Eurographics Conference on Visualization 2012
S. Bruckner, S. Miksch, and H. Pfister
(Guest Editors)

Volume 31 (2012), Number 3

Automating Transfer Function Design with Valley Cell-Based Clustering of 2D Density Plots

Yunhai Wang[1], Jian Zhang[2], Dirk J. Lehmann[3], Holger Theisel[3], and Xuebin Chi[2]

[1]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[2]Computer Network Information Center, Chinese Academy of Sciences, China
[3]Department of Simulation and Graphics at the University of Magdeburg, Germany

Abstract

Two-dimensional transfer functions are an effective and well-accepted tool in volume classification. The design of them mostly depends on the user's experience and thus remains a challenge. Therefore, we present an approach in this paper to automate the transfer function design based on 2D density plots. By exploiting their smoothness, we adopted the Morse theory to automatically decompose the feature space into a set of valley cells. We design a simplification process based on cell separability to eliminate cells which are mainly caused by noise in the original volume data. Boundary persistence is first introduced to measure the separability between adjacent cells and to suitably merge them. Afterward, a reasonable classification result is achieved where each cell represents a potential feature in the volume data. This classification procedure is automatic and facilitates an arbitrary number and shape of features in the feature space. The opacity of each feature is determined by its persistence and size. To further incorporate the user's prior knowledge, a hierarchical feature representation is created by successive merging of the cells. With this representation, the user is allowed to merge or split features of interest and set opacity and color freely. Experiments on various volumetric data sets demonstrate the effectiveness and usefulness of our approach in transfer function generation.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Direct volume rendering is a powerful and flexible visualization technique for exploring scalar volume data. By using appropriate transfer functions an amount of important structures of the data can be revealed. The use of multi-dimensional transfer functions has already attracted much attention due to their unique capability to identify various structures within a volume. In particular, certain 2D transfer functions based on scalar values and gradient magnitudes are very effective in extracting multiple materials and their boundaries [Lev88]. The 2D transfer functions can be specified through an iterative process in which the user has to place widgets on the potential regions within a feature space and then manipulate them according to the rendering results. Without having adequate prior data knowledge this is a time-consuming process. Moreover, the commonly used widgets (e.g., rectangle, triangle, ellipse) are regular, while features located in the multi-dimensional feature space have complex shapes. Thus, extensive interaction is required to properly identify these features.

In order to ease the interaction, many proposed algorithms decompose the feature spaces into several meaningful clusters [MWCE09, RBS05, ŠVG06, TM04, WCZ*11] and allow the user to explore these clusters. Although the classification results produced by these algorithms seem to be reasonable, most of them require some input from the user, e.g., to specify the number of clusters. This poses a difficulty on interactive volume classification, especially when the user is not familiar with the data. Furthermore, several of these methods are based on traditional discrete histograms, which ignore the spatial relationship of neighboring data. For most of the data sets, they may contain spurious peaks and valleys which will reduce the performance of clustering algorithms. More

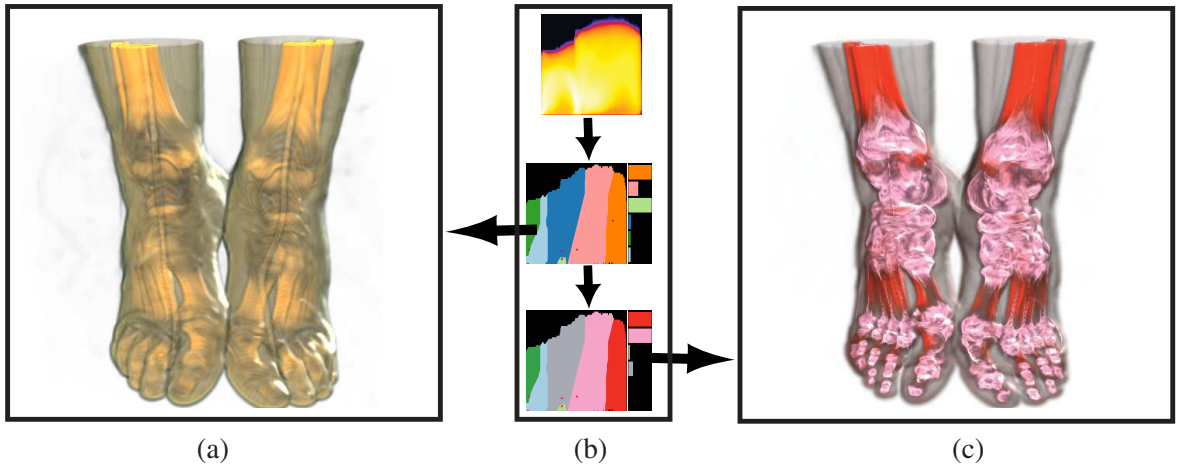


Figure 1: *Clustering the density plot on the value (x-axis) and value gradient magnitude (y-axis) feature space to visualize the Feet data set. By decomposing the density plots (top one in (b)), the automatically generated transfer function (middle one in (b)) is obtained. The corresponding rendering result (a) reveals the relationship between the skin and the bones. After recoloring the bone and lowering the opacity of soft tissue, the ankle and toe bones are clearly shown in (c). The bar charts in (b) represent the opacities of the corresponding features.*

importantly, they may lead to some feature being glossed over or completely missed [BW08].

In this paper, we propose an approach to automate the generation of 2D transfer functions with neither assumption on the number nor the shape of the clusters in the feature space. Rather than brushing the 2D histogram, our approach is based on the analysis of the density plots, which can draw the viewer to preattentively identify clusters while avoiding uncertain ones [FKLT10]. In general, a density function of the density plots can be constructed by kernel density estimation [MWCE09] or continuous scatterplots [BW08]. The former estimates distribution comes from the statistical uncertainty of the samples themselves, the latter considers interpolation between data voxels and conservation of physical variables. We refer to [MWCE09] and [BW08] for detailed description of these methods. Following the principle that features tend to form peaks of the density function [Par96], we use the theory of Morse complex [Sma61] to decompose the feature space into several *valley cells* separated by valley lines. These cells form the initial clustering where each one contains a local maximum.

Due to the noise in the original data, the initial clustering contains a large number of small valley cells. Noise artifacts disturb the quality of the visualization. Therefore, inspired by the persistence-based topological simplification [EHZ03], we merge adjacent valley cells according to their separability. We define boundary persistence of two adjacent cells, which measures the degree of separability. A threshold, according to some heuristics [GNP*06], is determined by analyzing the persistence histogram. All boundaries whose persistences are smaller than the threshold are then successively eliminated and the corresponding cells are merged together to suppress the noise. The final configuration of cells represents a potential feature of the data.

This automatic classification and transfer function design scheme can help the user to get a fast grasp on the spatial relationship between features. However, further exploration such as refining the clustering, adjusting opacities and colors are usually necessary to produce a satisfactory rendering result. There are many reasons for these user interactions, one being that human beings are often not able to identify more than five targets [Hea96]. In the case that there are many statistically significant features in volume data, showing them all in one rendering is neither simple nor appropriate. Another reason is that the user may have some prior knowledge about the features of interest and some preference about how they should be presented. Thus, we further design a persistence-based hierarchical representation of the semantical features by successive simplification of the automatic classification. Based on this feature hierarchy, the user is allowed to merge or split features and set opacities and colors for them. In summary, the main contributions include:

- a new topology-based non-parametric clustering algorithm that structures the 2D density plots into several meaningful volumetric features,
- a topological simplification of valley cells based on the proposed boundary persistence to remove noisy cells,
- and a new transfer function design scheme that allows the user to effectively explore the volume data.

We present an approach to automate the transfer function design. Here, automating means that the initial transfer func-

tion is automatically generated and the feature hierarchy is automatically adjusted within the user interaction. Fig. 1 shows an example of our classification results on the CT Feet data set by clustering the density plot of the value and value gradient magnitude. The rest of the paper is organized as follows. We provide the related work in Section 2. The valley cell decomposition and automatic transfer function design are described in Section 3 and Section 4. Finally, the implementation and experiment results are presented in Section 5 and the conclusions are drawn in Section 6.

2. Related Work

Previous work related to our work is divided into three different categories: multi-dimensional transfer function, topology-based transfer function, and Morse-Smale (MS) complex.

Multi-dimensional Transfer Function. Since 1D transfer functions cannot identify different materials with similar intensities, the 2D transfer functions based on scalar values and gradient magnitudes are proposed [Lev88]. They are very effective in extracting multiple materials and their boundaries. Inspired by this work, several effective metrics are incorporated into the feature space, such as curvature [KWTM03], feature size [CM08], and ambient occlusion [CM09]. With the feature space constructed by these metrics, the structures of interest in a volumetric data set can be characterized. However, adding dimensions to the feature space further complicates the problem of the transfer function design.

To reduce the number of degrees of freedom in transfer function design, several semi-automatic methods are proposed which provide the user with some suggestive transfer functions. By using machine learning algorithms to analyze these feature spaces, such as the ISODATA [TM04], hierarchal clustering [ŠVG06], kernel density estimation [MWCE09], and Gaussian mixture model [WCZ*11], the suggestive transfer functions can be automatically obtained. Unfortunately, most of these algorithms require the user to specify several parameters before the clustering. This poses a great workload to the user. Following [MWCE09], our clustering method also works on the continuous feature space but it does not require the number of the histogram bins. Although the hierarchal clustering algorithm [ŠVG06] does not require the number of clusters, its classification results are sensitive to the initial clustering result while our density function-based clustering resolves this issue.

Topology-based Transfer Function. Because topology is convenient to characterize global structures of the data set, it has been introduced into the transfer function specification. Based on the analysis of 3D field topology, Fujishiro et al. [FAT99] proposed an automating transfer function design scheme. Takahashi et al. [TTFN05] used the topological attributes derived from the contour tree to define transfer functions. In order to distinguish distinct features that share the same scalar value, Weber et al. [WDC*07] indexed various subregions of a volume by using contour tree and applied separate transfer functions for each subregion. Zhou et al. [ZT09] further extended this work by introducing contour tree-controlled residue flow model and color harmonic to automatically generate an appropriate transfer function for each subregion. Unlike these works, our work focuses on the automating of the 2D transfer function design based on the analysis of the topological structures of 2D density plots.

Morse-Smale Complex. The MS complex provides an abstract representation of the gradient flow behavior of a scalar field [Sma61]. However, it was originally developed for smoothing functions. By extending its construction to piecewise linear 2-manifolds, Edelsbrunner et al. [EHZ03] used it to perform a controlled simplification of the height field. Bremer et al. [BHEP04] improved this algorithm and described a multiresolution representation to approximate the simplified results. Through repeated application of atomic cancellation operations, Gyulassy et al. [GNP*06] used discrete MS complex to simplify volumetric data. To accurately extract salient edges on the surface, Weinkauf et al. [WG09] introduced a new concept of separatrix persistence, which treats ridge lines and valley lines independently. By measuring the significance of the boundary between two adjacent cells, our proposed boundary persistence can be used to maximize the separability of the extracted structures.

3. Topology-based Non-Parametric Clustering

Our topology-based clustering method decomposes the feature space into several clusters. Compared to previous methods [MWCE09, WCZ*11], our method does not make any assumption on the number or shape of the clusters. In order to perform peak-valley analysis, we apply the theory of Morse complex [Sma61]. For a smooth function, non-intersecting ridge lines and valley lines exit between saddles and extreme where each saddle is connected with two maxima via ridge lines and two minima via valley lines. The valley (ridge) lines together with their ending points decompose the domain into non-overlapping valley (ridge) cells where each one contains an isolated maximum (minimum).

Our decomposition algorithm is inspired by the Morse complex construction scheme in [EHZ03]. As in [EHZ03], we tessellate the feature space into a 2D triangular mesh, which enforces a piecewise linear approximation of the density function. This discrete representation enables a critical point identification and ridge line and valley line tracing. However, it also introduces degeneracy, for example, multi-fold saddles and intersecting valley or ridge lines. Special treatments such as path duplication and extension are proposed in [EHZ03] to deal with the degeneracy and construct the Morse complex. The main difference between our algorithm and [EHZ03] is that our goal is to extract valley cells

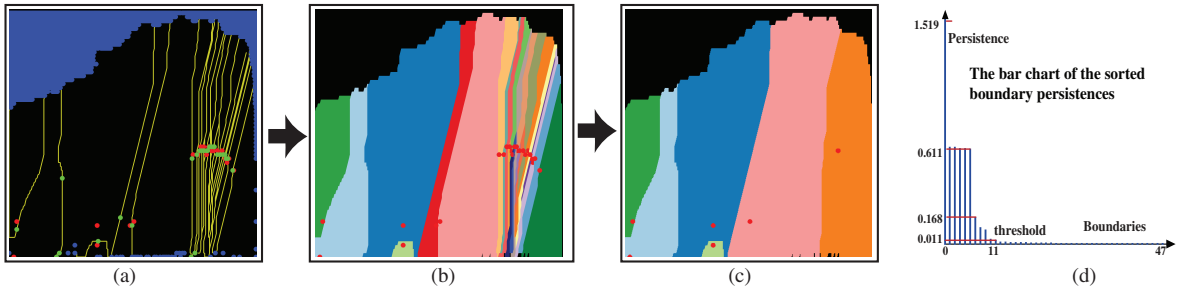


Figure 2: *The pipeline of our clustering algorithm illustrated with the 2D density plot of the Feet data set (value on the x-axis and value gradient magnitude on the y-axis). First, the critical points are identified where maxima are colored in red, saddles in green, and minima in blue in (a), and the valley lines are extracted shown in yellow. Due to the low resolution (*256×256*) of the density plot the obtained valley lines are not smooth. Then, the density plot is decomposed into several valley cells (b), where each cell has a maximum, and the bar chart of sorted boundary persistence (d) is provided. By analyzing the bar chart, the valley cells are simplified with a threshold and most of the small valley cells are merged in the results (c).*

and not the MS complex. No saddle-maximum relation has to be determined in the line tracing stage, and we simplified the special treatments needed to deal with the degenerated cases. The boundary persistence introduced in Section 3.2 can be considered as an analogue of the saddle-maximum pair persistence in [EHZ03] and is calculated later in the decomposition stage using an effective region growing technique [HP74]. Finally, we remove insignificant cells introduced by the noise with a modified persistence-based topological simplification process. Afterwards, each cluster formed by merging valley cells represents a potential feature in the volume data, as shown in Fig. 1. Fig. 2 illustrates the pipeline.

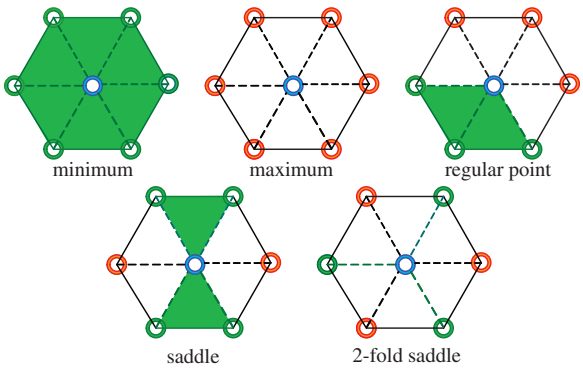


Figure 3: *The classification of a vertex in blue is determined by the relative value of the vertices in its star. The neighboring vertices are colored red if its value is larger and green if its value is smaller.*

3.1. Tracing Valley Lines

Following the definition in [EHZ03], the *star* of a vertex u consists of all simplices (vertices, edges and triangles) that share u as a vertex, and the *lower star* consists of the simplices that have u as the highest vertex. We define a *lower wedge* as a contiguous section of the lower star. As shown in Fig. 3, the lower wedge of u contains a number $k+1$ of wedges. The vertex u is defined as a *regular* if $k = 0$, and as a *k-fold saddle* if $k \geq 1$.

After identifying the critical points, we trace $k+1$ paths of steepest descending from every k-fold saddle. Starting from each saddle, each path follows a sequence of steepest descent edges. When the function is smooth, each path will terminate at a minimum ($\mathbf{m}_1$, $\mathbf{m}_2$, $\mathbf{m}$ in Fig. 4). For degenerated cases caused by piecewise linear discretization, we terminate the path when it hits: (a) another saddle ($\mathbf{S}_5$ in Fig. 4(a)) or (b) a previously traced path at a regular point ($\mathbf{J}$ in Fig. 4(b)) .

In case of Fig. 4(b), we consider vertex J as an *artificial* saddle because it is the maximum vertex on the path separating regions R_2 and R_3. Then, each boundary between adjacent cells has at least one saddle, which is a local maximum of the density function on that boundary.

3.2. Valley Cell Construction

In order to construct the valley cells, we use the region growing method from [HP74]. Starting from each maximum, this technique recursively grows the regions of neighboring triangles and stops at the valley lines or the triangles whose vertices are minima. During the growing procedure, we also construct the correspondence between maxima and saddles.

Saddles Update: After the region growing, each saddle has a connected maximum set. For correct merging of valley cells later, we have to adjust the correspondence between

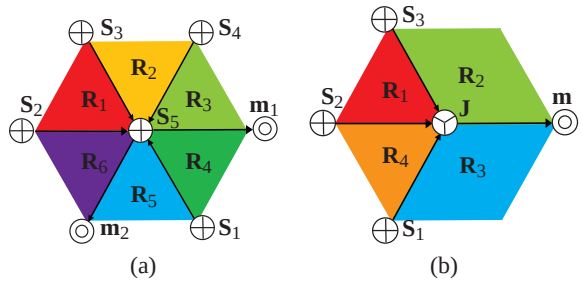


Figure 4: *Two degenerated cases when tracing valley lines: (a) paths ending at a saddle S_5; (b) paths ending at a junction **J**. To ensure that every boundary element is associated with at least one saddle, the junction **J** is regarded as an artificial saddle.*

saddles and maxima. We calculate its persistence by taking the smaller difference between the saddle and the maxima. As shown in Fig. 4, some saddles belong to more than one cell in the degenerated cases. In Fig. 4(a), saddle S_5 belongs to six cells. In Fig. 4(b), the artificial saddle J belongs to 4 cells. In other words, they have been associated to more than two maxima. To resolve this issue, we introduce saddle duplication. If the maximum set of the saddle **S** has k maxima, we duplicate $k-1$ saddle as shown in Fig 5. This is similar to the unfolding of the k-fold saddle [EHZ03]. However, we only duplicate the saddle without paths.

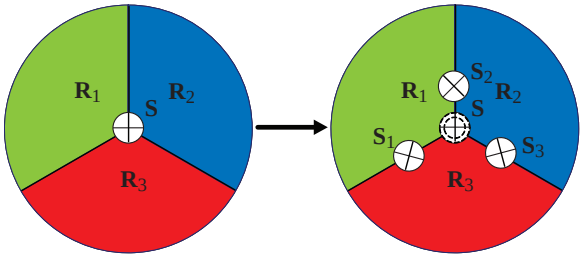


Figure 5: *Duplication of a saddle where several valley lines meet. (Left) Three valley lines cross at the saddle **S**; (Right) The original saddle **S** is replaced by three saddles S_1,S_2, and S_3.*

Boundary Persistence: Up to now, each saddle has a set of two maxima and each boundary has at least one saddle. To define the degree of separation of two adjacent cells, we introduce the boundary persistence, which is defined as the minimal persistence of its saddles. Although it is still defined by the persistence of saddles, it determines the lifetime of the boundary in the merging procedure.

3.3. Valley Cell Simplification

Due to the discretization of the density plots and the noise in the original data, there are a large number of small valley cells after the region growing procedure. Since the goal of volume exploration is to reveal semantic features in volume data, we introduce a boundary persistence-based cell simplification process to remove the noisy cells. Similar to the saddle maximum pair persistence [EHZ03], the boundary persistence measures the separability of the peaks on adjacent cells. Therefore, we use it to determine the order of cell merging because the cells caused by the noise usually have a small persistence.

We successively eliminate the boundary associated with the smallest persistence and merge the corresponding cells together until a given persistence threshold is reached. Consistency of the boundary persistence is maintained by updating the cells after every merging. Fig. 2(d) shows that a high percentage of persistence takes place at a very small value. Gyulassy et al. [GNP*06] observed that 10% of the maximal persistence is sufficient to detect and remove all insignificant features in their case. For density plot, our experience indicates that $0.1\% - 1.5\%$ is sufficient. As shown in Fig. 2(c), most of cells are merged with the threshold 0.5% of the maximal persistence. With this default threshold, our algorithm automatically generates meaningful clusters. To help the user to select a proper threshold, we additionally provide a persistence bar chart to reveal the noise level of the density function.

3.4. Comparison to Other Clustering Algorithms

In principle, our Morse complex-based clustering algorithm is a new variant of watershed transformation [Beu94] but in a hierarchical sense. For twice continuously differentiable functions, its decomposition result is similar to the watershed transformation. The major difference between them is the hierarchy construction algorithms. To determine the separability of boundaries between clusters, hierarchical watershed segmentation [Beu94] requires first flooding the cluster and then detecting whether overflows occur and mean shift involves a computationally expensive process [Sha03, ZZM10] to find saddles between clusters. Instead, such relation in our algorithm is intrinsically determined in the region growing procedure, which can be carried out quickly using flood-fill. In addition, we are the first who apply those techniques for the automatic design of transfer functions.

Concerning this, Maciejewski et al. [MWCE09] clearly show the need for an automatic transfer function design for volume data. Their approach clusters the feature space based on frequency binning of the 1D density histogram where regions with similar frequencies are regarded as one cluster. In more detail, their approach estimates a 1D histogram $h(f)$ from a density $f(x,y)$ given over a 2D feature space (x,y). The size of the bins of the 1D histogram h are finally used to cluster the feature space (x,y) and to build up the transfer function. Unfortunately, only the value of the density $f(x,y)$ in the feature space is considered but not the position (x,y),

although the same value of the density $f(x,y)$ which appears in different regions in the feature space (x,y) might belong to different structures within the volume data. Thus, our approach also considers the position within the feature space by directly considering topological information.

Fig. 6 illustrates the comparison of the clustering results produced by these two algorithms on a 1D density function. This density function consists of the two mixture Gaussian functions. It can be seen that our method separates the density function into two parts, while the binning method groups it into three bins where each bin has several same colored disjoint segments after setting the bin number of the density histogram to 3. A visible peak in the feature space corresponds to an interesting feature, which has been demonstrated in [KD98], while frequency binning lacks an intuitive explanation.

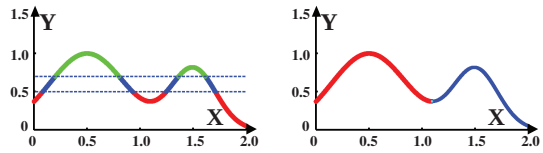


Figure 6: *Comparison of the frequency binning-based clustering method result and our method. (a) After specifying the bin size of the density histogram to 3, the binning method groups the density function into 9 segments; (b) our method separates it into two clusters.*

4. Transfer Function Design

With a certain persistence threshold, our clustering algorithm decomposes the 2D density plot into several valley cells. Each cell represents a potential structure within the volume data. Similar to [MWCE09, WCZ*11], we generate a transfer function by coloring these cells. By using a qualitative color scheme [Bre04], different cells can be well distinguished in the feature space. However, the user is often not able to identify more than five targets [Hea96]. Although we can automatically select the persistence threshold so that there are always no more than five clusters left after the simplification, we do not think this is appropriate because there is no guarantee that the number of significant features in a volume data is always below five. Instead, we set a safe persistence, just enough to remove the noise and construct a hierarchical tree to reveal the level of significance. From our experience, the number of remaining clusters are usually less than 10 using a threshold of 1% of the maximal persistence. Based on this tree, we introduce a new opacity transfer function generation method to maximize the differences between different features.

4.1. Hierarchy Construction

After the valley cell construction, each boundary between two adjacent cells is associated with a boundary persistence which determines the order of the cell merging operation. The tree can be constructed by merging the initial valley cells with a progressively increasing persistence threshold. If the boundary between two adjunct cells is removed, we build the parent and child relationships and update the maximum sets and boundary persistences of the other boundaries. In this tree, the root corresponds to the cell with the largest local maximum. Fig. 7 shows the tree constructed for the Feet data set.

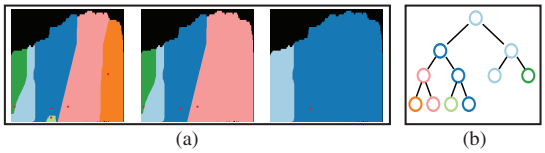


Figure 7: *The tree construction for the density plot of the Feet data set. (a)Three snapshots in the merging process. (b)The constructed tree.*

With this tree, the user can interactively merge and split features. When the user selects one cell, the cells which can be merged or split are highlighted. This makes the exploration of the feature's hierarchy convenient. By default, we do not introduce new color in merging and splitting. After merging two cells, the cell with the smaller maximum is colored the same as the one with a larger maximum, while the cell generated by splitting is colored by its original color before merging. Certainly, the user can also assign a new color.

4.2. Opacity Transfer Function Generation

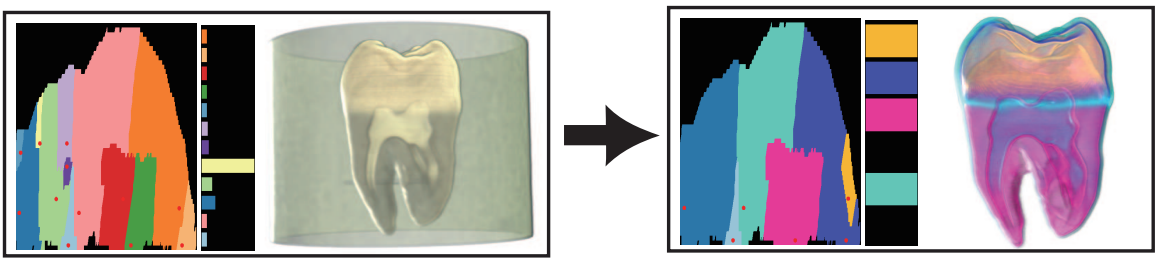
Cell Persistence When removing the boundary between two adjacent cells, these cells as well as their corresponding features are merged together. Thus, we define the persistence of each cell as the smallest persistence of its boundaries, which reflects the degree of separation of the corresponding features.

Initial Opacity Range Usually, the feature with a small size is occluded by other features. To help the user perceive all features, the opacity range for each node in the tree is computed by:

$$\alpha_i = \alpha_{\min} + \frac{W}{s_i \cdot p_{c_i}} g(d_i),$$

where $\alpha_{\min}$ is the minimum opacity specified by the user, s_i is the feature size , and p_{c_i} is the cell persistence of the node i. W is the normalization weight, which normalizes the $1/(s_i \cdot p_{c_i})$ into $[0,1]$, and $g(d_i)$ is a depth-based control function:

$$g(d_i) = \frac{d_i - d_{\min}}{d_{\max} - d_{\min}},$$



(a) (b)

Figure 8: *Exploring the CT Teeth data set using the density plot on the value (x-axis) and value gradient magnitude (y-axis) feature space. (a) The result with the automatically generated transfer function; (b) By merging several small cells and recoloring the left cells, the spatial relations among enamel, dentine and pulp are clearly shown.*

	Ours	BTF [MWCE09]
Parameter plot	density plot	density plot
Feature specification	peak	binning
Number of features	automatic	specified
Merging of features	persistence	no guideline
Interactive refinement	hierarchy	arbitrary

Table 1: *Properties of our method compared to BTF [MWCE09]*

where d_i is the depth of node i in the merge tree and $d_{\min}$ and $d_{\max}$ are the minimum and maximum depth. The opacity bar charts in Fig. 1(b) show the automatically generated opacity for each feature where skin and air are set to very low opacities. We do not expect that the initial opacities always produce desired rendering results. Thus, we allow the user to manually adjust the opacity value for each cell.

Gaussian Transfer Function With the automatically obtained opacity ranges, we use the Gaussian transfer function (GTF) which can facilitate high quality pre-integrated volume rendering [KPI*03] to generate actual opacity values:

$$\alpha(x) = \alpha_{\max} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)},$$

where $\alpha_{\max}$ is the determined opacity range and Σ^{-1} is density covariance in the corresponding valley cell. By default, μ is the position of the maximum. We also provide the user another option to set μ as the center of the cluster. To avoid that the Gaussian function makes contribution to other features, we compute a bounding box for its corresponding valley cell. The opacity of the region outside the bounding box is zero.

4.3. Comparison to the State of the Art

In this section, we qualitatively compare this method with the approach in [MWCE09]. We refer to this approach as BTF (binning-based transfer functions). Both methods are based on density plots. However, there are four main differences. First, the clustering principles are different, as discussed in Section 3.4. Second, BTF requires the user to specify the number of histogram bins, while our method can automatically determine the number of features. Third, BTF does not provide any guideline for parameter space simplification, our method uses cell persistence. Last, their initial classification result can be arbitrarily merged and split, while our hierarchy representation can help the user to effectively explore features of interest. Table 1 summarizes the comparison between these two methods.

5. Experiment and Discussion

We have implemented and tested our approach on a PC with an Intel Core 2 Duo E6320 1.8 GHZ CPU, 2.0 GB RAM, and an NVIDIA Geforce GTX 260 video card (256 MB video memory) using the Cg Language. For the density plots provided by CSPs, we use the code of [BW09]. By setting the error threshold to 50 and the resolution to 256^2, the density plots can be generated in less than 1 hour. After obtaining the density plots, our clustering algorithm finishes the volume classification in 10 seconds and results in an interactive volume exploration. Note that the rendering performance gradually decreases with the number of valley cells, because the GTF-based volume rendering is directly evaluated on the GPU. In general, the number of valley cells is less than 10, which results in a volume rendering frame rate of approximately 25 fps.

5.1. Medical Data

Fig. 8 shows an example that classification of the value and value gradient magnitude feature space in the Teeth data set. By simplifying this density plot with persistence threshold 1.0% of the maximal persistence, our method automatically clusters this density plot into twelve parts and obtains an initial classification result, as shown in Fig. 8(a), where enamel

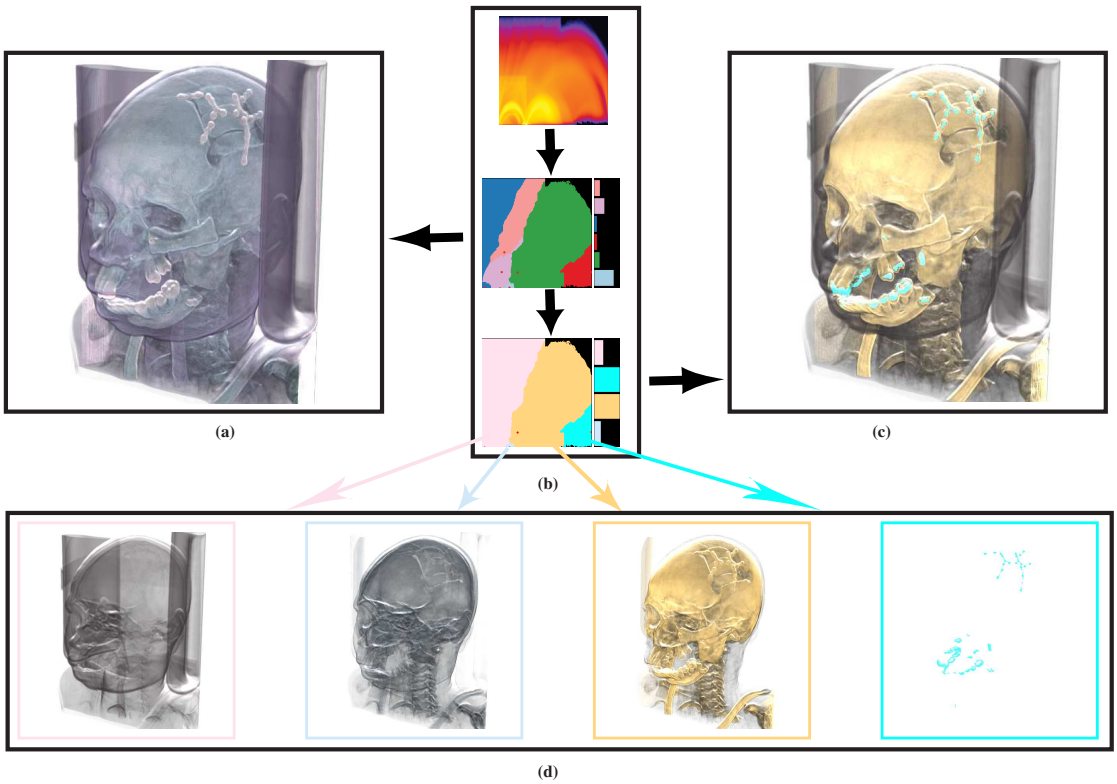


Figure 9: *Exploring the CT facial deformity data set using the density plot on the value and value gradient magnitude feature space (top one in (b)). The automatically generated transfer function (middle one in (b)) and its corresponding rendering result (a). After merging the peach, pale green and blue cells and recoloring and adjusting their opacities, a new transfer function (bottom one in (b)) and its corresponding rendering result (c) is obtained, and the corresponding features of the four clusters in the new transfer function are shown in (d). The bar charts in (b) represent the opacities of the corresponding features.*

and pulp are revealed but not clearly due to some noise. Through merging of some cells located at the region of low densities and adjusting their colors and opacities, we obtain a better result where the spatial relation among enamel, dentine and pulp are clearly revealed, as shown in Fig. 8(c).

In the above example, each cell in the density plot corresponds to a meaningful feature. While our clustering algorithm is based on topology-based peak analysis, it cannot guarantee that the user is interested in all extracted features, especially some features produced by noise. Thus, merging the uninteresting features not only facilitates the exploration of feature space but also improves the classification quality. Fig. 9 shows the effectiveness of our exploration scheme in surgery repair on a CT facial deformity data set ($512 \times 512 \times 361$). The data set was acquired from a facial deformity patient where the regions located near the upper jaw and the top of the skull are damaged. The damaged regions must be identified before the surgical planning procedure. We obtained the initial result (Fig. 9(a)) after decomposing the density plot of value and value gradient magnitude feature space into 7 cells (the middle one in Fig. 9(b)). A lesion and a damaged region in cyan where some teeth are absent are seen, but the relationship between these regions to the skull and the face is not clear. By examining each cell, we found that the corresponding features of peach, pale green and blue cells are skin and then merged them into one cell (the bottom one in Fig. 9(b)). After recoloring and adjusting their opacities, a better result (Fig. 9(c)) was achieved, where the lesion in cyan and the damaged region with some teeth absent are clearly illustrated. Fig. 9(c) shows the corresponding feature of each cell. We can see that skin, bone and lesion are clearly differentiated.

5.2. Simulation Data

To demonstrate the effectiveness of our approach on simulation data sets, an experiment was conducted on the Supernova data set ($432 \times 432 \times 432$). In this data, scientists are interested in the turbulent structures which are near the core of the supernova but occluded by the outer layer. To reveal these structures, we apply our automating transfer function

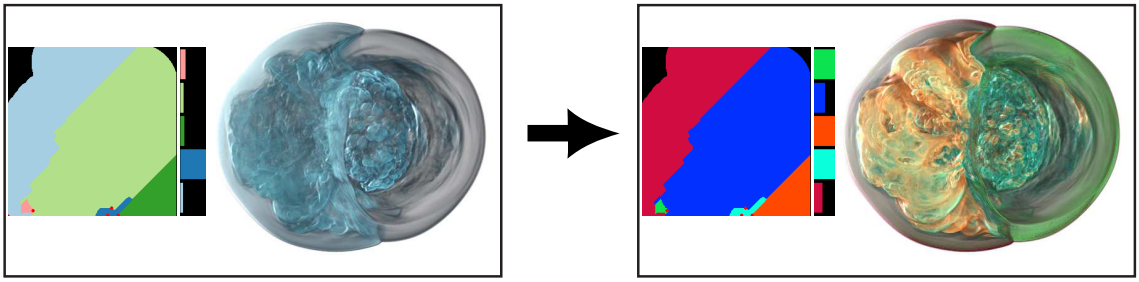


Figure 10: *Exploring the Turbulent dataset by decomposing the density plot on the value and value gradient magnitude feature space. The result (left) with the automatically generated transfer function. After recoloring and adjusting opacities, the relationship between inner and outer layer structures is clearly shown.*

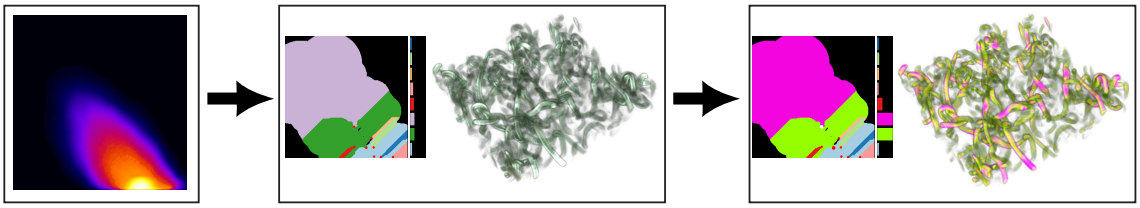


Figure 11: *Exploring the Turbulent dataset by decomposing the density plot (a) on the pressure (x axis) and vorticity (y axis) feature space. (b) The result with the automatically generated transfer function; (c) By adjusting the opacities of the plum cell and green cell and recoloring them, the kinking and tangling vortex tubes become clearly visible with a large contrast.*

generation method to decompose the density plot of scalar value and value gradient magnitude feature space and obtain five clusters, as shown in the left of Fig. 10. From the initial rendering result, we can see that the cell in sky blue corresponds to the inner turbulent structure. Unfortunately, the opacities of other features are too small to reveal their structures. After examining each feature, we find two different outer layers for two cores and another inner layer in the left core. By recoloring and adjusting their opacities, the spatial relationship between the inner turbulent structures and the out layers is well illustrated.

For the extraction of features from some scientific data sets, the domain scientist usually has specific knowledge about the feature space construction. For example, the vortices exist in regions with large vorticity and small pressure in some turbulent data sets. To demonstrate the effectiveness of our method in classification of knowledge-aware feature space, we conducted an experiment on the Turbulent data set produced by a 128-cubed simulation of a compressible and turbulent slip surface. By decomposing the density plot (Fig. 11(a)) with pressure on the x-axis and vorticity on the y-axis, we obtained a classification result (Fig. 11(b)), where regions with large vorticities and small pressures are decomposed into two large cells. After recoloring and adjusting their opacities, a desired rendering result (Fig. 11(c)) is obtained, which clearly reveals the kinking and tangling vortex tubes.

6. Conclusion

This paper introduces a new transfer function design with a unique capability to capture the data characteristics, while still affording favorable user interactivity. This flexibility is especially helpful for inexperienced users because it can automatically provide a suggestive volume classification by decomposing density plots. By allowing the user to interactively explore the pre-computed clusters in the feature space, he/she gets an initial understanding of the underlying data set. With a feature hierarchy, the uninterested features can be merged or removed to improve the visualization quality. This scheme has proven to be effective and efficient over various types of volumetric data.

In the future, we would like to extend our approach to decompose 3D volume by browsing the idea from the 3D morse complex [EHNP03]. Likewise, analysis of the density volume can well classify the time-varying data set.

7. Acknowledgment

This paper was supported by the National Natural Science Foundation of China (No. 60902104), Natural Science Foundation of Guangdong Province (No. S2011040000433), the the National Natural Science Foundation of China (No. 91130019). The datasets are courtesy of General Electric, the Terascale Supernova Initiative, and the OsiriX Foundation.

References

[Beu94] BEUCHER S.: Watershed, hierarchical segmentation and waterfall algorithm. *Mathematical morphology and its applications to image processing* (1994), 69–76. 5

[BHEP04] BREMER P., HAMANN B., EDELSBRUNNER H., PASCUCCI V.: A topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics 10*, 4 (2004), 385–396. 3

[Bre04] BREWER C.: *Designing better Maps: A Guide for GIS users*. Environmental Systems Research, 2004. 6

[BW08] BACHTHALER S., WEISKOPF D.: Continuous scatterplots. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1428–1435. 2

[BW09] BACHTHALER S., WEISKOPF D.: Efficient and adaptive rendering of 2-D continuous scatterplots. *Computer Graphics Forum 28*, 3 (2009), 743–750. 7

[CM08] CORREA C., MA K.: Size-based Transfer Functions: A New Volume Exploration Technique. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1380–1387. 3

[CM09] CORREA C., MA K.: The Occlusion Spectrum for Volume Visualization and Classification. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1465–1472. 3

[EHNP03] EDELSBRUNNER H., HARER J., NATARAJAN V., PASCUCCI V.: Morse-Smale complexes for piecewise linear 3-manifolds. In *Proceedings of the nineteenth annual symposium on Computational geometry* (2003), pp. 361–370. 9

[EHZ03] EDELSBRUNNER H., HARER J., ZOMORODIAN A.: Hierarchical MorseąłSmale Complexes for Piecewise Linear 2-Manifolds. *Discrete and Computational Geometry 30*, 1 (2003), 87–107. 2, 3, 4, 5

[FAT99] FUJISHIRO I., AZUMA T., TAKESHIMA Y.: Automating transfer function design for comprehensible volume rendering based on 3D field topology analysis. In *Proceedings of IEEE Visualization'99* (1999), pp. 467–563. 3

[FKLT10] FENG D., KWOCK L., LEE Y., TAYLOR R.: Matching Visual Saliency to Confidence in Plots of Uncertain Data. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 980–989. 2

[GNP*06] GYULASSY A., NATARAJAN V., PASCUCCI V., BREMER P., HAMANN B.: A topological approach to simplification of three-dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics 12*, 4 (2006), 474–484. 2, 3, 5

[Hea96] HEALEY C.: Choosing effective colours for data visualization. In *Proceedings of IEEE Visualization'96.* (1996), pp. 263–270. 2, 6

[HP74] HOROWITZ S., PAVLIDIS T.: Picture segmentation by a directed split-and-merge procedure. In *Proc. 2nd Int. Joint Conf. Pattern Recognition* (1974), pp. 424–433. 4

[KD98] KINDLMANN G., DURKIN J.: Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of IEEE Symposium on Volume Visualization'98* (1998), pp. 79–86. 6

[KPI*03] KNISS J., PREMOZE S., IKITS M., LEFOHN A., HANSEN C., PRAUN E.: Gaussian transfer functions for multi-field volume visualization. In *Proceedings of IEEE Visualization'03* (2003), pp. 65–72. 7

[KWTM03] KINDLMANN G., WHITAKER R., TASDIZEN T., MÖLLER T.: Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications. In *Proceedings of IEEE Visualization'03* (2003), pp. 513–520. 3

[Lev88] LEVOY M.: Display of surfaces from volume data. *IEEE Computer Graphics and Applications 8*, 3 (1988), 29–37. 1, 3

[MWCE09] MACIEJEWSKI R., WU I., CHEN W., EBERT D.: Structuring Feature Space: A Non-Parametric Method for Volumetric Transfer Function Generation. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1473–1480. 1, 2, 3, 5, 6, 7

[Par96] PARKER J.: *Algorithms for image processing and computer vision*. John Wiley & Sons, Inc. New York, NY, USA, 1996. 2

[RBS05] ROETTGER S., BAUER M., STAMMINGER M.: Spatialized transfer functions. In *Proceedings of IEEE/Eurographics Symposium on Visualization'05* (2005), pp. 271–278. 1

[Sha03] SHAMIR A.: Feature-space analysis of unstructured meshes. In *Proceedings of the IEEE Visualization 2003* (2003), pp. 185–192. 5

[Sma61] SMALE S.: On gradient dynamical systems. *Annals of Mathematics 74*, 1 (1961), 199–206. 2, 3

[ŠVG06] ŠEREDA P., VILANOVA A., GERRITSEN F.: Automating transfer function design for volume rendering using hierarchical clustering of material boundaries. In *Eurographics/IEEE VGTC Symposium on Visualization* (2006), pp. 243–250. 1, 3

[TM04] TZENG F., MA K.: A cluster-space visual interface for arbitrary dimensional classification of volume data. In *Proceedings of IEEE/Eurographics Symposium on Visualization'04* (2004), pp. 17–24. 1, 3

[TTFN05] TAKESHIMA Y., TAKAHASHI S., FUJISHIRO I., NIELSON G.: Introducing topological attributes for objective-based visualization of simulated datasets. In *Proceedings of the Volume Graphics'05* (2005), pp. 137–236. 3

[WCZ*11] WANG Y., CHEN W., ZHANG J., DONG T., SHAN G., CHI X.: Efficient volume exploration using the gaussian mixture model. *IEEE Transactions on Visualization and Computer Graphics 17*, 11 (2011), 1560–1573. 1, 3, 6

[WDC*07] WEBER G., DILLARD S., CARR H., PASCUCCI V., HAMANN B.: Topology-controlled volume rendering. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (2007), 330–341. 3

[WG09] WEINKAUF T., GÜNTHER D.: Separatrix persistence: Extraction of salient edges on surfaces using topological methods. *Computer Graphics Forum 28*, 5 (2009), 1519–1528. 3

[ZT09] ZHOU J., TAKATSUKA M.: Automatic Transfer Function Generation Using Contour Tree Controlled Residue Flow Model and Color Harmonics. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1481–1488. 3

[ZZM10] ZHOU F., ZHAO Y., MA K.: Parallel mean shift for interactive volume segmentation. In *Proceedings of MICCAI Workshop on Machine Learning in Medical Imaging* (2010), Springer, pp. 67–75. 5