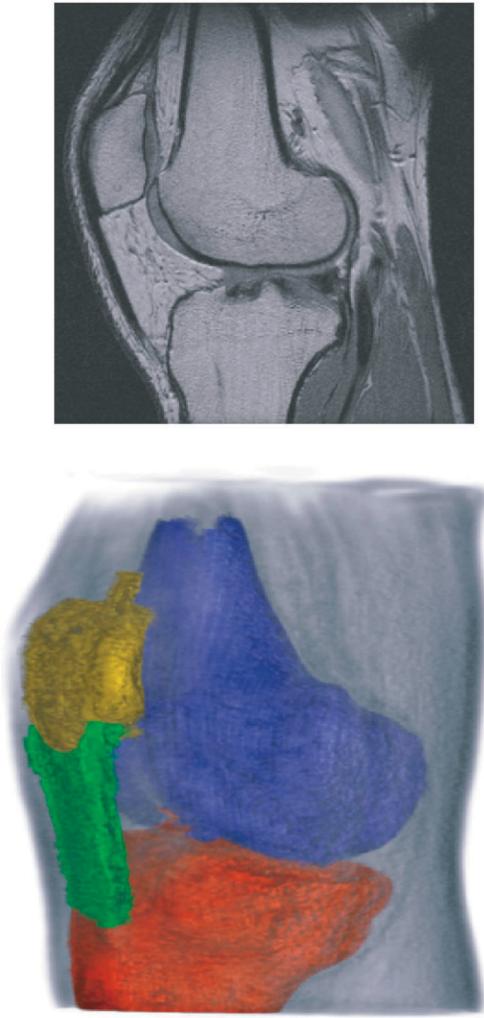


Fig. 19. MRI scan of a knee ($256 \times 256 \times 87$). The colored segments were obtained by using the region growing. The cost coefficients were $k_{OD} = 1$, $k_{ND} = 2$, $k_{NC} = 4$, and $k_{BS} = 1$.

the LH Histogram with the region growing technique in order to allow fast and flexible selections of boundaries.

The model used in this paper 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. [21] 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.

We also want to improve the robustness of the presented method for estimating the F_L and F_H values in order to improve compactness of the LH Histogram, especially in the presence of noise. A faster estimation of F_L and F_H would allow a postclassification rendering.

To conclude, the LH Histogram is a promising domain that has the potential of facilitating the visualization of volume data.

ACKNOWLEDGMENTS

The work presented in this publication has been funded by Philips Medical Systems in Best, The Netherlands.

REFERENCES

- [1] M. Levoy, "Display of Surfaces From Volume Data," *IEEE Computer Graphics and Applications*, vol. 8, no. 3, pp. 29-37, 1988.
- [2] G. Kindlmann and J.W. Durkin, "Semiautomatic Generation of Transfer Functions for Direct Volume Rendering," *Proc. IEEE Symp. Volume Visualization*, pp. 79-86, 1998.
- [3] J. Kniss, G. Kindlmann, and C. Hansen, "Interactive Volume Rendering Using Multidimensional Transfer Functions and Direct Manipulation Widgets," *Proc. IEEE Visualization Conf.*, pp. 255-262, 2001.
- [4] E.B. Lum and K.L. Ma, "Lighting Transfer Functions Using Gradient Aligned Sampling," *Proc. IEEE Visualization Conf.*, pp. 289-296, 2004.
- [5] S. Fang, T. Biddlecome, and M. Tuceryan, "Image-Based Transfer Function Design for Data Exploration in Volume Visualization," *Proc. IEEE Visualization Conf.*, pp. 319-326, 1998.
- [6] 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.
- [7] C.L. Bajaj, V. Pascucci, and D. Schikore, "The Contour Spectrum," *Proc. IEEE Visualization Conf.*, pp. 167-174, 1997.
- [8] V. Pekar, R. Wiemker, and D. Hempel, "Fast Detection of Meaningful Isosurfaces for Volume Data Visualization," *Proc. IEEE Visualization Conf.*, pp. 223-230, 2001.
- [9] I. Fujishiro, T. Azuma, and Y. Takeshima, "Automating Transfer Function Design for Comprehensible Volume Rendering Based on 3D Field Topology Analysis," *Proc. IEEE Visualization Conf.*, pp. 467-470, 1999.
- [10] T. He, L. Hong, A. Kaufman, and H. Pfister, "Generation of Transfer Functions with Stochastic Search Techniques," *Proc. IEEE Visualization Conf.*, pp. 227-234, 1996.
- [11] J. Marks, B. Andelman, 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," *Proc. SIGGRAPH Conf.*, pp. 389-400, 1997.
- [12] A. König and E. Gröller, "Mastering Transfer Function Specification by Using VolumePro Technology," *Proc. Spring Conf. Computer Graphics*, vol. 17, pp. 279-286, 2001.
- [13] J. Hladívka, A. König, and E. Gröller, "Curvature-Based Transfer Functions for Direct Volume Rendering," *Proc. Spring Conf. Computer Graphics*, vol. 16, pp. 58-65, 2000.
- [14] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Möller, "Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications," *Proc. IEEE Visualization Conf.*, pp. 513-520, Oct. 2003.
- [15] J. Kniss, S. Preimoze, M. Ikits, A. Lefohn, C. Hansen, and E. Praun, "Gaussian Transfer Functions for Multifield Volume Visualization," *Proc. IEEE Visualization Conf.*, pp. 497-504, 2003.
- [16] F.Y. Tzeng, E.B. Lum, and K.L. Ma, "A Novel Interface for Higher-Dimensional Classification of Volume Data," *Proc. IEEE Visualization Conf.*, pp. 505-512, 2003.
- [17] F.Y. Tzeng and K.L. Ma, "A Cluster-Space Visual Interface for Arbitrary Dimensional Classification of Volume Data," *Proc. Eurographics/IEEE TCVG Visualization Symp. (VisSym)*, pp. 17-24, 2004.
- [18] 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.
- [19] R. Huang and K.L. Ma, "RGV IS: Region Growing Based Visualization Techniques for Volume Visualization," *Proc. Pacific Graphics Conf.*, pp. 355-363, 2003.
- [20] J. Kniss, G. Kindlmann, and C. Hansen, "Multidimensional Transfer Functions for Interactive Volume Rendering," *IEEE Trans. Visualization and Computer Graphics*, vol. 8, no. 3, pp. 270-285, 2002.
- [21] 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 Proc.*, part 2, pp. 175-183, 2003, <http://www.ph.tu.tedelft.nl/iwo/frames.html>.
- [22] H.W. Shen, C. Hansen, Y. Livnat, and C.R. Johnson, "Isosurfacing in Span Space with Utmost Efficiency (Issue)," *Proc. IEEE Visualization Conf.*, pp. 287-294, 1996.
- [23] B.M. ter Haar Romeny, *Front-End Vision and Multiscale Image Analysis*. Kluwer Academic, 2003.

- [24] M.T. Vlaardingerbroek and J.A.D. Boer, *Magnetic Resonance Imaging: Theory and Practice*. Springer-Verlag, 1999.
- [25] H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler, "The VolumePro Real-Time Ray-Casting System," *Proc. 26th Ann. Conf. Computer Graphics and Interactive Techniques*, pp. 251-260, 1999.



Petr Šereda received the Ing. (MSc) degree in 2002 from the University of West Bohemia in the field of computer graphics and visualization. He is a PhD student in the Biomedical Engineering Department at the Eindhoven University of Technology. His research interests include 3D volume visualization of medical data and flexibility and automation of the visualization process. He is particularly focused on transfer functions.



Iwo W.O. Serlie received the MSc degree in technical informatics from the Delft University of Technology. He is a PhD student in the Quantitative Imaging group of the Department of Imaging Science and Technology at the Delft University of Technology. His research interests include medical image analysis, visualization via the unfolded cube method, and cleansing for virtual colonoscopy.



Anna Vilanova Bartrolí received the PhD degree in 2001 from the Vienna University of Technology. She is an assistant professor in the Biomedical Image Analysis section of the Biomedical Engineering Department at the Eindhoven University of Technology. Her research interests include medical visualization, volume visualization, tensor visualization, and medical image analysis.



Frans A. Gerritsen received the PhD degree from the Applied Physics Department at Delft University of Technology in December 1982. He is a part-time professor in the Biomedical Engineering Department of Technische Universiteit Eindhoven as well as a full-time director of Research and Advanced Development of the business group Medical IT at Philips Medical Systems in Best, The Netherlands. His research interests include clinical applications of medical image processing and, especially, computer-aided detection, quantification, and diagnosis.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.