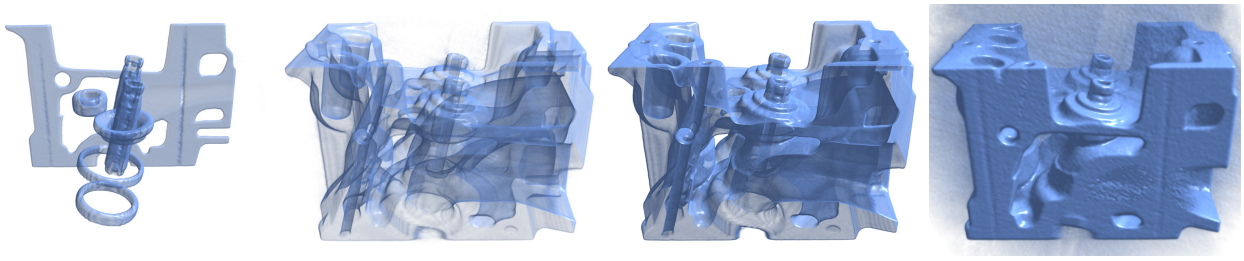
Visibility-Difference Entropy for Automatic Transfer Function Generation

Philipp Schlegel and Renato Pajarola

Visualization and MultiMedia Lab, Department of Informatics, University of Zürich



(a) (b) (c) (d)

Figure 1: Visibility-difference entropy (VDE) transfer function generation applied to the engine volume dataset. VDE transfer function generation creates a set of distinct transfer functions, which reveal the different structures of the dataset. Four transfer functions from this set have been picked showing the valve guides (a), the internal structure (b), the exhaust duct (c), and the opaque surface (d).

ABSTRACT

Direct volume rendering allows for interactive exploration of volumetric data and has become an important tool in many visualization domains. But the insight and information that can be obtained are dependent on the transfer function defining the transparency of voxels. Constructing good transfer functions is one of the most time consuming and cumbersome tasks in volume visualization. We present a novel general purpose method for automatically generating an initial set of best transfer function candidates. The generated transfer functions reveal the major structural features within the volume and allow for an efficient initial visual analysis, serving as a basis for further interactive exploration in particular of originally unknown data. The basic idea is to introduce a metric as a measure of the goodness of a transfer function which indicates the information that can be gained from rendered images by interactive visualization. In contrast to prior methods, our approach does not require a user feedback-loop, operates exclusively in image space and takes the characteristics of interactive data exploration into account. We show how our new transfer function generation method can uncover the major structures of an unknown dataset within only a few minutes.

Keywords: Volume rendering, automatic transfer function generation, image-space methods, entropy.

1. INTRODUCTION

An important part of volume visualization is the transfer function, specifying the mapping of scalar field to color and opacity values and thus the visual appearance of an image. Since the manual design of good transfer functions can be a very time consuming, often unintuitive and difficult task, better (semi-)automatic solutions are desired. The majority of prior solutions are based on the analysis of the volumetric data in *object space*, identifying and exploiting specific features in the data. Often targeted at specific domains, the underlying algorithms primarily generate transfer functions revealing the data properties that they have specifically been designed for. This is reasonable as long as the sought features are known in advance. In contrast, there are far less general purpose approaches allowing for insight into an originally unknown dataset.

In this paper we propose a new general purpose approach, which unlike the majority of existing methods, is computed entirely in *image space* and generates not one single but a set of best transfer function candidates. We want to explore the

Further author information: (Send correspondence to Philipp Schlegel)
Philipp Schlegel: E-mail: schlegel@ifi.uzh.ch
Renato Pajarola: E-mail: pajarola@acm.org

potential of defining good transfer functions automatically and purely from rendered volume images. The benefits of this approach are its independence of domain specific feature analysis and description methods that have to be known or defined for object-space transfer function generation. Our purely image-space centric method presents the user with a choice of multiple good transfer function candidates that reveal the different characteristics of the volume dataset. Furthermore, our approach is orthogonal to object-space volume data analysis techniques and thus could be a complementary component.

The qualifying idea of our method is to introduce a metric derived from information theory taking specifically the interactive visual exploration of the data into account. The goal is to measure the *goodness* of a transfer function in a simple way based on the information that can be gained from images by interactive visualization. Therefore, the information of a transfer function is not measured from a static image but considers the differences in visual appearance, i.e. visibility, when animating the data. Then finding a set of best transfer functions is a non-linear optimization problem based on that metric. To solve this optimization problem we define a set of simple parametrized basis transfer functions. In the resulting parameter space, the optimization is carried out by an adaptive combined simulated annealing and gradient ascent search.

For the remainder of the paper we will abbreviate visibility-difference entropy with VDE and call our method *VDE transfer function generation* and the metric *VDE metric*. The main contributions presented in this paper include:

- A general purpose method for automatic generation of a set of best transfer functions based on the analysis of rendered images.
- The definition of a *visibility-difference entropy* metric taking interactive visual exploration into account.
- An adaptive combined simulated annealing/gradient ascent search method for finding the best distinct transfer functions including adaptive seed handling.
- A ranking and grouping method for transfer functions based on their similarity.

2. PREVIOUS WORK

2.1 Object-Space Methods

Object-space transfer generation methods analyze and use the properties and features of the volume data itself. Kindlmann et al.[1] use the scalar field, first- and second-order derivative values and capture their relationship and thus the boundary information in a histogram volume. Bajaj et al.[2] propose the contour spectrum consisting of scalar data and contour attributes computed over a range. This includes surface area, volume, and gradient integral etc., which are presented to the user as signature graphs to help select the right parameters. In a recent method[3] the opacities are distributed over the branches of the contour tree by a residue flow model based on Darcy's Law. Further, it is shown that the topological attributes can be used to generate harmonic color transfer functions as well. Hyper Reeb graphs[4,5] are able to identify and highlight critical iso-surfaces and keep the change in hue as well as the opacity constant except for the critical iso-surfaces.

Other ideas include the generation of multi-dimensional transfer functions by presenting slices of the dataset with the option to paint regions of interest.[6] A neural network is then used to generate the multi-dimensional transfer function. General regression neural networks are also exploited for transfer function design[7] as well as for generating segmentations.[8,9] Focused on tissue classification,[10] 3D filters based on the gradient vector and Hessian matrix of the volume data can be employed to generate multi-dimensional transfer functions. Another method for tissue detection is based on partial range histograms,[11] by detecting a peak pattern in ranges of local intensity histograms. The visualization of boundaries is the goal of an approach based on low-high histograms.[12] A very recent method incorporates Gaussian mixture models.[13]

Our method differs by focusing exclusively on the generated images and not involving complex and costly data analysis (e.g. for very large datasets). However, the main reason for using an image-space method is the possibility to take the effect of different viewpoints in interactive exploration into account. This is hardly possible with pure object-space methods.

2.2 Image-Space Methods

In contrast, image-space methods do not analyze the properties and features of the volumetric dataset, but rely on the generated images and often adjust the transfer function parameters in a feedback loop. This includes semi-automatic methods where the adjustment of the parameters is directed by the user after inspecting the rendered images. Early work includes a model that defines a transfer function as a sequence of 3D image processing procedures.[14] The design of transfer functions can also be treated as a parameter optimization problem.[15] Based on stochastic algorithms an initial set

of transfer functions is generated, which is then evaluated by the user. The search is repeated using a genetic algorithm until a satisfying result is obtained. The fitness can be influenced interactively in a feedback loop by the user or assigned automatically using entropy, variance, or edge energy. Similarly a particle swarm can be used to solve the optimization problem.[16] Design galleries[17] are another method if neither interactive evolution is suitable nor the output quality can be quantified, thus making automatic optimization impossible. Our method does not apply a feedback loop but presents the user with a set of best transfer function candidates, thus could be viewed as an enhanced design gallery approach.

A more recent approach introduces visibility-driven transfer functions[18] where histograms representing the visibility of a sample from a given viewpoint are defined. This is not a pure image-space method since the visibility-histograms are computed on the samples in the volumetric dataset. Moreover, the user has to provide an initial transfer function requiring previous knowledge, which is then optimized in terms of the visibility and the error to the initial transfer function. Incorporating informational divergence[19] is a way to fit the visibility distribution to a user defined target distribution. Again the user must choose a target distribution beforehand, requiring previous knowledge. A single transfer function matching the chosen target distribution is then generated. In our approach no previous knowledge is required and the result is not a single transfer function but a set of distinct transfer functions. Other works focus on good user interfaces and widgets.[20–24]

Our VDE transfer function generation is performed in image space and employs a fully automatic optimization for searching the best transfer functions. Unlike most existing methods, no user interaction is required and an entire set of best transfer function candidates is computed instead of just a single one. To our best knowledge this is unique to our solution as there is no other method that generates a set of distinct good transfer functions in any similar way.

2.3 Visual Quality and Saliency

The evaluation of transfer functions based on rendered images is important for semi-automatic solutions using a feedback loop but the literature is rather sparse when it comes to defining a metric. However, the problem of saliency or finding salient viewpoints is closely related. One way is based on obscurance,[25] representing the occlusion information associated with voxels. The best viewpoint is computed from the variation of obscurance of visible voxels. An alternative is the viewpoint entropy,[26] which is based on the projected area of faces relative to their total area. Replacing the viewpoint entropy by a linear combination of viewpoint entropy, luminance, and chrominance is another variation.[27] A further evolution leads to the viewpoint mutual information, which is a channel between a set of viewpoints and the viewpoint entropy.[28] A different way is to take the entropy of the intensity image of the visible boundary structures relative to the viewpoint.[29] Basically, the boundary structures describe the shape of the objects from the volumetric dataset and the variance between the shapes and the original objects describes the details. Semantic driven approaches for viewpoint selection rely on view-dependent shape properties[30] where the viewpoint is selected such that the visibility of meaningful features are maximized. A different idea is the decomposition into feature components of a volume for viewpoint selection.[31] Another option is to choose the viewpoint such that the objects based on shape properties can optimally be discriminated from objects in a database.[32] On the other hand, the saliency map[33] is focused on scene analysis. The saliency map can be extended to a quality metric[34] where the user defines a relevancy map.

A major difference between our VDE metric and metrics for saliency and salient viewpoints is that the latter focus on the quality of images from particular viewpoints whereas our metric focuses on the *changes* between images from different viewpoints. This is important as our goal is to find good transfer functions for interactive viewing and not for static single shots. Furthermore, for many saliency metrics psychological models of stimuli and attention as well as esthetics play an important part, and some also require data relevancy as a prerequisite. All this is not required in our metric.

3. VISUAL INFORMATION-BASED METRIC

The goal of our VDE transfer function generation is to provide a set of transfer functions that are best for interactive applications. They should reveal as much information as possible of the volume data when rotating, panning and zooming during interactive exploration, and provide excellent revelation of structural features during animation.

The basic idea is to use Shannon's entropy[35] to evaluate the information content as already suggested in previous work (see Section 2). But instead of computing the entropy of resulting images, the entropy of *differential images* from different viewpoints is computed, see Figure 2. This means that a transfer function revealing much of the features or the complex boundary of a volume dataset when animated, thus providing good structural perception, will be rated high. In contrast, a transfer function with images looking similar from most viewpoints will be rated low. The pathological cases

are spherically symmetric datasets looking the same from every viewpoint (i.e. a sphere). Since animating such datasets interactively does not make much sense anyway, these pathological cases are not a major drawback for our method.

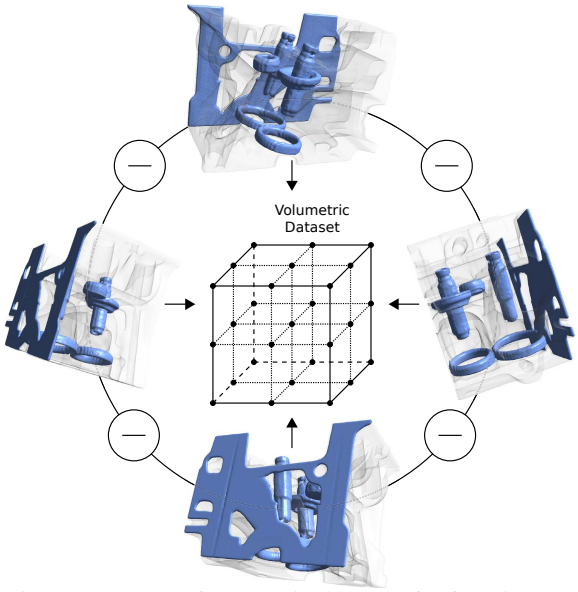


Figure 2: Images for a particular transfer function are rendered from different viewpoints. The VDE metric is then computed from the difference of these images.

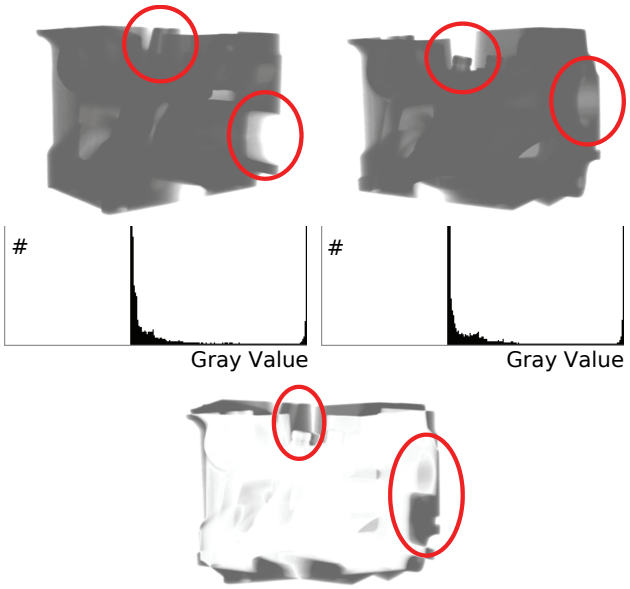


Figure 3: The engine dataset from two different viewpoints. Even though different details are shown (red circles), the histograms of the two images are nearly identical. Thus, pure statistical methods such as informational divergence are not suitable to quantify the differences. In contrast, the differential image (bottom) nicely shows the different details such as intake and exhaust duct.

An alternative to the entropy of differential images could be the informational divergence[35] of the images from different viewpoints. We decided not to take the informational divergence for good reasons. First, the entropy of differential images is symmetric, unlike the informational divergence which depends on the order in which images are processed. Consequently, the results for a volumetric dataset and its mirrored counterpart may not be the same when using informational divergence, but will be with symmetric differential images. Symmetry is also prerequisite for a metric in the mathematical sense. Second, by computing differential images not only the statistical distribution of values is important but also their spatial distribution. This is not the case for the informational divergence where only the statistical distribution is taken into account. Hence statistically similar images revealing different volume structures due to spatial changes would not be considered positively by information divergence. Figure 3 shows an example where the informational divergence fails.

Another alternative could be the Jensen-Shannon divergence.[36] The Jensen-Shannon divergence is symmetric and basically states how far away two probability distributions from the likely joint-source are. However, the Jensen-Shannon divergence still does not account for the spatial distribution. For our VDE metric the images from different viewpoints are always taken through the geometric center of the dataset. Even if a certain structure in the dataset is visible in images from different viewpoints it is not necessarily at the same position in the different images. In this case the structure will only partially or not at all cancel itself out in the differential images and therefore lead to different (possibly higher) entropy. This is strongly desired as it reflects the change of the spatial distribution due to interactive viewing (see also Figure 3).

Apart from our definition of visibility-difference entropy, two additional contributions are made. Because entropy over visibility differences is prone to noise, resulting in high ratings for overly noisy images, a noise dampening term is applied to the VDE metric. Furthermore, transfer functions that exclusively visualize only small parts of the entire volume are not as desirable, despite excessively high ratings. Consequently an additional coverage control term is introduced.

3.1 Visibility-Difference Entropy

The basis for our VDE metric is Shannon's entropy[35] defined as:

$$H(X) = E(I(X)), \tag{1}$$

where X is a discrete random variable, E is the expected value, and I is the information content of X. If X is defined over an alphabet $\{x_1,...,x_n\}$ and the information content $I(p)$ of the probability p of a character is defined as $-\log p$, the entropy can be written as:

$$H(X) = \sum_{x \in X} p_x I(p_x) = -\sum_{x \in X} p_x \log p_x. \tag{2}$$

In the context of VDE transfer function generation the alphabet X is characterized by the intensity or color space of a pixel (this can be luminance only, RGB, or HSV depending on the actual implementation). To avoid empty space having a big impact on the entropy, empty space is not considered and the corresponding value is excluded from the alphabet. Since the actual implementation only operates on gray values the alphabet consists of values $\{0,\ldots,255\}$.

The entropy is computed on the differential images as follows: Given a set of rendered images from different viewpoints T where $\tau_i \in T$ denotes the i^{th} image, $|\tau_i|$ the size of the image, Υ_i the set of pixel indices for the respective image, and $\tau_i(m,n)$ the value of the pixel at position $(m,n) \in \Upsilon_i$, the probabilities of a pair of images are obtained by dividing the histogram by the number of non-zero pixels:

$$P_{i,j}(x \in X) = \frac{\sum_{(m,n)\in\Upsilon_i} \begin{cases} 1, & \text{if } |\tau_i(m,n) - \tau_j(m,n)| = x \\ 0, & \text{otherwise} \end{cases}}{\sum_{(m,n)\in\Upsilon_i} \begin{cases} 1, & \text{if } |\tau_i(m,n) - \tau_j(m,n)| \neq 0 \\ 0, & \text{otherwise} \end{cases}} \tag{3}$$

Consequently, the visibility-difference entropy (VDE) for a pair of images is:

$$H_{i,j} = -\sum_{x \in X} P_{i,j}(x) \log P_{i,j}(x) \tag{4}$$

Finally, the entropy contribution of the VDE metric M^H can then be defined as the average over all differential images:

$$M^H = \binom{|T|}{2}^{-1} \sum_{i,j \in T \text{ where } j>i} H_{i,j} \tag{5}$$

3.2 Noise Term

So far the definition of the VDE metric is defined solely on the entropy of the differential images. A problem with this definition is that it is prone to random, uniform noise leading to high ratings from small pixel fluctuations. To compensate for this, two different measurements of noise are introduced. They are computed on individual and not on differential images because the absolute amount of noise is important and not how the noise changes when animated.

The first noise measure is the standard deviation of pixel variations in homogeneous regions. The basic idea is to subdivide each image into a set of K small uniform patches ($\approx 20^2$ pixels), and determine the overall noise as the average over the standard deviation of all non-empty patches in all images.

With τ_i^k denoting the k^{th} patch of image τ_i, $|\tau_i^k|$ the size of that patch, and Υ_i^k the respective pixel indices, the mean pixel value for that patch is

$$\phi_i^k = \frac{1}{|\tau_i^k|} \sum_{(m,n)\in\Upsilon_i^k} \tau_i^k(m,n), \tag{6}$$

and the standard deviation is

$$\sigma_i^k = \sqrt{\frac{1}{|\tau_i^k|} \sum_{(m,n)\in\Upsilon_i^k} (\phi_i^k - \tau_i^k(m,n))^2}. \tag{7}$$

The average standard deviation over all patches and all images is then:

$$\sigma = \frac{1}{|T||K|} \sum_{i \in T} \sum_{k \in K} \sigma_i^k \tag{8}$$

The standard deviation, however, does not work well if a patch contains a steep color gradient. In this case the standard deviation is high even if the patch does not contain significant noise at all. Hence another measure is introduced, which we call pixel deviation. Basically, it measures the deviation of a pixel to its direct neighbors, expecting the deviation to be relatively small for smooth color gradients but large in the presence of noise. To efficiently skip empty regions, pixel deviation is also computed on patches and averaged for all patches and images.

The pixel deviation is defined as

$$\phi_i^k(m,n) = \frac{1}{4} \sum_{\substack{u,v \in \\ \{(1,0),(-1,1),\\ (0,1),(1,1)\}}} |\tau_i^k(m,n) - \tau_i^k(m+u,n+v)|, \tag{9}$$

where for each pixel only half of the 8-neighborhood has to be considered due to symmetry. For an entire patch it is

$$o_i^k = \frac{1}{|\tau_i^k|} \sum_{(m,n)\in\Upsilon_i^k} \phi_i^k(m,n). \tag{10}$$

The average over all patches and images is then:

$$o = \frac{1}{|T||K|} \sum_{i\in T} \sum_{k\in K} o_i^k \tag{11}$$

Eventually, the noise measurements are integrated into the VDE metric in terms of an exponential drop-off windowing factor M^N. This leaves the entropy term largely unaffected if the noise is low but starts cutting off exponentially above a certain threshold level. Parameters $a_{\sigma,o}$ and $b_{\sigma,o}$ can be used to adjust the window for level and steepness (see Table 2):

$$M^N = \left(1 - \left(\frac{\sigma}{a_\sigma}\right)^{b_\sigma}\right)\left(1 - \left(\frac{o}{a_o}\right)^{b_o}\right) \tag{12}$$

With the noise term integrated, the VDE metric can be rewritten as:

$$M^{VDE} = M^H M^N \tag{13}$$

3.3 Coverage Term

To avoid empty space dominating the VDE metric, empty space was excluded when defining the entropy part. However, due to the exclusion of empty space, transfer functions which make only small parts of the volume visible may cause inappropriately high entropy measures. Therefore, we introduce a coverage term to penalize such transfer functions.

The coverage term is simply the area covered by non-zero pixels in relation to the entire area

$$\theta_i = \frac{1}{|\tau_i|} \sum_{(m,n)\in\Upsilon_i} \begin{cases} 1, & \text{if } \tau_i(m,n) <> 0 \\ 0, & \text{otherwise} \end{cases}, \tag{14}$$

and the average over all images is

$$\theta = \frac{1}{|T|} \sum_{i\in T} \theta_i. \tag{15}$$

Similar to the noise term, the coverage term is included into the VDE metric as a drop-off window filter M^C where parameter b_θ can be used to control the steepness:

$$M^C = 1 - (1-\theta)^{b_\theta} \tag{16}$$

The final version of the VDE metric is thus:

$$M^{VDE} = M^H M^N M^C \tag{17}$$

Images generated by a transfer function can now be evaluated with the VDE metric and thus the transfer function can be evaluated. Figure 4 shows an example how the VDE metric works in practice.

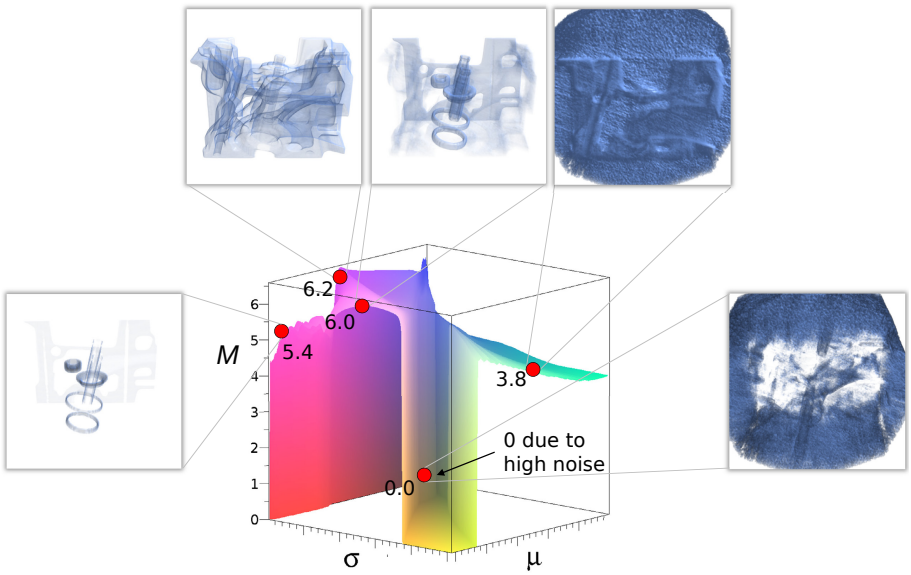
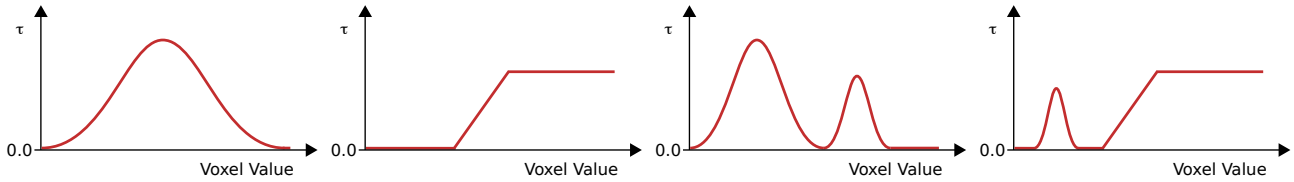


Figure 4: Function plot of the VDE metric for different transfer functions generated using parameters σ and μ. The highest rated transfer functions are along a crest allowing for good insight into the dataset. The medium rated transfer functions already substantially limit the insight into the dataset. In the middle of the plot is a region where transfer functions are penalized due to high noise. Intuitively the images confirm the semantic of the VDE metric.

4. BASIS TRANSFER FUNCTIONS

An important part when searching for the transfer functions with the highest ratings is the search space, which is the space of all transfer functions for VDE transfer function generation. Without previous knowledge, the search space must be big enough to contain good transfer functions with high ratings but it must not be too big since otherwise a search will be infeasible. For example, a mapping of 8-bit scalar to 8-bit alpha values defines a space of 256^{256} possible transfer functions, clearly a space too large to effectively search in.

A possible way to dramatically reduce the search space is to define the transfer function based on parametrized basis functions, and then search in this parametric space. In the following we propose a small set of basis functions with four to eight parameters and thus a search space of 256^4 to 256^8 combinations. Even though these are still large search spaces, it is within reach of randomized search and optimization algorithms with acceptable time cost. Our general approach also applies to other basis transfer functions and larger transfer function parameter sets, given an efficient search can be implemented. We would like to point out that VDE transfer function generation is not limited to the suggested basis transfer functions but can be used with any basis transfer function as long as it can be parametrized. This also includes the popular 2D transfer functions where the second dimension is the derivative of the voxel value. For example, the Gaussian basis function can easily be extended to a 2D transfer function by adding a single additional parameter. VDE transfer function generation will work out-of-the-box with such a 2D transfer function.



(a) (b) (c) (d)

Figure 5: The basis transfer functions: Gaussian (a), ramp (b), double Gaussian (c), ramp Gaussian (d).

Gaussian The Gaussian basis function (Figure 5(a)) is defined as

$$f^G(x) = a \cdot e^{-\left(\frac{x-\mu}{\sigma}\right)^2} + b, \tag{18}$$

where a is the amplitude, b is the base line offset, μ is the center, and σ is the width of the Gaussian. We experienced that only varying the amplitude a generates very similar images and raising the base line with b seldom produces images with high ratings. It is thus practicable to use a restricted value range for parameters a and b or to fix them.

Ramp The ramp basis function (Figure 5(b)) is defined as

$$f^R(x) = \begin{cases} c, & \text{if } x < a \\ c + (b-a)m, & \text{if } x > b \\ c + (x-a)m & \text{otherwise} \end{cases}, \tag{19}$$

where a and b are the beginning and end of the ramp, c is the base line offset, and m is the gradient. The parametric space can slightly be reduced by the observation that ramps with $c \neq 0$ for $m > 0$ or $c + (b-a)m \neq 0$ for $m < 0$ seldom produce images with high ratings.

Double Gaussian The double Gaussian (Figure 5(c)) is the combination of two individual Gaussians f^{G1} and f^{G2}:

$$f^{DG}(x) = \max(f^{G1}(x), f^{G2}(x)) \tag{20}$$

The advantage of the double Gaussian is that multiple hot spots in the definition range of the transfer function can be accentuated separately. However, the parametric space of the double Gaussian consists of eight parameters.

Ramp Gaussian The last basis function is basically the combination of a ramp with a Gaussian (Figure 5(d)):

$$f^{RG}(x) = \max(f^{R}(x), f^{G(x)}) \tag{21}$$

Often the ramp is the predominant function with the Gaussian accentuating certain ranges. Without restrictions, the parametric space consists of eight parameters.

5. TWO-STAGE OPTIMIZATION

To find transfer functions with highest ratings, a search in the parameter space of the basis functions is performed. When choosing a search algorithm various considerations have to be made. Most important is the characteristics of the value space to be searched. For VDE transfer function generation this is the output of the VDE metric over a set of images generated by the volume renderer. The renderer itself takes the volume data and the transfer function defined by a set of parameters as input (see Figure 6). The rendered images then depend on the volume data and so does the VDE metric. Experiments indicate (see also Figure 7) that the VDE metric output is not random when plotted for a particular dataset and different parameters but has some structure to it, which is important for a search strategy. Furthermore, the main goal is not to find the single global maximum but rather a set of local maxima or near-maxima located as far away from each other as possible. Intuitively, this results in transfer functions with high ratings that are strongly distinct in what they reveal. Finally, evaluating the VDE metric is expensive, as we have to render and process many images, and is limited by the graphics hardware. Hence, the goal is to test as few parameter sets as possible to keep the search time within acceptable limits.

If the function plot of the VDE metric is as smooth as in Figure 7(a), gradient ascent[37] can be chosen for quickly approaching a local maximum. However, the function plot is not always that smooth and may contain many local maxima as exhibited in the green part of Figure 7(b). In this case gradient ascent starting from a coarse set of initial seed positions would head straight for the nearest local maxima not considering any other local maxima farther away. Simulated annealing[38] on the other hand is a randomized search method that can find widely distributed maxima without requiring densely placed seeds. We combine simulated annealing and gradient ascent to profit from the advantages of both methods. In smooth regions of the search space gradient ascent is preferred to quickly approach the local maximum, while in rugged areas with many local maxima simulated annealing is better to identify the best local maxima with a certain probability. For a seed point, the idea is to quickly find out if the neighborhood is sufficiently smooth to continue with gradient ascent.

Given a seed point, initially simulated annealing is executed for a number of iterations and all values obtained during these iterations are recorded. Since simulated annealing only moves to a direct neighbor in each iteration, the sequence

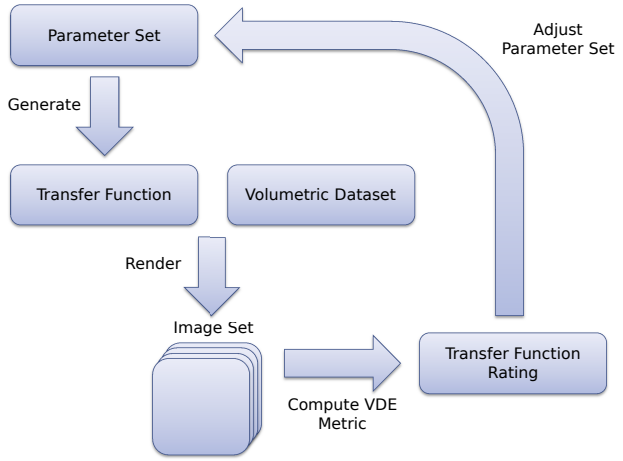
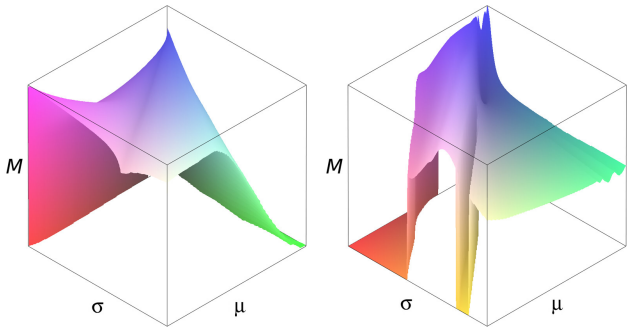


Figure 6: The optimization loop for the transfer function search. The parameter set is used to generate the transfer function from a basis function. Together with the volumetric dataset a set of images from different viewpoints is rendered and the VDE metric is computed. Finally, the parameter set is adjusted according to the search algorithm.



(a) (b)

Figure 7: Function plots of the VDE metric for a Gaussian transfer function with varying parameters (μ, σ) for the aneurysm (a) and skull dataset (b). Both plots show a structure and a limited number of local maxima.

of recorded values can be used to analyze the immediate neighborhood of the starting point. Thus we use the standard deviation of the recorded values as an indicator for smoothness of a seed point

$$\sigma^S = \sqrt{\frac{1}{|X|}\sum_{x\in X}(\emptyset - f(x))^2}, \tag{22}$$

where X is the set of all positions recorded during simulated annealing and $\emptyset$ is the average of all values processed in X. If this standard deviation is above a certain threshold, the search is continued with simulated annealing, otherwise gradient ascent is applied on this seed point.

6. ADAPTIVE SEEDING

The search is conducted starting from a number of initial seeds where the time required for the entire search correlates to this number. Choosing too few seeds increases the risk of missing interesting areas whereas choosing too many seeds takes a lot of time. The parametric space is subdivided into a grid (i.e. stratified sampling) with as many initial cells as seeds, and each seed is randomly placed in an empty cell. Changing the initial conditions by re-initializing the seeds may be used in case a search does not yield acceptable solutions.

To further improve the search, the potential of all cells is rated after each n iterations, where n is a user defined number. If a cell shows high potential, it is subdivided into smaller cells with additional seeds. In contrast, if a cell is not showing good potential it is aborted prematurely. Since initial seeds exhibiting a low potential are eliminated after the very first iteration, as explained in more detail in the next section, the potential of a cell is estimated by its variation rather than by its absolute value. For this the standard deviation σ^S from Equation 22 is used. If the standard deviation of a cell is high, it is likely that the cell contains high frequencies requiring additional seeds to better search that region. In contrast, if σ^S is very low, it is likely that the cell contains a more uniform value space. Hence the decision if a cell is further subdivided or aborted is made by comparing the standard deviation σ_i^S of a cell i to a threshold range. If σ_i^S is higher than an upper threshold the cell is subdivided until a maximum number of subdivisions is reached, whereas the cell is aborted if σ_i^S is less than a lower threshold.

7. IMPLEMENTATION

Our VDE transfer function generation is implemented using GPU volume ray-casting as described in.[39,40] First, for a given volume dataset one of the available basis transfer functions is selected. Since the focus lies on the visibility, transparency

and the structure of the volumetric data and not so much on the color, without restriction to generality, we compute the VDE metric on gray values only. This reduces the size of the alphabet for computing the entropy from 24 bits for a typical *RGB* pixel to 8 bits. The transition from color to gray values is achieved by the perceptually motivated linear combination of $0.3 \cdot \text{red} + 0.59 \cdot \text{green} + 0.11 \cdot \text{blue}$. The basic processing flow of VDE transfer function generation includes:

- Initialization of seeds.
  1. Execution and evaluation of simulated annealing/gradient ascent search iterations including update evaluation of current VDE metric results.
  2. Evaluation of subdivision and abortion criteria for the cells and seeds respectively.
  3. While not all seeds are completed, go back to 1.
- Computation of the similarity index between top transfer functions and presentation of the filtered and grouped results to the user (see also Figure 11).

After each iteration over all active seeds, the results are evaluated and either stored in the list the top results or dropped. Apart from the VDE metric, the evaluation in Step 2. includes filtering based on similarity. This is necessary due to the likelihood of many seeds around the global or a high local maximum producing equally looking images with high ratings. If a result is similar to another existing result in the top list, the result with the higher rating is kept and the other one is dropped. The similarity is computed by the Euclidean distance in the parametric space of the basis function. Given a basis function f, two results are considered similar if

$$\|x - x'\| < \varepsilon \text{ for } x, x' \in \text{dom}(f), \tag{23}$$

where x and x' are parameter sets and ε is a threshold.

An exception is the first iteration over all seeds. At the beginning a single iteration over all initial seeds is performed and the seeds are ranked as described above. Then, the top seeds, currently set to 25, are selected for further processing and all other seeds are dropped. This is due to diminishing returns from using hundreds of seeds for which the process would take significantly more time. Before presenting the results to the user, a final similarity index is computed between all results of the top list and the results are grouped according to this index. The similarity index is computed in image space from thumbnail images. The reason is that the results should be coherently grouped based on the thumbnail images presented to the user. For this, first the histograms of all thumbnails are computed:

$$G_i(x \in X) = \sum_{(m,n)\in\Omega_i} \begin{cases} 1, & \text{if } \psi_i(m,n) = x \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

where ψ_i is the i^{th} thumbnail of the set of thumbnails Ψ, Ω_i is the set of pixel indices of the respective thumbnail, and $\psi_i(m,n)$ corresponds to the pixel at position $(m,n) \in \Omega_i$. X is the dynamic range of the computed gray value, which is typically 8 bit. Then the normalized similarity index is the average distance to all other thumbnails:

$$S_i = \frac{1}{|\Psi||X|\max(X)} \sum_{j\in\Psi\setminus\{i\}} \sum_{x\in X} |G_i(x) - G_j(x)| \tag{25}$$

8. RESULTS AND DISCUSSION

8.1 VDE Transfer Functions

All experiments were performed on a 2.9GHz Intel Core i7 computer with NVIDIA GeForce GTX 680 graphics. If not otherwise stated a viewport of 800^2 pixels was used. The images presented in Figures 1, 8, 10, and 12 were created as follows. First, VDE transfer function generation was blindly applied to each input dataset, taking a couple of minutes to generate a set of 20 best transfer functions. Second, the images were generated by picking one from the set of best transfer functions (see Figure 11 for the respective GUI), choosing a single color, selecting a default set of lighting parameters and rendering them with a GPU volume ray-caster. Unless a gray material is desired, a color must be chosen because VDE

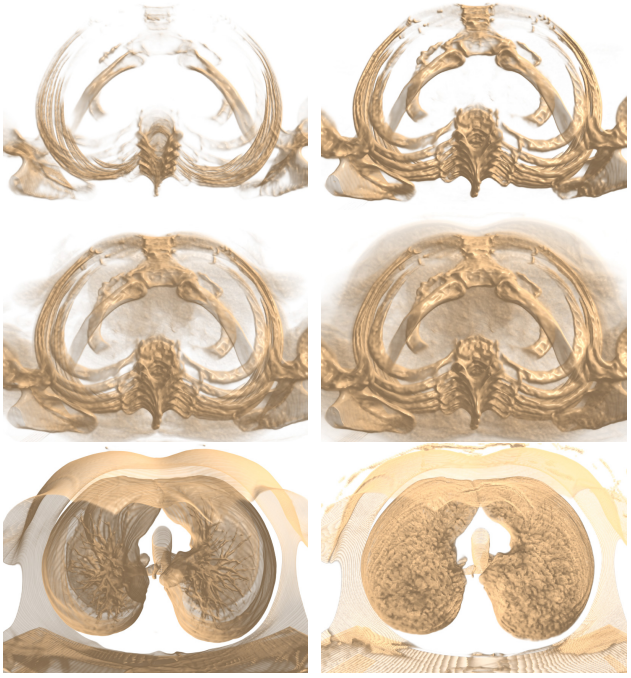


Figure 8: Different insights into the chest dataset. The transfer functions have been picked from the set of best Gaussian and ramp transfer functions. As expected, they focus on different features and parts allowing for identification of major structures of the dataset. http://dx.doi.org/doi.number.goes.here

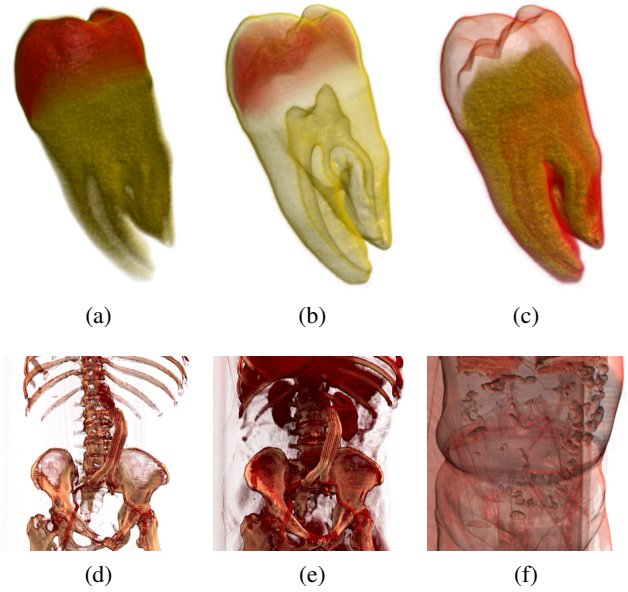


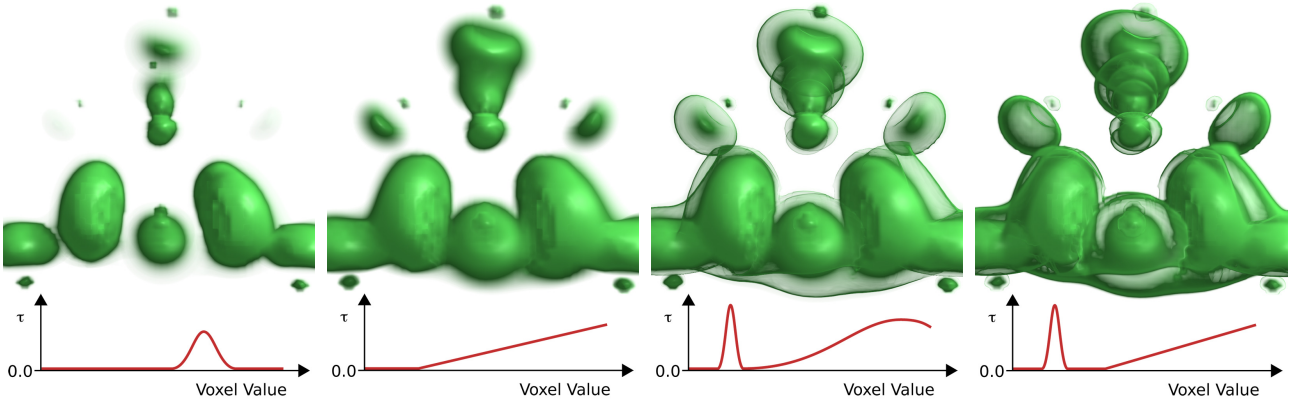
Figure 9: Different insights into the tooth and the stent dataset achieved by our VDE transfer function generation. (a) Shows the root and crown of the tooth, (b) shows the dental pulp, (c) shows the dentin. (d) Shows the skeleton and the stent, (e) shows the skeleton, the stent, the kidney and parts of the lungs, and (f) shows the body shape and some blood vessels.

transfer function generation currently only considers the opacity channel. If the dataset originates from a CT scan, it is also possible to apply a default color map for the Hounsfield scale. The images in Figure 9 were created in the same way, except that more than one color was applied. Also the images in Figure 13 were created in the same way except that the dataset was pre-segmented and a different color was chosen for each segment. All examples show that transfer functions from optimizing the VDE metric indeed enable distinct insights and identification of major features and structures.

In Figure 8, the chest is rendered semi-transparently with one of the generated transfer functions to show the skeleton, while the lungs are omitted. Another time the trachea, lungs, and bronchi are revealed while the skeleton part is suppressed.

Figure 9 highlights the difference of our method to the most recent approaches.[18,19] The approach from Ruiz et al.[19] requires the user to specify a target distribution first. Even though they do not describe how they do it for the tooth dataset their respective results look similar to the image in Figure 9(b). In contrast, our method does not require specifying a target distribution and yet we find similar results. Additionally, we also get transfer functions that focus on other parts of the dataset such as the crown and root of the tooth (Figure 9(a)) or the dentin (Figure 9(c)), which would require adjusting the target distribution by the user in the approach of Ruiz et al. The approach of Correa and Ma[18] requires the user to provide an initial transfer function which is then optimized in order to maximize voxel visibility taking into account the divergence to the provided transfer function. Unless the user exactly knows what he wants, this leads typically to results like in Figure 9(e) where the overall voxel visibility is quite high. On the other hand it's difficult to obtain a result like in Figure 9(f) without basically providing the final transfer function as initial transfer function. In contrast, our method does not require any initial transfer function and yet we obtain results with high voxel visibility (Figure 9(e)) but also other ones with focus on different parts such as the skeleton and stent (Figure 9(d)), or the body shape and blood vessels (Figure 9(f)). However, an advantage of the approach of Correa and Ma is the possibility to view-dependently re-optimize a transfer function for visibility, which could be applied on top of initial VDE transfer functions.

Figure 10 demonstrates the effect of the different basis transfer functions. For this, a set of best transfer functions was generated for each type of basis transfer function and a good resulting transfer function was picked from each set.



(a) (b) (c) (d)

Figure 10: Impact of the basis transfer function to the generated transfer functions. All images are rendered with a transfer function from the set of best transfer functions where (a) is based on a Gaussian, (b) on a ramp, (c) on a double Gaussian, and (d) on a combined ramp Gaussian. The semi-transparent surface from the steep Gaussian in (c) and (d) is well visible. http://dx.doi.org/doi.number.goes.here

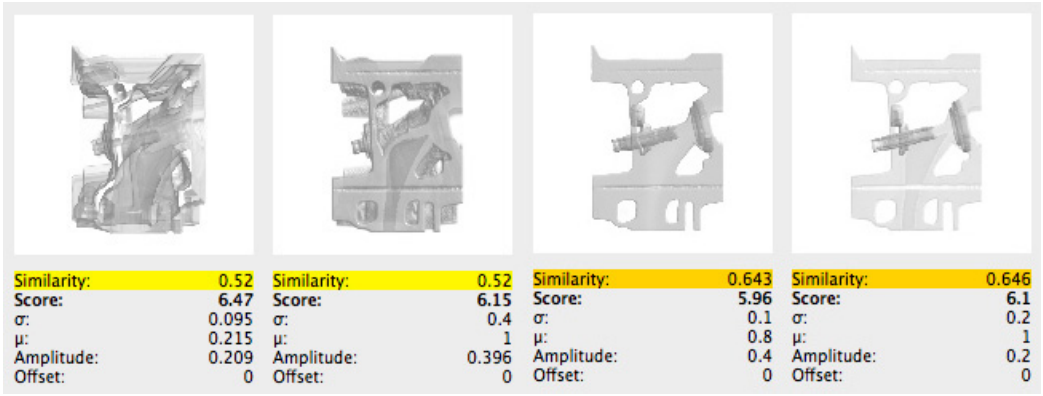


Figure 11: Excerpt from the GUI where the generated transfer functions are presented to the user including VDE metric score, similarity index, and parameters. The transfer functions can be applied by simply clicking the thumbnail images. http://dx.doi.org/doi.number.goes.here

Remarkably, for the double and combined ramp Gaussian basis functions, a steep Gaussian reflects the outer shell of the electron probability distribution of the neghip dataset. This is very well visible in the images as a semi-transparent surface.

Table 1 presents an overview of the timings required for generating the set of best VDE transfer functions for different datasets and basis transfer functions respectively. The time required for generating a set of best transfer functions is directly related to the total number of iterations executed and the performance of the GPU volume ray-caster, thus the VDE metric evaluation. The number of iterations is dependent on the particular dataset and ranges typically from a few hundred to a few thousand. For example, the average number of iterations for the Gaussian basis function and all datasets from Table 1 is 1820. Any optimization algorithm that can reduce the number of VDE metric evaluations will improve the performance.

8.2 User Experiments

In addition, Table 1 shows the results from a small user study and survey. The hypothesis of the study is that VDE transfer function generation delivers a set of transfer functions which makes at least as many features of the respective dataset identifiable as a similar set created by a user in the same amount of time. Identifiable means that large parts of a particular feature are recognizable, but it is not distinguished whether the feature is rendered opaque or semi-transparent. The null hypothesis is that there is no difference in terms of identifiable features between automatically generated VDE transfer functions and the transfer functions created by the study participants. 11 undergraduate and Ph.D. computer science students with at least basic knowledge in computer graphics and visualization took part in the study. They were

Dataset	Volume Size	Timings [min]				Study [# Features]			Survey [Rating]	
		Gaussian	Ramp	Double Gaussian	Ramp Gaussian	VDE TFG	Users ø	t-test [p]	ø	σ
Engine	256x256x128	7:52	8:38	8:52	20:14	15	13.1	0.041	4.0	0.66
Chest	384x384x240	3:04	4:00	8:57	15:27	10	7.0	0.000	4.2	0.42
Pelvis	512x512x461	3:42	3:56	8:43	13:51	-	-	-	-	-
Stent	512x512x174	5:12	5:41	13:10	17:39	-	-	-	-	-
Tooth	256x256x161	5:00	5:10	11:53	21:54	-	-	-	-	-
Neghip	64x64x64	10:06	11:00	14:21	17:21	-	-	-	-	-
Aneurysm	256x256x256	3:04	3:54	9:51	14:24	-	-	-	-	-
Feet	512x512x250	12:58	14:59	19:05	22:33	-	-	-	3.6	0.70
Fuel	64x64x64	5:21	6:33	7:01	10:52	-	-	-	2.3	0.48
Head	128x256x256	10:15	11:11	16:32	21:11	-	-	-	-	-
Heart	512x512x75	3:56	3:44	10:41	13:30	-	-	-	-	-
Knee	379x229x305	3:57	6:17	10:02	14:30	7	6.4	0.095	-	-
Skull	256x256x256	10:57	6:22	12:36	22:12	-	-	-	-	-

Table 1: Timings (minutes), results of the user study, and of the survey for VDE transfer function generation for different datasets. For the user study the table shows the number of identifiable features when using the transfer functions generated by VDE transfer function generation (VDETFG) and the transfer functions created manually by the study participants respectively. For the survey the scale goes from 1 corresponding to "not useful at all" to 5 corresponding to "very useful".

asked to manually construct transfer functions using a Gaussian basis function with the goal of revealing as many features as possible and a time limit equal to the time required for the automatic generation of the transfer function set. The results show that on average the automatically generated Gaussian transfer functions make more features identifiable than the transfer functions created by the study participants and therefore support the hypothesis. The null hypothesis can be rejected for a 10% confidence interval (see Table 1). For the survey 10 computer science expert and non-expert users were asked how useful they consider VDE transfer function generation for the respective dataset where a score of 1 corresponds to "not useful at all" and 5 corresponds to "very useful". The results indicate that the survey participants consider VDE transfer function generation generally useful, maybe with the exception of the very simple fuel dataset.

8.3 Limitations

VDE based transfer function generation is not limited to specific basis transfer functions except that the performance does depend on the parameter search space. With further improved optimization algorithms we can expect to directly extend the presented method to cover a growing range of parametrized basis transfer functions, suitable for a wide range of application scenarios. In future work we would like to take colors into account and therefore generate full RGBA transfer functions. For example this can be achieved by using a predefined set of colors as an additional parameter. VDE transfer function generation requires a couple of parameters to be set. The parameters have shown to be independent of the particular dataset and thus can be fixed. We suggest starting with our parameters from Table 2, which have proven to work well.

Figure 12: Example scan of the inside of a tooth. This dataset has no clear structures with clean boundaries. Nevertheless VDE transfer function generation is able to uncover growth patterns.

ACKNOWLEDGMENTS

The authors wish to thank volvis.org for the engine, neghip, and stent model, Department of Radiology, University of Iowa for the chest data, GE Aircraft Engines for the tooth data, and OsiriX for the pelvis DICOM images. The authors also

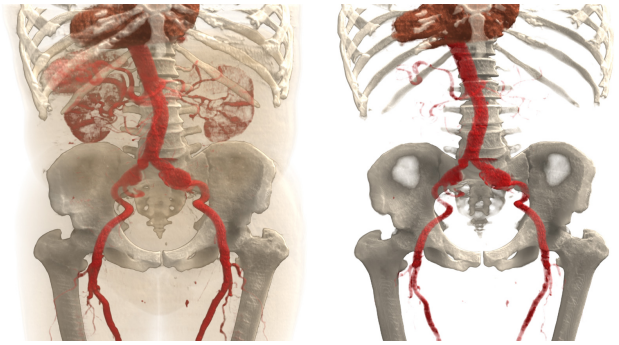


Figure 13: Different insights into the segmented pelvis dataset. The transfer functions have been picked from the set of best ramp transfer functions. Additionally, a different color has been applied to each segment.

Parameter	Value
Number of viewpoints	6
a_σ and a_o	14
b_σ and b_o	7
b_θ	30
Gaussian parameter range: a and b	$1/2$ and $1/3$
Ramp parameter range: $b-a$ and c	$1/2$ and $1/3$
Double Gaussian parameter range: a and b	$1/5$ and 0
Ramp Gaussian parameter range: $b-a$ and c	$1/5$ and 0
Seeds for Gaussian and Ramp	216
Seeds for double Gaussian and ramp Gaussian	2500
Seed cancellation and subdivision threshold	0.6 and 3.2
Similarity ε	0.05

Table 2: Parameter set used for all experiments.

wish to thank for the tooth (growth structures) dataset, which was made available through the Anthropological Institute and Museum, University of Zurich.

REFERENCES

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

[2] Bajaj, C. L., Pascucci, V., and Schikore, D. R., "The contour spectrum," in [*Proceedings of the Conference on Visualization*], 167–175 (1997).

[3] Zhou, J. and Takatsuka, M., "Automatic transfer function generation using contour tree controlled residue flow model and color harmonics," *IEEE Transactions on Visualization and Computer Graphics* **15**, 1481–1488 (2009).

[4] Fujishiro, I., Azuma, T., and Takeshima, Y., "Automating transfer function design for comprehensible volume rendering based on 3D field topology analysis," in [*Proceedings of the Conference on Visualization*], 467–470 (1999).

[5] Weber, G. H. and Scheuermann, G., [*Geometric Modeling for Scientific Visualization*], ch. Automating Transfer Function Design Based on Topology Analysis, 293–306, Springer (2004).

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

[7] Zhang, J.-W. and Sun, J.-Z., "Adaptive transfer function design for volume rendering by using a general regression neural network," in [*International Conference on Machine Learning and Cybernetics*], **4**, 2234–2239 (2003).

[8] Duch, W. and Jankowski, N., "New neural transfer functions," *Neural Computing Surveys* **7**, 639–658 (1997).

[9] Hall, L., Bensaid, A., Clarke, L., Velthuizen, R., Silbiger, M., and Bezdek, J., "A comparison of neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain," *IEEE Transactions on Neural Networks* **3**(5), 672–682 (1992).

[10] Sato, Y., Westin, C.-F., Bhalerao, A., Nakajima, S., Shiraga, N., Tamura, S., and Kikinis, R., "Tissue classification based on 3D local intensity structures for volume rendering," in [*IEEE Transactions on Visualization and Computer Graphics*], 160–180 (2000).

[11] Lundstrom, C., Ljung, P., and Ynnerman, A., "Local histograms for design of transfer functions in direct volume rendering," *IEEE Transactions on Visualization and Computer Graphics* **12**, 1570–1579 (2006).

[12] Šereda, P., Vilanova Bartroli, A., Serlie, I. W. O., and Gerritsen, F. A., "Visualization of boundaries in volumetric data sets using lh histograms," in [*IEEE Transactions on Visualization and Computer Graphics*], **12**, 208–218 (2006).

[13] Wang, Y., Chen, W., Zhang, J., Dong, T., Shan, G., and Chi, X., "Efficient volume exploration using the Gaussian mixture model," *IEEE Transactions on Visualization and Computer Graphics* **17**, 1560–1573 (November 2011).

[14] Fang, S., Tom, B., and Tuceryan, M., "Image-based transfer function design for data exploration in volume visualization," in [*Proceedings of the Conference on Visualization*], 319–326 (1998).

[15] He, T., Hong, L., Kaufman, A., and Pfister, H., "Generation of transfer functions with stochastic search techniques," in [*Proceedings of the Conference on Visualization*], 227–234 (1996).

[16] Li, K., Qi, R., Xiao, D., Yang, L., and Li, Z., "Applying particle swarm optimization to transfer function specification for direct volume rendering," in [*Proceedings of the International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing*], 573–576 (2009).
[17] Marks, J., Andalman, B., Beardsley, P. A., Freeman, W., Gibson, S., Hodgins, J., Kang, T., Mirtich, B., Pfister, H., Ruml, W., Ryall, K., Seims, J., and Shieber, S., "Design galleries: a general approach to setting parameters for computer graphics and animation," in [*Proceedings of the Conference on Computer Graphics and Interactive Techniques*], 389–400 (1997).
[18] Correa, C. D. and Ma, K.-L., "Visibility histograms and visibility-driven transfer functions," *IEEE Transactions on Visualization and Computer Graphics* **17**, 192–204 (February 2011).
[19] Ruiz, M., Bardera, A., Boada, I., and Viola, I., "Automatic transfer functions based on informational divergence," *IEEE Transactions on Visualization and Computer Graphics* **17**, 1932–1941 (dec. 2011).
[20] König, A. and Gröller, M. E., "Mastering transfer function specification by using volumepro technology," in [*Proceedings of the Spring Conference on Computer Graphics*], 279–286 (2001).
[21] Tzeng, F.-Y. and Ma, K.-L., "A cluster-space visual interface for arbitrary dimensional classification of volume data," in [*IEEE TCVG Symposium on Visualization*], 17–24 (2004).
[22] Rezk Salama, C., Keller, M., and Kohlmann, P., "High-level user interfaces for transfer function design with semantics," *IEEE Transactions on Visualization and Computer Graphics* **12**, 1021–1028 (2006).
[23] Wang, L. and Mueller, K., "Harmonic colormaps for volume visualization," in [*IEEE/EG Symposium on Volume Graphics*], 30–40 (2008).
[24] Kniss, J., Kindlmann, G., and Hansen, C., "Multidimensional transfer functions for interactive volume rendering," *IEEE Transactions on Visualization and Computer Graphics* **8**(3), 270–285 (2002).
[25] Ruiz, M., Boada, I., Viola, I., Bruckner, S., Feixas, M., and Sbert, M., "Obscurance-based volume rendering framework," in [*Proceedings IEEE/EG Symposium on Volume and Point-Based Graphics*], 113–120 (2008).
[26] Vázquez, P.-P., Feixas, M., Sbert, M., and Heidrich, W., "Automatic view selection using viewpoint entropy and its application to image-based modelling," *Computer Graphics Forum* **22**(4), 689–700 (2004).
[27] Nakagawa, M., Takata, M., and Joe, K., "Automatic viewpoint selection for a visualization I/F in a PSE," in [*Proceedings IEEE International Conference on e-Science and and Grid Computing*], 100–106 (2006).
[28] Feixas, M., Sbert, M., and González, F., "A unified information-theoretic framework for viewpoint selection and mesh saliency," *ACM Transactions on Applied Perception* **6**(1), 1–23 (2009).
[29] Tao, Y., Lin, H., Bao, H., Dong, F., and Clapworthy, G., "Structure-aware viewpoint selection for volume visualization," in [*Proceedings IEEE Pacific Visualization Symposium*], 193–200 (2009).
[30] Mortara, M. and Spagnuoloa, M., "Semantics-driven best view of 3D shapes," *Computers & Graphics* **33**(3), 280–290 (2009).
[31] Takahashi, S., Fujishiro, I., Takeshima, Y., and Nishita, T., "A feature-driven approach to locating optimal viewpoints for volume visualization," in [*IEEE Visualization*], 495–502 (2005).
[32] Laga, H., "Semantics-driven approach for automatic selection of best views of 3D shapes," in [*Proceedings Eurographics Workshop on 3D Object Retrieval*], 15–22 (2010).
[33] Itti, L., Koch, C., and Niebur, E., "A model of saliencey-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**, 1254–1259 (November 1998).
[34] Jänicke, H. and Chen, M., "A salience-based quality metric for visualization," *Computer Graphics Forum* **29**, 1183–1192 (June 2010).
[35] Gray, R. M., [*Entropy and Information Theory*], Springer, 2nd ed. (2011).
[36] Bordoloi, U. D. and Shen, H.-W., "Viewpoint evaluation for volume rendering," in [*IEEE Visualization*], 487–494 (2005).
[37] Snyman, J., [*Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*], Springer (2005).
[38] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P., "Optimization by simulated annealing," *Science* **220**(4598), 671–680 (1983).
[39] Kruger, J. and Westermann, R., "Acceleration techniques for GPU-based volume rendering," in [*Proceedings IEEE Visualization*], 287–292 (2003).
[40] Scharsach, H., "Advanced GPU raycasting," in [*Proceedings Central European Seminar on Computer Graphics (CESCG)*], 69–76 (2005).