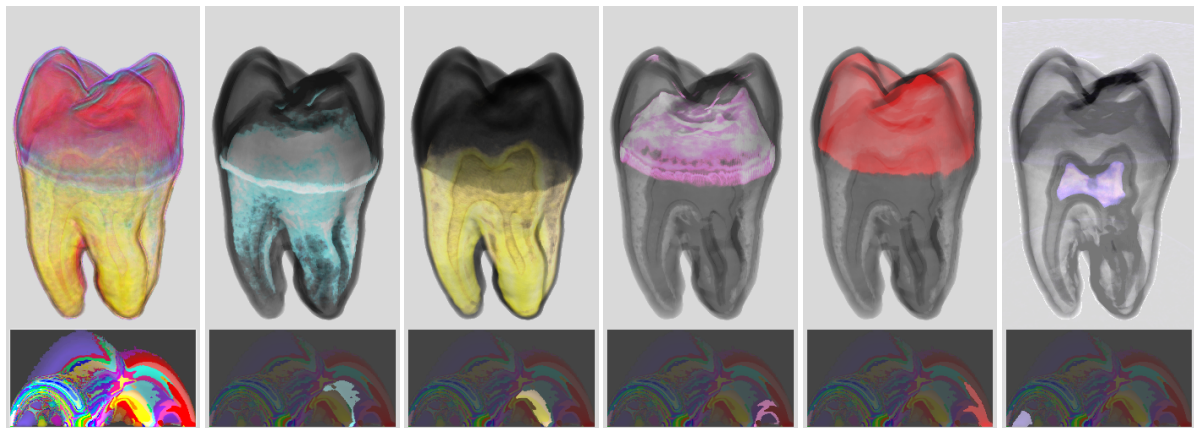# Spatialized Transfer Functions

Stefan Roettger, Michael Bauer and Marc Stamminger

Computer Graphics Group, University of Erlangen, Germany

Left: automatic visualization of a tooth. To the right: interactive selection of dentin, dentin boundary, enamel boundary, enamel and nerve cavity. Each feature is selected by pointing and clicking at the corresponding high-lighted region in the transfer function below each image.

Abstract

Multi-dimensional transfer functions are an efficient way to visualize features in scalar volume data produced by CT or MRI scanners. However, the optimal transfer function is difficult to find in general. We present an automatic yet powerful method for the automatic setup of multi-dimensional transfer functions by adding spatial information to the histogram of a volume. Using this information we can easily classify the histogram and derive a transfer function by assigning unique colors to each class of the histogram.

Each feature can be selected interactively by pointing and clicking at the corresponding class in the transfer function. In order to render the classified volume with adequate quality we propose an extension of the well-known pre-integration technique. Furthermore, we demonstrate the flexibility of our approach by giving examples for the imaging of segmented, diffusion-tensor and multi-modal data.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

1. Introduction and Previous Work

Direct volume rendering techniques [Rez00] are widely used for the display of medical volume data generated by computed tomography (CT) or magnetic resonance imaging (MRI). In comparison to the display of cross-sections which are still the standard visualization tool in clinical practice, the main problem of direct volume rendering is that features may occlude each other. Therefore, the efficiency and accep-

tance of direct volume rendering techniques depends on the ability to isolate a specific feature.

For this purpose, clipping methods either apply a clipping plane, a volume manipulator [MLB03] or a voxelized clip geometry [WEE02, RGWE03] to cut away occluding material.

Since a CT or MRI scanner generates a scalar volume, transfer functions [WM92, Max95] are needed to assign specific colors and opacities to the range of scalar values. Manually steered methods for the setup of transfer functions give feedback about a region of interest in the volume in order to find the corresponding region in the domain of the transfer function [KKH01, HM03]. Then a specific feature can be emphasized by adding emission and opacity to the corresponding region of the transfer function.

Another manual approach is the generation of a variety of random transfer functions. Then the user selects the transfer function which yields the best result. This procedure is repeated until the transfer function converges against a useful setup [HHKP96, MAB*97].

Semi-automatic setup methods analyze statistical properties to identify interesting regions. The *contour spectrum* [BPS97], for example, locates scalar values which correspond to uniquely shaped iso surfaces [LC87], so that the user can select an appropriate iso value. Other approaches automatically adapt the opacity function to make uninteresting regions more transparent than regions with high information content [KD98, TLMM02, PM04].

1.1. Multi-dimensional Transfer Functions

A fundamental drawback of transfer function based methods is the fact that the mapping from the domain of the volume to the domain of the transfer function is injective. Therefore the separation of features with the same statistical properties is impossible. For example, tissue can be separated from bone in CT scans, but a single bone cannot be separated from other bones.

The flexibility of transfer functions has been improved by the introduction of *multi-dimensional transfer functions* [KD98]. These do not only depend on the scalar function but also consider its higher order derivatives. This allows for better feature localization in the domain of the transfer function.

The most widely-used type are two-dimensional transfer functions (e.g. 2D opacity maps) which depend on the scalar value and the gradient magnitude. Ten-dimensional types [TM03] and curvature based transfer functions [HKG00, KWTM03] have also been described, recently.

1.2. Transfer Function Setup

On the one hand, multi-dimensional transfer functions offer more degrees of freedom to mask certain properties than a standard transfer function. On the other hand they are also more difficult to setup by the user. Therefore, the actual research is concentrating on automatic methods to ease this setup process.

Due to the inclusion of the gradient magnitude, material boundaries can be visualized with 2D opacity maps. The material boundaries correspond to arcs in the two-dimensional histogram or scatter plot which connect the footprints of the materials (compare left images in Figure 1).

In practice, however, these arcs can be more or less pronounced or even indistinguishable. This mainly depends on the imaging quality of the scanner. Reconstruction and quantization artifacts of the CT scanner and the density distribution of the scanned object also have a great influence on the shape of the arcs.

A good strategy to achieve a meaningful visualization is to mark the top of the arcs with different colors. If the arcs are not well defined it is therefore challenging to find a meaningful transfer function setup.

This is especially true for MRI scans. As a fundamental property, an MRI scanner cannot distinguish between air and bone. Thus, the histogram usually shows only one large unpronounced arch between the footprints of water and non-water bearing material. In such a case the setup of a two-dimensional transfer function requires a high degree of practical knowledge since the distribution of scalar values and gradients in the scatter plot provides only very few clues for the separation of features.

In other words, the ability to isolate a feature in the transfer function domain depends on how much a feature shares its statistical footprint in the histogram with other features. If there is an overlap in the distribution of two features they cannot be separated completely.

1.3. Segmentation

In order to circumvent this fundamental restriction, an explicit segmentation needs to be performed on the domain of the volume. A variety of segmentation algorithms [GW01] have been developed for a broad range of application areas. Most algorithms are based on a region growing or diffusion process.

Although hardware-accelerated methods have been presented recently [VKG03, SHN03, HBH03], processing times are still too slow for the purpose of interactive exploration. For this reason, segmentation methods are usually not utilized to give a quick first impression of a dataset. This is still the domain of transfer function based methods.

2. Spatialized Transfer Functions

In this paper we present a new and fast method for the automatic setup of multi-dimensional transfer functions. Our strategy is described as follows: we do not try to identify a single specific feature because this may not be possible at all with transfer functions. We rather automatically separate as many features as possible using what we call *spatialized transfer functions* (STF).

As the next step, a specific target feature can be isolated interactively by removing all the features from the STF which do not include the target. Then this scenario is an excellent starting point for further exploration or segmentation as described in Section 4.

2.1. Basic Principle

At the IEEE Visualization 2001 conference G. Kindlmann noted in his tutorial [Kin01] that "histograms/scatterplots entirely loose spatial information" and asked if there would be "any way to keep some of it?". Spatial information is important, because a feature by definition is a spatially connected region in the volume domain with a unique position and certain statistical properties. If we only use the statistical information of the scatter plot we effectively ignore the most important part of a features definition.

A naive approach to enrich transfer functions with spatial information would be to use a 5D transfer function depending on the scalar values, gradients, and the volume coordinates *x*, *y* and *z*. However, such a high-dimensional transfer function would be even more difficult to setup than a 2D transfer function.

At this point we would like to recall that the standard procedure to separate features manually is to mask the opacity at interesting regions and to give distinct features different colors. On the one hand, interesting regions basically are regions with a high gradient in the scatter plot, so the statistical information in the histogram is suitable for the setup of the opacity. On the other hand, distinct features by definition are distinguished by their position in space, so we need to use the spatial information for the setup of the color in the transfer function. As a result, we do not require a 5D transfer function, but the missing three dimensions correspond to the RGB color channels of a standard 2D transfer function.

In general, we say that a *spatialized transfer function* is a one- or more-dimensional transfer function, where spatial information has been used to derive the color, whereas statistical (and/or spatial information) is used to setup the opacity. In the following section we describe an automatic setup procedure for this type of transfer function.

2.2. Automatic Transfer Function Setup

As the first step, we describe the automatic setup of the opacity values of a 2D transfer function. Let $c_\alpha > 0$ be a con-
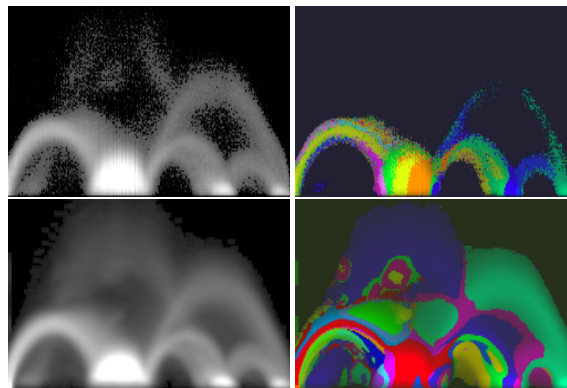
Figure 1: *2D Histograms based on scalar values on the x-axis and gradients on the y-axis (left) and corresponding spatialized transfer functions for $r = 0.1$ (right). In the bottom row noise has been smoothed out by using two-times super-sampling and a k-neighborhood with $k = 2$.*

stant defining the global opacity, then the opacity of the entry $F(s,t)$ of the transfer function with $s \in [0,1]$ being the normalized scalar value and $t \in [0,1]$ being the normalized gradient magnitude is $F_\alpha(s,t) = tc_\alpha$.

This has the effect that we blend out uninteresting regions with low gradients and emphasize the material boundaries with high gradients. This is also known as gradient weighted rendering. More advanced opacity setup methods [KD98, PM04] could also be used.

As the second step, the separation of the features is performed using the color channel of the 2D transfer. As a matter of fact, a feature is uniquely determined by its position in space. Thus, we need to find a mapping which transforms unique features in space to unique colors in the transfer function. In particular, we would like to assign the same color to all entries in the transfer function which map to almost the same position in the volume domain.

A mapping with the desired properties is described in the following. Let $p_i(s,t)$, $i = 1..n$, be the normalized positions of the n contributing voxels of the entry $H(s,t) = n$ of the histogram, let $b(s,t) = \frac{1}{n}\sum_{i=1}^n p_i(s,t)$ be the barycenter of the voxels, let $v(s,t) = \frac{1}{n}\sum_{i=1}^n ||p_i(s,t) - b(s,t)||$ be the spatial variance of the voxels and let r be the maximum radius of the features to be detected. The barycenters and the variance are computed once for each volume in a preprocessing step.

If we start with a reference tuple $T_0 = (s,t)$ all other tuples T with $||b(T) - b(T_0)|| < r$ belong to the same feature, assumed that the variance $v(T)$ of the tuples is sufficiently small. With increasing variance $v(T)$ the relationship between the barycenters $b(T)$ and the real location of a feature is becoming more and more imprecise, especially if the

spatial variance exceeds the feature radius r. However, the distance norm

$$N(T, T_0) = ||b(T) - b(T_0)|| + |v(T_0) - v(T)|$$

is a reliable measure for spatial correspondence, that is whether or not a tuple T belongs to the same feature as the reference tuple T_0. Based on this norm $N(T, T_0)$ we now classify all entries of the transfer function into groups that belong to the same feature (as far as we can tell without performing an explicit time consuming segmentation). Let c_ε be a constant which defines the global emission and let C_{RGB} be a RGB color triple with random hue value, then each group entry T is given the following emission:

$$\forall T : N(T, T_0) < r, \quad F_\varepsilon(T) := C_{RGB} c_\varepsilon$$

For simplicity we start with the reference tuple which has the highest histogram count. Then we compute all entries which belong to the same group as the reference tuple and assign a specific color to all these entries. This procedure is repeated for the remaining entries until the classification is complete.

As an example, the color channel F_ε of the transfer function for $r = 0.1$ of the Tooth dataset [PLB*01] is shown in Figure 1. On the top left the original 2D histogram is depicted. On the top right the intensity peaks of the scatter plot are clearly broken down in uniquely colored regions which correspond to different tooth materials and their respective boundaries.

An automatic visualization of the Tooth dataset using the STF is shown in the leftmost image of the Teaser. The dentin, the enamel and the boundary between the two materials are colored automatically. The only parameter we need to choose manually is the maximum feature radius r (and once for each dataset the global opacity c_α and the global emission c_ε).

We usually start with a moderately large feature radius and slowly decrease the radius until the desired features can be separated from each other. In our experiments a radius between 0.05 and 0.25 was a good choice. Note that the complex shapes of the classes are detected automatically using the spatial information in the histogram. Without the aid of the the STF the exact shape is almost impossible to setup manually.

2.3. Interactive Exploration

For further exploration we can now point and click into the histogram to select a specific class. Then the corresponding feature is displayed by setting the emission of the remaining classes to zero. In the Teaser this interactive process is illustrated by high-lighting the most important feature classes of the Tooth dataset.

Using the described spatialization of a transfer function we effectively perform a segmentation on the domain of the transfer function. The main advantage of this approach is that we do not need to perform an explicit time-consuming segmentation of the volume domain. If the feature radius is changed interactively we only need to reclassify all entries in the transfer function. Since we do not need to reprocess all voxels explicitly, the total update time including a redraw of the volume is well below a quarter of a second (on a PC with AMD Athlon 1GHz and NVIDIA GeForceFX 5800 Ultra).

2.4. Pseudo-Shading

In order to enhance the discrimination of features, it turns out that we can optionally apply a simple technique, which we call pseudo-shading.

At the boundary of an object the scalar values quickly drop to zero. If we decrease the emission for the lowest occurring scalar values of the object, the features silhouette is rendered dark, since the accumulated opacity becomes predominant over the emission. This effectively simulates the appearance of a head light. In the context of the STF we implement pseudo-shading by first determining the scalar range of each class. The we attenuate the emission of each class using a linear ramp over its scalar range.
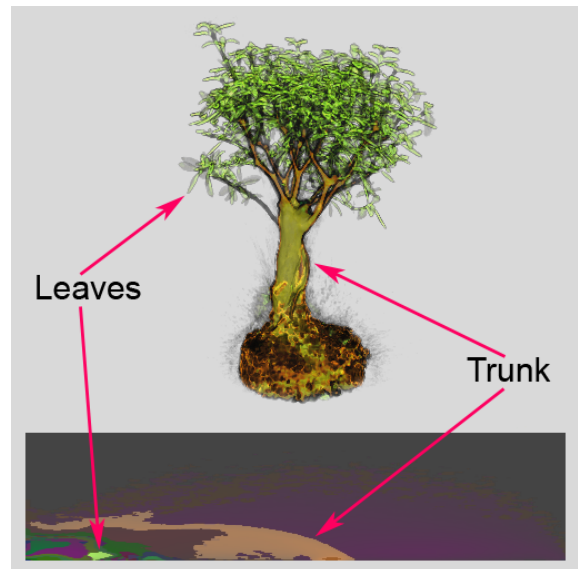

Figure 2: *STF based visualization of the Bonsai #1 dataset [Roe04] using pseudo-shading. The leaves (green) and the trunk (brown) have been selected by high-lighting the corresponding classes in the STF.*

As an example, the Bonsai dataset [Roe04] has been visualized with pseudo-shading in Figure 2. The leaves and the trunk were selected by pointing and clicking at the corresponding high-lighted classes in the STF as depicted at the

bottom of the figure. For each selected class the emission was attenuated linearly which lead to the shaded appearance of the bonsai.

2.5. Reduction of Noise

For the most prominent peaks in the histogram the histogram distribution is sufficiently smooth. But usually there also exist underrepresented regions where the histogram is very noisy. If the number of voxels per histogram entry is very low (e.g. below 5-10 voxels), a valid classification cannot be made. In such a case the opacity of these entries needs to be set to zero.

In order to enable a correct classification throughout the entire range of the histogram we need to reduce the noise by adding more samples to the histogram. First, we increase the number of samples by performing super-sampling. For example, two-times super-sampling increases the number of samples in the histogram by a factor of eight. In our practical tests, however, this was not fully sufficient, so we tried four-times super-sampling. This resulted in a sufficient reduction of the noise but the preprocessing time was also increased dramatically, since a huge number of trilinear interpolations had to be performed.

As an alternative, we may add each voxel to a k-neighborhood in the histogram, since coherent regions in the volume translate to coherent regions in the transfer function. A k−neighborhood with $k = 1$ increases the number of samples by a factor of 9. Using both a k−neighborhood with $k = 2$ and two-times super-sampling, noisy regions are smoothed out well, while at the same time the number of trilinear interpolations is kept reasonable small. The noise reduction and the corresponding improvement of the classification is demonstrated in Figure 1. The top row shows the original noisy histogram and the bottom row the smoothed version.

3. Implementation

In order to display a regular scalar volume we use a 3D texture based approach with view-port aligned slices. We also apply the pre-integration technique [MHC90, RKE00] to achieve high-quality volume visualizations. This has first been demonstrated by Engel et al. [EKE01] for 1D transfer functions using a 2D pre-integration table. More recently, pre-integrated shading [MGS02, LWM04] has been proposed as an extension to allow high-quality volume lighting.

In the case of 2D transfer functions, however, it is unclear how to apply the pre-integration technique, since this would require a 4D pre-integration table. Kniss et al. [KPI*03] proposed to use a special type of 2D transfer function representation based on the sum of Gaussians, which can be pre-integrated analytically in the texture shader. For spatialized

2D transfer functions, however, this solution is not feasible, since STF cannot be approximated efficiently by a sum of Gaussians.
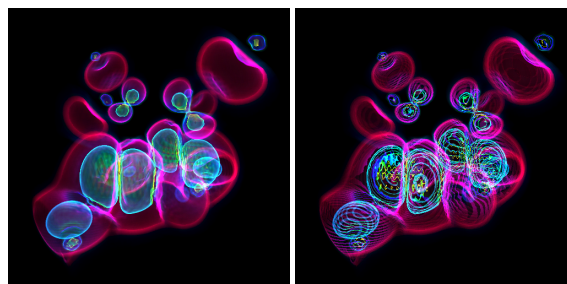

Figure 3: *Benefit of pre-integrating along the scalar axis of 2D transfer functions. A pre-integrated visualization of the Neghip dataset [Roe04] is shown on the left. Highest gradients are depicted in yellow. On the right slicing artifacts are clearly visible which are due to disabled pre-integration.*

Instead, we choose a rather pragmatic approach: since a four-dimensional pre-integration table is infeasible we use a three-dimensional table where each slice contains the 2D pre-integration table for a constant gradient magnitude. This simplification is valid since the highest frequencies of the ray integral come from high frequencies along the scalar axis of the 2D transfer function. Along the gradient axis the frequencies are usually lower, so that we may assume a constant gradient for each ray segment. Figure 3 shows the effect of this approach.

4. Practical Applications

In the previous sections, we demonstrated that the spatialization of transfer functions is a powerful method for the automatic setup of multi-dimensional transfer functions. In this section we substantiate this by giving a variety of practical examples.

4.1. STF Guided Segmentation

Our first practical example is the application of spatialization to segmentation: As mentioned previously, the ability to locate features in the STF is limited by the injective mapping. While the Tooth and the Bonsai dataset have few injective regions, other datasets may require an explicit segmentation. But if a specific feature is not directly localizable in the STF, this is often due to the fact that it is only one out of many similar objects with the same material properties. This case is often encountered in medical imaging, where, for example, bone fractures need to be analyzed. While the different parts of the fracture are not separable in the transfer function, the complete bone structure is very well distinguishable from other materials like tissue, fat or muscles.
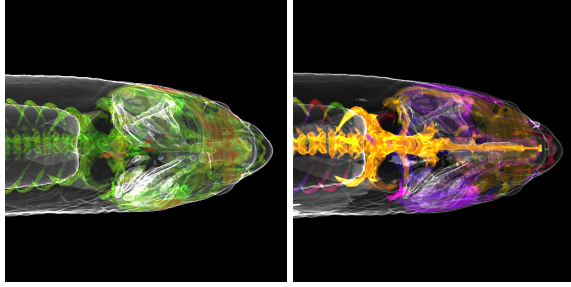
Figure 4: *STF guided segmentation of the Carp. In the left image the bones were selected in the STF (the skin is additionally shown in white). Then the Carp was segmented by performing region growing on the selected parts. Each segment is given a random hue value. After that the orange spine segment was picked and emphasized. As a result, the spine protrusion is clearly visible. Otherwise it would be obscured by the head bones.*

Our proposal for the separate visualization of multiple objects with the same characteristics is to first to select all objects in the STF. Then the selected objects are characterized by a non-zero opacity entry in the transfer function. Therefore we can obtain a segmentation by performing region growing on the voxels with non-zero opacity. This has the effect that the objects are broken down into individual components according to their spatial connectivity. This special type of segmentation which we call *STF guided segmentation* is extremely fast, since for each voxel only one lookup into the 2D opacity map is required.

Each detected segment is assigned a unique random tag which determines the hue value of the segment. Now we can click into the render view and select a segment by its hue value, just as we clicked into the STF to select features by their color. An example is given in Figure 4 where the spine protrusion of the Carp dataset has been made visible.

4.2. Visualization of Aneurysms

Our second practical example is the semi-automatic detection of aneurysms in CT angiography data (CTA): In our example the aneurysm is the small red blob in the middle of the brain as depicted in the top left image of Figure 5. It corresponds to the tiny red spot at the bottom of the STF. The small brown protrusion above the red spot maps to the interior of the arteria in the brain. Without the aid of the STF it takes some time to locate this yellow spot, because the slightest offset also selects the arteria. Using the STF the shape of the spot is found automatically and the aneurysm can be visualized just by clicking at it. We tested a variety of angiography datasets with aneurysms. In all cases, the aneurysm was instantly visible as a small characteristic spot in the STF.

4.3. Improved Magnetic Resonance Imaging

Our third practical example is the improved visualization of brain MRI data: Here, one principal problem is that the skull is pictured very badly, since it maps to almost the same scalar values as air. The same problem arises for the brain, because it has similar properties as skin and tissue. So it is usually very challenging to visualize both the skull and the brain separated from other tissue. But if we do not use a standard 2D transfer function (scalar values and gradients) but a transfer function based on both the T1-weighted and the proton-density-weighted response of the MRI scanner we get a much better result: in the STF two regions which map to the skull and the brain quite clearly stand out. In the top right image in Figure 5 these two regions have been highlighted (x-axis = T1, y-axis = PD). Using the STF is not aimed as a replacement for a real brain segmentation but it is a convenient way to display the brain and the skull without much effort.

4.4. Diffusion Tensor Imaging

Our fourth practical example is the visualization of nerve pathways using diffusion tensor imaging: The standard procedure to visualize the nerve pathways in a DTI scan is to trace path lines along the largest eigenvector of the tensor field. The pathways are discriminated from nerve cells by considering the so-called fractional anisotropy of the diffusion tensor. High anisotropy values are characteristic for the pathways (white matter), whereas nerve cells have low anisotropy (grey matter). We use a 2D transfer function based on both the scalar values and the fractional anisotropy. Here, the pathways correspond to a characteristic region in the upper middle of the transfer function. This can be seen in the bottom left image of Figure 5. Another characteristic region is the ventricle in the lower right area of the transfer function.

4.5. Registered Multi-Modal Imaging

Our fifth practical example is the visualization of registered multi-modal data: In medical imaging tumors often have to be visualized. For this purpose it is not sufficient to aquire MRI data alone in order to see the tumor. Due to the requirements of surgical planning the skull must also be visible. Thus, a CT scan needs to be made and registered with the MRI data. This is usually the starting point for an explicit segmentation based on both the two multi-modal datasets. Using a 2D transfer function based on the two scan modes we can instantly select the tumor and the surrounding bone. The tumor corresponds to the tiny yellow spot whereas the skull corresponds to a large green area as shown in the bottom right image in Figure 5.

In summary, the STF is a very convenient means to visualize features in scalar, diffusion tensor or multi-modal volume data. If a feature has a characteristic foot print in the
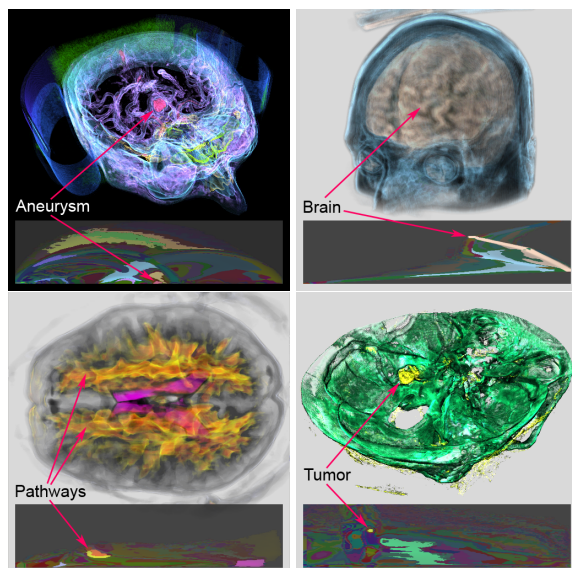
Figure 5: *STF example visualizations from top left to bottom right: visualization of an aneurysm (CTA), separation of brain and skull (MRI/T1+PD), identification of nerve pathways (DTI/MRI+FA), visualization of tumor and skull (CT+MRI).*

domain of the transfer function, the feature can be displayed very quickly by selecting the corresponding class in the STF. This has been shown to work remarkably well in a variety of medical setups.

5. Conclusion

In this paper we have shown how to include spatial information into multi-dimensional transfer functions. This has enabled us to come up with an automatic transfer function setup scheme. The display of specific features is very natural in this context: we can select features just by pointing and clicking into the 2D histogram. We have also given a variety of examples that underline the applicability of our approach to a broad range of issues in medical visualization.

6. Acknowledgements

References

[BPS97] BAJAJ C. L., PASCUCCI V., SCHIKORE D.: The Contour Spectrum. In *Proc. Visualization '97* (1997), IEEE, pp. 167–174.

[EKE01] ENGEL K., KRAUS M., ERTL T.: High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading. In *Eurographics Workshop on Graphics Hardware '01* (2001), ACM SIGGRAPH, pp. 9–16.

[GW01] GONZALEZ R., WOODS R.: *Digital Image Processing*, 2nd ed. Prentice Hall, Upper Saddle River, New Jersey, 2001.

[HBH03] HADWIGER M., BERGER C., HAUSER H.: High-Quality Two-Level Volume Rendering of Segmented Data Sets on Consumer Graphics Hardware. In *Proc. Visualization '03* (2003), IEEE, pp. 301–308.

[HHKP96] HE T., HONG L., KAUFMAN A., PFISTER H.: Generation of Transfer Functions with Stochastic Search Techniques. In *Proc. Visualization '96* (1996), IEEE, pp. 227–234.

[HKG00] HLADUVKA J., KOENIG A., GROELLER E.: Curvature-Based Transfer Functions for Direct Volume Rendering. In *Proc. SCCG '00* (2000), pp. 58–65.

[HM03] HUANG R., MA K.-L.: RGVis: Region Growing Based Techniques for Volume Visualization. In *Proc. Pacific Graphics '03* (2003), pp. 355–363.

[KD98] KINDLMANN G., DURKIN J. W.: Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering. In *Proc. Visualization Symposium '98* (1998), pp. 79–86.

[Kin01] KINDLMANN G.: IEEE Visualization 2001 Tutorial 5, Presentation 3: Transfer Functions for Direct Volume Rendering. In *Proc. Visualization '01* (2001).

[KKH01] KNISS J., KINDLMANN G., HANSEN C.: Interactive Volume Rendering Using Multi-Dimensional Transfer Functions and Direct Manipulation Widgets. In *Proc. Visualization '01* (2001), IEEE, pp. 255–262.

[KPI*03] KNISS J., PREMOZE S., IKITS M., LEFOHN A., HANSEN C., PRAUN E.: Gaussian Transfer Functions for Multi-Field Volume Visualization. In *Proc. Visualization '03* (2003), pp. 497–504.

[KWTM03] KINDLMANN G., WHITAKER R., TASDIZEN T., MOELLER T.: Curvature-Based Transfer

Functions for Direct Volume Rendering: Methods and Applications. In *Proc. Visualization '03* (2003), pp. 513–520.

[LC87] LORENSEN W. E., CLINE H. E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *ACM Computer Graphics (Proc. SIGGRAPH '87) 21*, 4 (1987), 163–169.

[LWM04] LUM E. B., WILSON B., MA K.-L.: High-Quality Lighting and Efficient Pre-Integration for Volume Rendering. In *Proc. Visualization Symposium '04* (2004), pp. 25–34.

[MAB*97] MARKS J., ANDALMAN B., BEARDSLEY P. A., FREEMAN W. T., GIBSON S., HODGINS J. K., KANG T. H., MIRTICH B., PFISTER H., RUML W., RYALL K., SEIMS J., SHIEBER S. M.: Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation. In *Proc. SIGGRAPH '97* (1997), pp. 389–400.

[Max95] MAX N. L.: Optical Models for Direct Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics 1*, 2 (1995), 99–108.

[MGS02] MEISSNER M., GUTHE S., STRASSER W.: Interactive Lighting Models and Pre-Integration for Volume Rendering on PC Graphics Accelerators. In *Proc. Graphics Interface '02* (2002), pp. 209–218.

[MHC90] MAX N. L., HANRAHAN P., CRAWFIS R.: Area and Volume Coherence for Efficient Visualization of 3D Scalar Functions. *Computer Graphics (San Diego Workshop on Volume Visualization) 24*, 5 (1990), 27–33.

[MLB03] MCGUFFIN M. J., L. T., BALAKRISHNAN R.: Using Deformations for Browsing Volumetric Data. In *Proc. Visualization '03* (2003), IEEE, pp. 401–408.

[PLB*01] PFISTER H., LORENSEN W., BAJAJ C., KINDLMANN G., SCHROEDER W., SOBIERAJSKI AVILA L., MARTIN K., MACHIRAJU R., LEE J.: Visualization Viewpoints: The Transfer Function Bake-Off. *IEEE Computer Graphics and Applications 21*, 3 (2001), 16–22.

[PM04] POTTS S., MOELLER T.: Transfer Functions on a Logarithmic Scale for Volume Rendering. In *Proc. Graphics Interface '04* (2004), pp. 57–63.

[Rez00] REZK-SALAMA, C. AND ENGEL, K. AND BAUER, M. AND GREINER, G. AND ERTL, T.: Interactive Volume Rendering on Standard PC Graphics Hardware Using Multi-Textures and Multi-Stage Rasterization. In *Eurographics Workshop on Graphics Hardware '00* (2000), ACM SIGGRAPH, pp. 109–118.

[RGWE03] ROETTGER S., GUTHE S., WEISKOPF D., ERTL T.: Smart Hardware-Accelerated Volume Rendering. In *Proc. Visualization Symposium '03* (2003), IEEE Computer Society Press, pp. 231–238.

[RKE00] ROETTGER S., KRAUS M., ERTL T.: Hardware-Accelerated Volume and Isosurface Rendering Based on Cell-Projection. In *Proc. Visualization '00* (2000), IEEE, pp. 109–116.

[Roe04] ROETTGER S.: The Volume Library. *http://www9.cs.fau.de/~roettger/library/* (2004).

[SHN03] SHERBONDY A., HOUSTON M., NAPEL S.: Fast Volume Segmentation With Simultaneous Visualization Using Programmable Graphics Hardware. In *Proc. Visualization '03* (2003), IEEE, pp. 171–176.

[TLMM02] TAKANASHI I., LUM E. B., MA K.-L., MURAKI S.: ISpace: Interactive Volume Data Classification Techniques Using Independent Component Analysis. In *Proc. Pacific Graphics '02* (2002), pp. 366–374.

[TM03] TZENG F.-Y. LUM E. B., MA K.-L.: A Novel Interface for Higher-Dimensional Classification of Volume Data. In *Proc. Visualization '03* (2003), IEEE, pp. 505–512.

[VKG03] VIOLA I., KANITSAR A., GROELLER E.: Hardware-Based Nonlinear Filtering and Segmentation using High-Level Shading Languages. In *Proc. Visualization '03* (2003), IEEE, pp. 309–316.

[WEE02] WEISKOPF D., ENGEL K., ERTL T.: Volume Clipping via Per-Fragment Operations in Texture-Based Volume Visualization. In *Proc. Visualization '02* (2002), IEEE, pp. 93–100.

[WM92] WILLIAMS P. L., MAX N. L.: A Volume Density Optical Model. In *Computer Graphics (Workshop on Volume Visualization '92)* (1992), ACM, pp. 61–68.