# Visual Parameters and Transfer Functions

Christof Rezk Salama

## 1 Introduction

The purpose of scientific visualization is to generate meaningful images. Those images should depict certain properties of an underlying abstract data field in a way that is intuitively understandable for the user. As a consequence all visualization techniques require a type of mapping between abstract data values and a corresponding visual representation. One of the most frequently applied operations in this context is the direct assignment of visual attributes to data values. A simple example is the linear mapping of a scalar temperature or density sample to the hue value of a color in HSV space.

From the theoretical point of view, such a mapping is an arbitrary one in general, in the sense that there is no *correct* or *incorrect* mapping, as well as there is no correct or incorrect way a density or temperature value should *look like*. One might want to choose warm and cool colors to represent high and low temperatures, but this is simply a matter of choice for the domain scientist, probably based on existing conventions, and to a great extent on his own personal taste. The usefulness of the mapping, however, is determined by the way it helps the user finding the correct interpretation of the data.

In complex visualization systems, however, finding appropriate settings that yield a desired visual appearance is often a tedious process of manual parameter tweaking. There are two different types of approaches to tackle such problems. Many researchers have developed algorithms to automatize the assignment procedure. Other scientists work on concepts to facilitate the interactive process by making it more intuitive, controllable and goal-directed. Both the automatic approaches and the design of easy-to-use interfaces are still topics of active research. This chapter gives a survey on strategies for assignment of visual properties in scientific visualization.

Christof Rezk Salama
Computer Graphics Group, University of Siegen, Germany, e-mail: rezk@fb12.uni-siegen.de

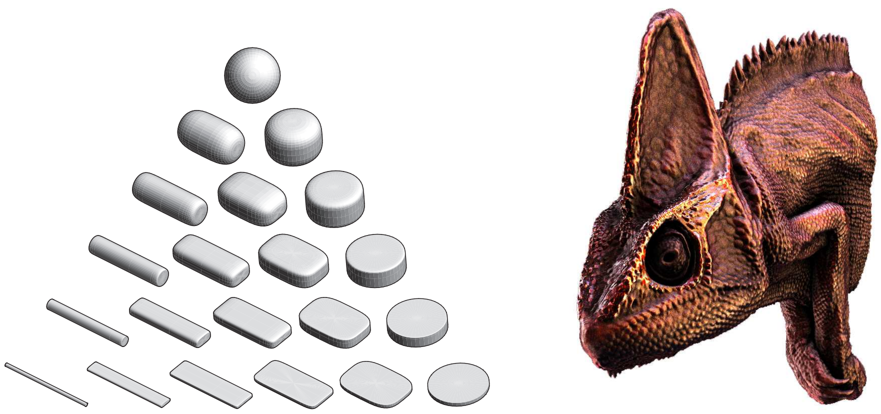

**Fig. 1** Left: Superquadrics can be used to generate a variety of different shapes, such as spheres, discs and rods. Image courtesy of Gordon Kindlmann [13] (©2004 IEEE). Right: Isosurface rendering of the Veiled Chameleon computed tomography from the UTCT data archive [2]. The shininess of the surface encodes the gradient magnitude

## 2 Visual Parameters

We have already mentioned *color* as the most prominent visual attribute. Appropriate strategies for the generation of color maps for different purposes have been investigated by various researchers [4, 30, 42]. Color, however, is not the only visual attribute that is important.

A student who tries to visualize his first vector field will most likely start with a program which draws tiny arrows on the screen. This is a typical example for what is called a *vector glyph*. The length and direction of the arrow shows the magnitude and orientation of the vector field at a given point. We then draw differently colored arrows and use the color to encode another variable from the data set, say, local pressure. We can as well vary the thickness of the arrows to display additional variables.

As we see, *shape* is another important visual property. For more sophisticated vector and tensor glyphs, a variety of shapes can be generated using parametric functions such as the super-quadrics (Figure 1, left) proposed by Kindlmann et al [13].

Almost all rendering techniques in computer graphics separate the shape of an object from its appearance. The shape is specified by a geometric description, usually by a set of polygons. The appearance of an object is defined by textures, material properties and shading algorithms. Figure 1 (right) shows an isosurface extracted from the 3D computed tomography (CT) of the Veiled Chameleon [2]. The shape is defined by carefully selecting an appropriate iso-value. The appearance of the object is used to visualize data properties beyond the isosurface: The magnitude of the gradient vector at every point on the isosurface is encoded as the shininess of the Phong illumination term. Surface areas which have a high gradient due to the

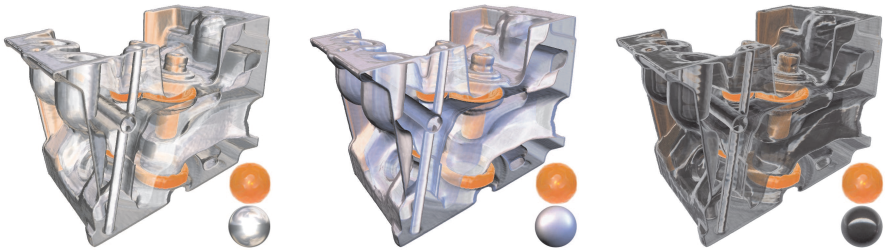

**Fig. 2** Visualization of an engine block with different style transfer functions for illustrative rendering. Images courtesy of Stefan Bruckner [5] (©2007 Eurographics Association)

bone being close to the skin will appear more shiny. Likewise, surface areas with low gradient magnitude will reflect light more diffusely. This is an unusual example of data attributes being mapped to material properties.

Isosurface display is considered an *indirect* volume rendering technique. For *direct* volume rendering, a 3D scalar field is interpreted as a participating medium which simultaneously emits and absorbs light. The scalar variable is mapped to an *emission* and an *absorption* coefficient which is used to calculate light transport in a participating medium. While the emission coefficient is usually specified as an RGB color value representing the wavelength of the emitted light, the absorption coefficient is given as an achromatic opacity value. Bruckner and Gröller expand the transfer function to transfer more complex visual style for illustrative visualization, as illustrated in Figure 2

As we see, there are numerous visual properties, including shape, color, transparency and material properties, which can be used to encode information from the underlying abstract data field.We have the freedom and flexibility to choose whatever visual attribute we believe is appropriate to depict a certain property of the underlying data set.

Nevertheless, in complex visualization systems, changing the mapping of one parameter may have considerably large influence on the final image. There may be other parameters which seem to have no effect on the visual result at all. Remarkably, the visual influence of a single parameter often strongly depends on the settings chosen for other parameters. Imagine a simple glyph, such as the arrow mentioned above, where the size and the color represent different data variables. If the glyph vanishes due to its size approaching zero, the color variable cannot be observed anymore. While in this simple example the relationship between the parameters is easy to recognize, in complex systems the interdependency of visual parameters is usually not that obvious and often hardly predictable. We will address this problem in Section 4.2. For now, let us start by examining how the mapping of visual parameters is achieved.

## 3 Transfer Functions

A function which maps abstract data values to visual parameters is called a transfer function, regardless of whether it is specified explicitly as an analytic function or discretized as a lookup table. One of the most prominent examples of a transfer function in scientific visualization is the assignment of optical properties for direct volume rendering. However, we point out that the general principles are the same for any kind of mapping of abstract data to optical properties. In practice, useful transfer functions are difficult to accomplish due to their high degrees of freedom. This is especially true for multivariate and multidimensional data. Before we have a look at techniques to improve usability, we will examine the mathematical aspects of transfer function design.

Let $\mathscr{D}$ be the range of our data field and $\mathscr{V}$ be a set of visual attributes. A transfer function T is defined as the mapping of a data sample $d \in \mathscr{D}$ to a visual attribute $v \in \mathscr{V}$:

$$\mathrm{T} : \mathscr{D} \mapsto \mathscr{V} \tag{1}$$

In many scientific terminologies, this mapping step is called *classification*, because it categorizes data samples into different *classes*, represented by distinct visual attributes. While the above definition is fairly simple, things get more intricate if we look at the discretization of continuous data fields.

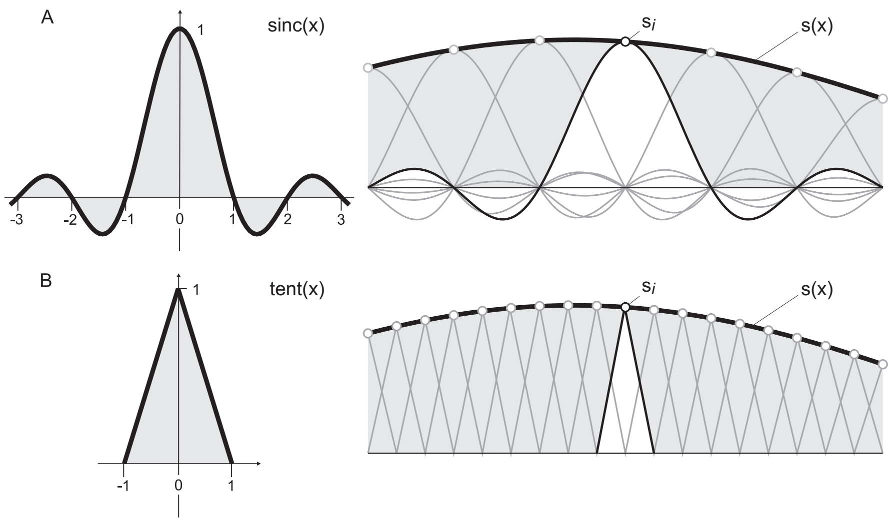


**Fig. 3** Reconstruction filters for discrete 1D signals. The ideal sinc filter (top row) cannot be used in practice due to its infinite extent. The tent filter (bottom row) approximates the signal by piecewise linear segments, which linearly interpolate between adjacent sample positions.

### 3.1 Mapping of Discrete Signals

To understand the sampling process, let us consider a 1D scalar field $s(x) \in \mathbb{R}$ with $x \in \mathbb{R}$. According to the Nyquist theorem, a continuous signal $s$ can be fully reconstructed from a set of discrete sampling points $s_i$,

$$s_i \;=\; s(x_i) \quad \text{with} \quad x_i \;=\; i \cdot \Delta x, \quad i \in \mathbb{N} \tag{2}$$

if the original signal $s$ is band-limited with a limiting frequency $F$ and the sampling step size $\Delta x$ obeys

$$\Delta x < \frac{1}{2F}. \tag{3}$$

In order to process continuous signals in our computer, we must discretize them according to this sampling theorem. In theory, we can then reconstruct the continuous signal $s(x)$ *without loss* by convolving the samples with a sinc filter (see Figure 3, top row):

$$s(x) = \sum_k s_k \cdot \mathrm{sinc}(\frac{x - k\Delta x}{\Delta x}) \qquad \mathit{with} \quad \mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \tag{4}$$

Due to its infinite extent the sinc filter, however, is never used for reconstruction in practice. We usually set the step size $\Delta x$ much smaller than required in Equation 3 and replace the sinc by a tent filter for linear interpolation (Figure 3, bottom row). It is important to note, that interpolation within a data grid means reconstructing the original *continuous* signal from its discrete samples. In most cases, however, this reconstruction is performed *lazily*, which means that signal values are reconstructed only on demand, e.g. values required to calculate the color of a pixel in the final image.

Now, in order to generate visual attributes $C$, the transfer function $T$ must be applied to the scalar signal. For a given position $x$, we reconstruct the scalar signal $s(x)$ and apply the transfer function afterwards:

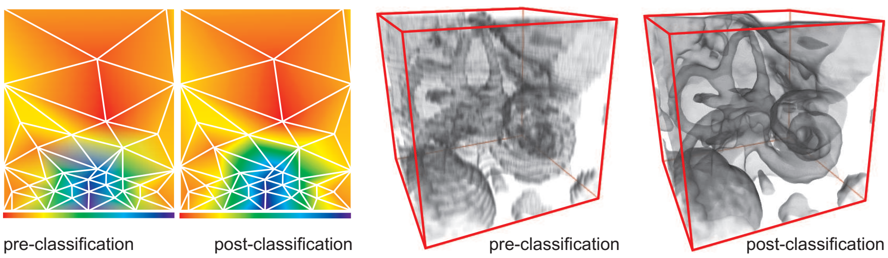


**Fig. 4** Comparison of pre- and post-classification for the 2D heat distribution (left) and for a high-resolution CT of the inner ear (right).

$$\begin{aligned} C(x) &= T(s(x)) \\ &= T\left(\sum_k s_k \cdot \mathrm{sinc}\left(\frac{x-k\Delta x}{\Delta x}\right)\right) \end{aligned} \tag{5}$$

This process is called *post-classification*, because the classification step is performed *after* signal reconstruction. The sinc filter may again be substituted by the tent filter. In practice this means, that for each pixel position $x$ of the image we want to generate, we have to reconstruct the original signal $s(x)$ and use this value as input for the transfer function.

A common pitfall is that it may seem adequate to apply the transfer function directly to the discrete samples $s_i$ and reconstruct the continuous signal from the classified samples $T(s_i)$:

$$\tilde{C}(x) = \sum_k T(s_k) \cdot \mathrm{sinc}\left(\frac{x-k\Delta x}{\Delta x}\right) \tag{6}$$

Obviously, this will result in a different visual attribute $\tilde{C}$ compared to Equation 5. The application of the transfer function *before* signal reconstruction is called *pre-classification*.

The question, whether pre- or post-classification is the correct way to apply a transfer function is easily answered by looking at resulting images. Figure 4 compares images obtained by pre- and postclassification. The left part of the figure shows an unstructured 2D grid with data values mapped to a HSV color scale. With pre-classification the image contains interpolation artifacts . The structure of the underlying grid is clearly visible, which should not be the case if reconstruction is done correctly. The right part shows a transparently rendered volume data set of the inner ear. The pre-classification image shows strong visual artifacts which appear rather disturbing to the observer.

How can these strong visual artifacts for pre-classification be explained? To find the answer we must take into account the frequency spectrum of the transfer function itself. While the original signal $s(x)$ might be band-limited with a Nyquist frequency of $F$, this does not hold true for the modified signal $T(s(x))$. Applying a transfer function changes the frequency spectrum of the signal. As a consequence, the step size $\Delta x$ must be adapted to the new limit frequency. As can be seen in Equation 6, the modified signal is reconstructed with the original step size, and this finally leads to the visual artifacts observed in the result images. The higher the frequencies contained in the transfer function, the stronger these artifacts will become. With post-classification the original signal is reconstructed at the pixel resolution before transfer function assignment. It can be reconstructed even at sub-pixel accuracy if required for anti-aliasing purposes.

As we see, a good piece of advice in general is to perform interpolation only in the data domain, not in the target domain of visual properties. This statement will become obvious, if we consider that both spaces may have different dimensions. For scientific data, the data domain is a spacial domain, with well-known spatial relationships between samples. In contrast, linear interpolation of color values depends,

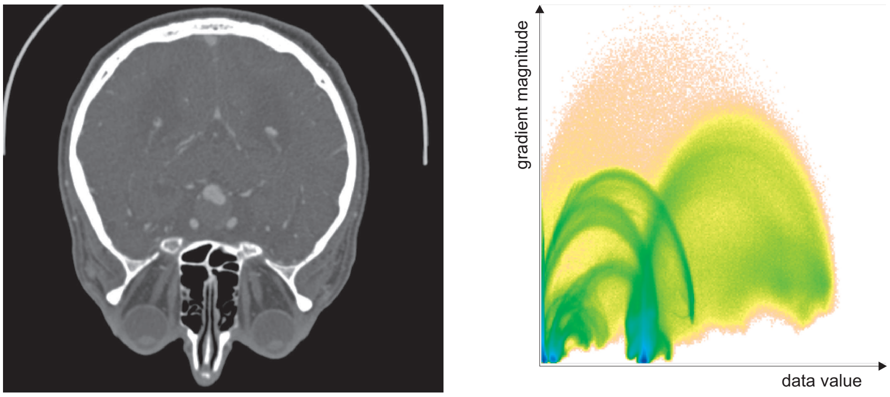


**Fig. 5** Left: Slice image of a CT angiography of the human brain. Right: 2D histogram of data value and gradient magnitude for the same data set.

of course, on the color space. In other cases, such as the tensor glyphs shown in Figure 1, interpolation is hard or even impossible to perform in the target domain.

### *3.2 Multidimensional Transfer Functions*

For multivariate data sets, a variety of different variables are given for each grid point. In weather simulation data sets, for example, those variables comprise velocities, temperature, pressure mixing ratios for cloud water, snow and ice and many others. To account for such data, $n$-dimensional transfer functions may be used, which take an input vector of data values $\mathbf{s} = (s_0, s_1, \dots s_n)^T$ to generate visual attributes:

$$C(x) = T(\mathbf{s}(x)). \tag{7}$$

Kniss et al. [16] show that even for scalar data, multi-dimensional transfer functions can be used if additional variables are computed from the scalar field, such as the gradient magnitude or the magnitude of the second-order derivative. Other variables, such as local curvature, can be used as well. The most frequently used 2D transfer function domain is spanned by the data value and its gradient magnitude, as shown in Figure 3.2. The sharp peeks at the bottom of the histogram are homogenous materials with low gradient magnitude. The transitions between different materials are represented by the parabolic arcs. A respective bi-dimensional transfer function is thus capable of differentiating structures based on homogeneity.

Bi-dimensional transfer functions ($n = 1$) can be realized as 2D lookup tables. For transfer functions of higher dimensionality lookup tables will usually require too much memory. They can, however, be implemented as analytic functions or defined procedurally as a set of multi-dimensional primitives, such as Gaussians [17] or blobs [6]. While multi-dimensional transfer functions are more expensive both in terms of computational load and memory requirements, the benefit of a more accurate and flexible classification outweighs the drawbacks in most cases. For non-expert users, however, even one-dimensional transfer functions often are difficult to set up. In the following sections we are going to review strategies for automatic, or at least semi-automatic generation of transfer functions.

## 4 Usability

The purpose of a visualization system is to support the user in finding answers to very specific questions about the individual data set. A good visualization system should be intuitively useable by the domain scientist. Ideally, it should be easy to understand without detailed knowledge of the system internals, such as the rendering and mapping techniques. Transfer functions are integral parts at the core of each visualization system. Without sophisticated tools and techniques, however, transfer function design, is a tedious and time-consuming task. The reason for this is twofold:

The general representation of a transfer function has a lot of degrees of freedom. Managing the vast *complexity* of the transfer function, makes it difficult to handle in practice. Many design techniques thus aim at reducing the complexity of the transfer function by eliminating redundant low-level parameters. The first step in most approaches is some type of dimensionality reduction technique.

The second and more important problem with transfer function design is the lack of an appropriate mental model. According to the seminal work by Donald Norman [24], the process of human computer interaction can be coarsely divided into the user's *intension* to do something, the resulting *action* and the control over the *execution* of this action. The critical point for the user is to find an appropriate *action* to achieve his original *intension*. As an example, a physician who utilizes a volume rendering system to display his computed tomography data, would like to render the soft tissue transparently in order to examine the underlying structures. In most volume rendering systems, however, finding the appropriate action which leads to the desired result is a trial-and-error process. According to Norman's terminology the lack of an appropriate mental model of the user interfaces leads to a *gulf of execution*.

### *4.1 The Automation Problem*

Many researchers have developed techniques to build an intelligent system, which determines an appropriate transfer function automatically, or at least comes up with an initial suggestion, which can be further edited and modified by the user. The usefulness of a transfer function, however, strongly depends on the underlying question the user wants to answer. Hence, the minority of automation techniques work fully autonomously and very few techniques automatically consider domain knowledge in the design process. Existing automatic and semi-automatic approaches can be divided into image-driven and data-driven techniques.

### 4.1.1 Image-Driven Techniques

The purpose of a transfer function is to create meaningful images. It thus seems obvious that we have to analyze images generated with different parameter settings in order to derive an optimal set of parameters. Such image-driven techniques mainly differ in the metric they use to evaluate the quality and significance of an image. State-of-the-art methods can be divided into interactive evolution algorithms and inverse design techniques.

*Interactive evolution* approaches are semi-automatic techniques which provide guidance for the user while he is exploring the parameter space interactively. Successful techniques are closely related to artificial intelligence systems. The most straight-forward technique is thumbnail selection [19]. The system proposes a set of rendered thumbnails images generated with a selection of different parameter settings. The user selects one or more images that he finds most appropriate. Based on the user's selection, the system proposes a new set of thumbnails generated by a genetic algorithm. In fact, many techniques originally developed as artificial evolutionary art systems for design automation [18, 36, 38]) can be adopted to the problem of transfer function design. One of the most general concepts for visual parameters in computer graphics and animation is the *Design Gallery* [22] which tries to build a distinctive set of visually distinguishable images.

*Inverse Design* techniques search for optimal parameter settings according to an objective quality measure. He et al. [9] have developed a technique for semi-automatic transfer function generation using stochastic search techniques. The search algorithm is controlled by evaluating an objective metric such as entropy,

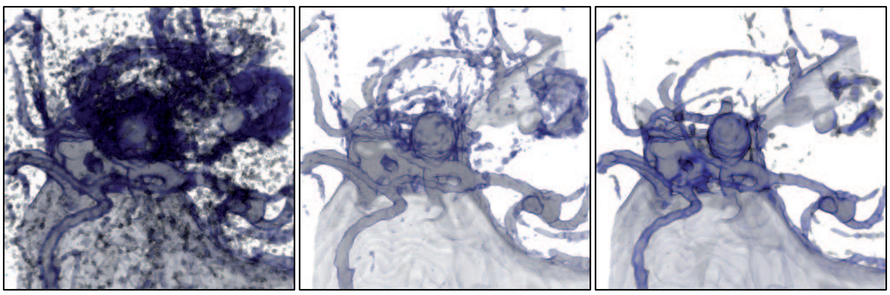

**Fig. 6** Results of non-linear adaptation of predefined transfer function templates. Left: No adjustment. Middle: Adjustment based on histogram matching. Right: Adjustment based on matching the position functions.

edge energy or histogram variance. Possible strategies for searching the huge parameter space most efficiently comprise genetic algorithms [11, 8], hill-climbing and simulated annealing [15]. Inverse design strategies can as well be based on concepts from artificial intelligence including sensor-actuator networks [25] and goal-based rendering [12].

While image-based techniques work quite well in practice, a general drawback is their unavoidable dependency on image-related parameters such as viewing position and pixel resolution. As a possible solution to the problem of view-dependence, image-driven techniques can be supplemented by approaches for automatic viewpoint selection. Many methods developed for image-based rendering [23, 10, 1, 32, 37] can be adopted for this purpose.

### 4.1.2 Data-Driven Techniques

Data-driven techniques analyze the volume data itself instead of the generated images. The process of transfer function design is thus decoupled from the influence of the image-related parameters mentioned in the previous section. Early approaches, such as the ones proposed by Fang et al. [7] and Sato et al. [34], derive optical properties by applying 3D image processing operations directly to the data. Bajaj et al. [3] propose a data-driven technique which generates a transfer function by evaluating statistical information about surface area and gradient magnitude of all the isosurfaces contained in the data.

The most prominent data-driven technique is a semi-automatic approach presented by Kindlmann and Durkin [14] in 1998. Their semi-automatic approach is capable of determining material boundaries within a given data set by evaluating statistical information about the first and second order directional derivatives. A material boundary is assumed if a maximum of the first-order derivative coincides with a zero-crossing of the second order derivative. The authors derive a *position function* $p(s)$, which describes the average distance of a data sample with scalar value $s$ from an assumed boundary in the data set. Kindlmann and Durkin's approach turns out to be successful in determining material boundaries in unknown data sets.

Up to this point, all automatic or semi-automatic approaches to transfer function design tried to build a transfer function from scratch for every new data set. What was missing up until now are strategies for reusing existing transfer functions for similar data sets. The position function approach was utilized by Rezk-Salama et al. [28] to reliably adapt a pre-defined reference transfer function to individual data by non-linear distortion of the parameter axis. As displayed in Figure 6, this work demonstrates that matching the position function of different data sets leads to more reliable results than matching the histograms only. Later, this adaptation approach was expanded to 2D transfer functions by Vega et al. [40].

Most existing applications provide a graphical user interface for the assignment of visual parameters. The histogram is one of the main means of orientation provided by the user interface for transfer function design. Among the very few approaches which incorporate domain knowledge into the classification process, is the work of Lundström et al. [20]. They introduce local histograms, which utilize a priori knowledge about spatial relationships to automatically differentiate between different tissue types. The classification is performed by a bi-dimensional transfer function, where the certainty of a voxel belonging to a specified tissue type is used as second parameter. The same group also introduced the $\alpha$- histogram [21]. The basic idea is to split the data domain into disjoint subsets, then calculate separate histograms for these subsets. The values of the local histograms are raised to the power of $\alpha > 1$. The global histogram is finally calculated by summing the contribution from the modified local histograms. The authors show that the applied local exponentiation results in an amplification of homogenous regions of the data in the histogram.

## 4.2 The Interface Problem

The most actively followed path in transfer function design in recent years is the design of intelligent user interfaces. On the one hand, automatic approaches are often not flexible enough. They are difficult to adapt to a wide range of data sets. On the other hand, they are often not specific enough to account for the precise task the user wants to perform. In medical environments, fully automatic approaches are often not widely accepted due to the considerable amount of uncertainty. If automatic approaches fail to deliver satisfying results, non-expert users are often left alone. In this section we review general concepts developed in recent years to create goal-directed user interfaces.

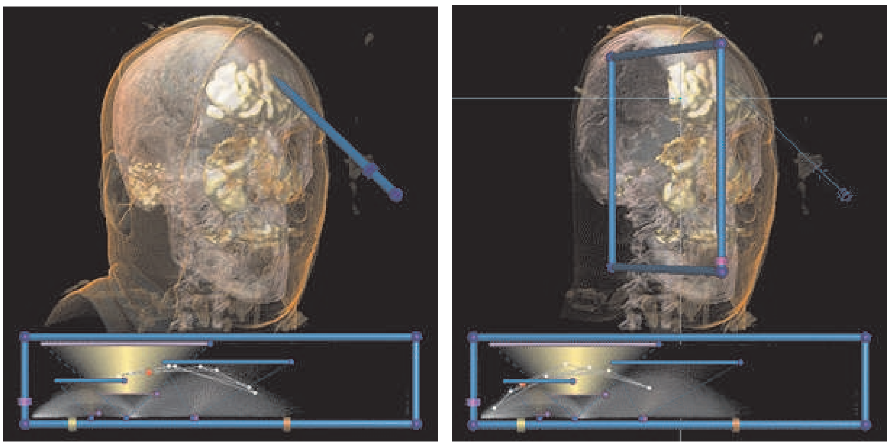

**Fig. 7** Dual domain interaction: The user can probe the scalar field with a virtual pen in 3D. The respective data values are marked in the small window at the bottom which represents the transfer function domain. Right: Clipping planes may be used to facilitate the probing. Image courtesy of Joe M. Kniss [16]

### 4.2.1 Geometric Primitives

In order to reduce the effort of editing single entries in a color table, many user interfaces compose the color mapping from multiple independent geometric primitives. In practice, such component may comprise polylines, ramps and trapezoids [16], paraboloids [41] or other primitive shapes. Compared to simple color tables, where every entry can be edited separately, composition of primitives provide effective solutions for reducing the complexity of transfer functions. Most visualization algorithms, however, employ a given color table directly. Hence, primitive shapes must be converted to a tabular representation before usage. Few publications give details about the compositing of overlapping primitives. For opacity, the maximum is usually chosen and the color values are mixed according to their opacity ratio [29].

In practice, single functions are better suited for 1D classification of continuous simulation data from natural science and engineering. Such data often do not contain sharp structures which need to be separated from each other. For tomographic scans, a composition based on geometric primitives is advantageous, allowing the individual treatment of different spatial structure. Anatomical structures are often represented by geometric primitives in the transfer function domain. For multi-dimensional transfer function domains, single functions are rarely used, regardless of the scientific area, due to difficulties in providing appropriate user interfaces for editing. Roettger et al.[33] have developed a concept called spatialized transfer functions which generates primitives automatically by segmenting the 2D transfer function domain into disjoint regions. The user may then specify optical properties selectively for those regions.

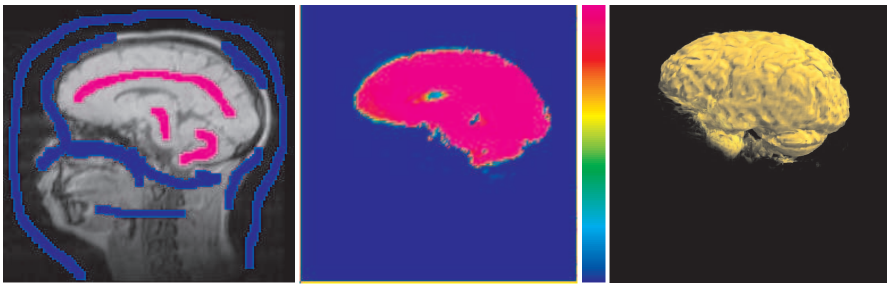

**Fig. 8** Neural network approach to multi-dimensional transfer function design. The user marks regions of interest by roughly setting the boundaries. The system automatically derives a multi-dimensional classification using a neural network. Image courtesy of Fan-Yin Tzeng [39] (©2003 IEEE)

### 4.2.2 Domains of Interaction

Kniss et al. have introduced the concept of dual domain interaction [16] for volume visualization. During their investigation on multi-dimensional transfer functions, the authors notice that finding appropriate settings is time-consuming if the user is not provided with some guidance. To this end they developed a dual user interface, which allows the user to probe the scalar field with a virtual pen in 3D (see Figure 7). The user can then decide to place a geometric primitive in the transfer function window to mark the data in 3D. With this approach it is easy for the user to point at interesting structures in the image domain, and assign visual attributes in the parameter domain. The effect of changing visual parameters are immediately visible in the 3D image.

The concept of interacting in different domains is not completely new. In the field of information visualization such an interaction is called *brushing*. Brushing represents an intuitive way for the user to visually select subsets of the data samples without the necessity to specify numerical ranges. Likewise, Pradhan et al.[26] demonstrate a visualization system which utilizes concepts borrowed from information visualization. They use parallel coordinates (see image 9, left) to select regions in a multi-dimensional transfer function domain called the signature space including intensity, gradient magnitude and higher-order statistical moments.

Inspired by the work of Kindlmann and Durkin [14], Sereda et al.[35] introduce an new domain for interaction called the LH-histogram, or low-high-histogram (Figure 9, left). The LH-histogram can be viewed as a 2D map of the different tissue spatial structues contained in the volume data set. Each spatial data sample with a gradient magnitude larger than a given threshold is considered a boundary sample. For each boundary sample a short pathline is traced along the gradient field in order to determine the low and the high intensity values used as 2D index into the map. While such an approach is highly efficient for expert user, it is not really intuitive to use for non-experts.

Tzeng et al. [39] suggest an intelligent visualization system with interaction in the spacial domain only, while hiding the transfer function domain completely. In their system the user marks regions of interest by roughly painting the boundaries on a few slice images. While he is painting, the marked regions are used to train an artificial neural network for multi-dimensional classification. Figure 8 shows an example slice image with the painted markings, the classified slice image and the final rendered 3D image from an MRI data set. Del Rio et al. have adapt this approach to specify transfer functions in an augmented reality environment for medical applications [31].

In practice, the applicability of such intelligent systems, which calculate the mapping during interaction strongly depends on the visual feedback provided to the user, the speed at which the effect of a user action is displayed in the image. This again depends on the computational complexity of specific visualization system, the size of the data set and the available memory bandwidth.

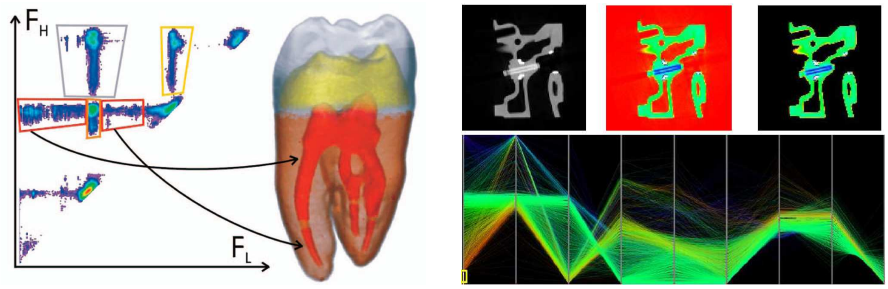


**Fig. 9** Left: Example of an LH-histogram for the tooth data set. Image courtesy of Sereda et al.[35]. (©2006 IEEE) Right: Parallel coordinates are used to select voxels with a given signature in a multi-dimensional parameter space. Image courtesy of Pradhan et al [26]

### 4.2.3 Semantic Models

If we consider domain knowledge from the field of human computer interaction, we will quickly notice that many graphical user interfaces are hardly goal-directed. For the non-expert user it is hard to figure out which modification to the complex parameters are necessary to cause the action he needs to perform. While the approaches described in the previous section still provide a data-centered view onto the visualization problem, there are novel approaches which try to improve the mental model of the application by providing clear semantics for the possible set of actions.

Rautek et al.[27] present a semantic framework for volume illustration and illustrative visualization in general. The mapping between abstract data values and visual properties is specified by rules based on fuzzy logic operations. The authors concept of semantic layers replaces the traditional transfer function setup and allows the user to specify the mapping from attributes to visual styles in natural language. Their system supports semantics like, *"if density is high and curvature is close to zero, then color coding is red"*. While fuzzy logic and the use of natural language represents a significant improvement, this approach does not immediately resolve the gulf of execution. If the non-expert user does not know, how to describe the structures of interest in terms of density or curvature values, the semantics will not help.

To this end Rezk-Salama et al. [29] suggest a general concept for tailoring the user interface to the visualization task. They target very specific visualization scenarios, such as routine examinations in medical practice. In their approach a visualization expert works together with a domain scientist to create a semantic model for the visualization task. With such a model, the user may directly select structures by name, such as *brain tissue* or *blood vessels* and directly adjust their visual parameters, such as color, contrast, or visibility. In order to create such a semantic model, a visualization task is performed several times on a collection of data sets which are considered representative for a specific examination scenario. Throughout this training phase the system collects the parameter vectors and analyzes them using principal component analysis.

Figure 10 shows a classified CT angiography data set with the separate anatomical structures brain, soft tissue and blood vessels. For this system, the authors have conducted a small user study. Physicians were presented with a semantic user interface without labels for the interaction elements (sliders and buttons). In most cases the physicians were capable of determining the semantics of actions triggered by the elements.

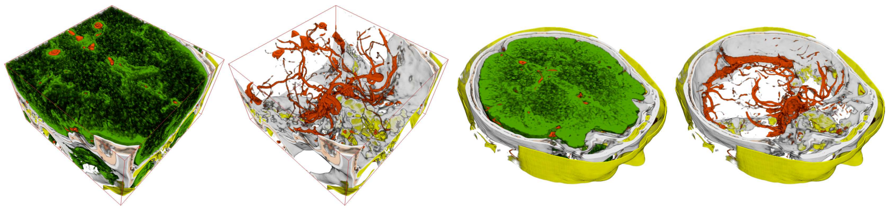

**Fig. 10** Examples of a semantic model for CT angiography, with the anatomical structures, brain, soft tissue bone and vasculature.

## 5 Conclusion

Transfer functions are at the heart of every visualization system. Transfer function design techniques are still an area of active research and will probably be for a very long time. The direction of research, however, has slightly changed in recent years. Transfer function design clearly shifts towards more user-centered applications, such as intelligent exploratory interfaces. Systems which hide most of the details of the underlying rendering algorithms are more intuitive to use by non-experts. Up until now only very few publications exist about including semantics into visualization systems. Especially for clinical systems, transfer function design must become faster and easier to achieve to become widely accepted. Semantic approaches have a high potential to achieve these goals.

We have only just begun to incorporate knowledge from other fields into our visualization systems to make them more flexible, more intuitive and more usable. In the near future we may expect important impulses from related fields such as computer animation and human computer interaction as well as influences from areas such as cognitive sciences.

Fully automatic transfer function design techniques are not widely accepted due to uncertainty in the data sets, which still needs to be resolved by the user. Scientific data sets contain much more information than can be displayed in a static image. The user must be provided with means of exploration, and this must include the transfer function as well. Fully automatic techniques are only applicable if structures contained in the underlying data are well-understood, which might be the case for simulation data, but usually not for measured data. For medical scans, and especially for pathologic cases, fully automatic techniques often fail. Nevertheless, they may be useful for generating good starting points for exploratory analysis.

Users of visualization systems increasingly ask for systems that are more usable and more efficient in practice. Companies which produce commercial systems have noticed that usability aspects are growing more important as a vital factor to secure market shares. In consequence, implementing general concepts to improve the usability and the acceptance of visualization systems is a research area of steadily growing importance.

At research facilities, multidimensional transfer functions have already become standard today. In commercial systems they are not yet widely used. The main reason for this might be that many commercial products are based on special-purpose hardware, which up until now only supports 1D lookup tables. The benefit of at least bi-dimensional classification, however, is incontrovertible, so we expect this will change in the near future.

## 6 Summary

There is a huge variety of different visual attributes that can be used to represent abstract data, including shape, color, transparency, and more complex optical properties such as emission and absorption coefficients, or specular reflectivity. Care must be taken to correctly reconstruct the original signal from discrete samples. We have seen that the sequence of reconstruction and classification operations makes a significant difference to the image quality and we have explained these differences with respect to sampling theory.

Transfer functions can be stored as lookup tables or applied procedurally as an explicit function. Multi-dimensional transfer functions are powerful and flexible, yet quite expensive with respect to computational cost and memory consumption. This chapter has given an overview of state-of-the-art techniques for automatic and semi-automatic techniques for parameter assignment. Image-driven techniques are based on interactive evolution or inverse design approaches. Data-driven algorithms analyze the data to generate meaningful parameter assignments. State-of-the art user interfaces for transfer function design are based on important strategies such as dual domain interaction and semantic models.

## References

1. Arbel, T., Ferrie, F.: Viewpoint Selection by Navigation Through Entropy Maps. In: Proc. IEEE Int. Conf on Computer Vision (ICCV) (1999)

2. of Texas at Austin, U.: Utct data archive, digimorph and ctlab. http://utct.tacc.utexas.edu/. Last visited July 2007
3. Bajaj, C., Pascucci, V., Schikore, D.: The Contour Spectrum. In: Proc. IEEE Visualization (1997)
4. Bergmann, L., Rogowitz, B., Treinish, L.: A Rule-Based Tool for Assisting Colormap Selection. In: Proc. IEEE Visualization (1995)
5. Bruckner, S., Gröller, M.E.: Style transfer functions for illustrative volume rendering. Computer Graphics Forum (accepted for publication) **26**(3) (2007)
6. Engel, K., Hadwiger, M., Kniss, J., Rezk-Salama, C., Weiskopf, D.: Real-Time Volume Graphics. AK Peters, Ltd. (2006)
7. Fang, S., Biddlecome, T., Tuceryan, M.: Image-Based Transfer Function Design for Data Exploration in Volume Visualization. In: Proc. IEEE Visualization (1998)
8. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley (1989)
9. He, T., Hong, L., Kaufman, A., Pfister, H.: Generation of Transfer Functions with Stochastic Search Techniques. In: Proc. IEEE Visualization (1996)
10. Hlavac, V., Leonardis, A., Werner, T.: Automatic Selection of Reference Views for Image-Based Scene Reprentations. In: Proc. European Conference on Computer Vision (ECCV) (1996)
11. Holland, J.: Adaption in Natural and Artifical Systems. University of Michigan Press (1995)
12. Kawai, J., Painter, J., Cohen, M.: Rapidoptimization – Goal-Based Rendering. In: Proc. SIGGRAPH (1993)
13. Kindlmann, G.: Superquadric tensor glyphs. In: Proc. IEEE TVCG/EG Symp. on Visualization, pp. 147–154 (2004)
14. Kindlmann, G., Durkin, J.: Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering. In: IEEE Symposium on Volume Visualization (1998)
15. Kirkpatrick, S., Gerlatt, C., Vecchi, M.: Optimization by Simulated Annealing. Science **220** (1993)
16. Kniss, J., Kindlmann, G., Hansen, C.: Interactive Volume Rendering using Multi-dimensional Transfer Functions and Direct Manipulation Widgets. In: Proceedings of IEEE Visualization, pp. 255–262 (2001)
17. Kniss, J., Premoze, S., Ikits, M., Lefohn, A., Hansen, C., Praun, E.: Gaussian Transfer Functions for Multi-Field Volume Visualization. In: Proc. IEEE Visualization (2003)
18. Kochhar, S.: A Prototype System for Design Automation via the Browsing Paradigm. In: Proc. Graphics Interface (1990)
19. König, A., Gröller, E.: Mastering Transfer Function Specification by Using VolumePro Technology. In: Proc. Spring Conference on Computer Graphics (2001)
20. Lundström, C., Ljung, P., Ynnerman, A.: Extending and simplifying Transfer Function design in medical Volume Rendering using local histograms. In: Proc. IEEE/EuroGraphics Symposium on Visualization, p. 263270 (2005)
21. Lundström, C., Ynnerman, A., Ljung, P., Persson, A., Knutsson, H.: The $\alpha$-histogram: Using Spatial Coherence to Enhance Histograms and Transfer Function Design. In: Proc. Eurographics/ IEEE-VGTC Symposium on Visualization (2006)
22. Marks, J., Andalman, B., Beardsley, P., Pfister, H.: Design Galleries: A General Approach for Setting Parameters for Computer Graphics and Animation. In: Proc. SIGGRAPH (1997)
23. Muehler, K., Neugebauer, M., Tietjen, C., Preim, B.: Viewport selection for intervention planning. In: Proc. IEEE/Eurographics Symposium on Visualization (EuroVis), pp. 267–274 (2007)
24. Norman, D.: The Psychology of Everyday Things. Basic Books (2002)
25. van de Panne, M., Fiume, E.: Sensor-Actuator Networks. In: Proc. SIGGRAPH (1993)
26. Pradhan, K., Bartz, D., Mueller, K.: SignatureSpace: A Multidimensional, Exploratory Approach for the Analysis of Volume Data. Tech. rep., Department of Computer Science (WSI), University of Tuebingen, (2005)
27. Rautek, P., Bruckner, S., Grller, M.E.: Semantic layers for illustrative volume rendering. In: Proc. IEEE Visualization (2007)

28. Rezk-Salama, C., Hastreiter, P., Scherer, J., G.Greiner: Automatic Adjustment of Transfer Functions for 3D Volume Visualization. In: Proc. Vision, Modeling and Visualization (VMV) (2000)
29. Rezk-Salama, C., Keller, M., Kohlmann, P.: High-Level User Interfaces for Transfer Function Design with Semantics. In: Proceedings of IEEE Visualization (2006)
30. Rheingans, P.: Task-Based Color Scale Design. In: Proc. SPIE Applied Image and Pattern Recognition (1999)
31. del Rio, A., Fischer, J., Köbele, M., Bartz, D., Straŝer, W.: Augmented Reality Interaction for Semiautomatic Volume Classification. In: Proc. of Eurographics Workshop on Virtual Environments (EGVE) (2005)
32. Roberts, D., Marshall, A.: Viewpoint Selection for Complete Surface Coverage of Three-Dimensional Objects. In: Proc. British Machine Vision Conference (1998)
33. Roettger, S., Bauer, M., Stamminger, M.: Spatialized transfer functions. In: Proc. IEEE/EuroGraphics Symp. on Visualization, pp. 271–278 (2005)
34. Sato, Y., Westin, C.F., Bhalerao, A.: Tissue Classification Based On 3D Local Intensity Structures for Volume Rendering. IEEE Transactions on Visualization and Computer Graphics **6** (2000)
35. Sereda, P., Vilanova Bartolí, A., Serlie, I., Gerritsen, F.A.: Visualization of Boundaries in Volumetric Data Sets Using LH Histograms. Trans. on Vis. and Comp. Graph. **12**(2), 208–218 (2006)
36. Sims, K.: Artificial Evolution in Computer Graphics. In: Proc. SIGGRAPH (1991)
37. Takeushi, V., Ohnishi, N.: Active Vision Systems Based on Information Theory. Systems and Computers in Japan **29**(11) (1998)
38. Todd, S., Latham, W.: Evolutionary Art and Computer Graphics. Academic Press (1992)
39. Tzeng, F.Y., Lum, E., Ma, K.L.: A Novel Interface for Higher-Dimensional Classification of Volume Data. In: Proceedings of IEEE Visualization, pp. 505–512 (2003)
40. Vega Higuera, F., Sauber, N., Tomandl, B., Nimsky, C., Greiner, G., Hastreiter, P.: Automatic adjustment of bidimensional transfer functions for direct volume visualization of intracranial aneurysms. In: Medical Imaging 2004, vol. 5367, pp. 275–284. SPIE (2004)
41. Vega Higuera, F., Sauber, N., Tomandl, B., Nimsky, C., Greiner, G., Hastreiter, P.: Automatic Adjustment of Bidimensional Transfer Functions for Direct Volume Visualization of Intracranial Aneurysms. In: Proceedings of SPIE Medical Imaging (2004)
42. Ware, C.: Color Sequences for Univariate Maps: Theory, Experiments, and Principles. IEEE Computer Graphics and Applications **8** (1998)