Automatic Transfer Function Generation Using Contour Tree Controlled Residue Flow Model and Color Harmonics

Jianlong Zhou, Student Member, IEEE, and Masahiro Takatsuka, Member, IEEE

Abstract—Transfer functions facilitate the volumetric data visualization by assigning optical properties to various data features and scalar values. Automation of transfer function specifications still remains a challenge in volume rendering. This paper presents an approach for automating transfer function generations by utilizing topological attributes derived from the contour tree of a volume. The contour tree acts as a visual index to volume segments, and captures associated topological attributes involved in volumetric data. A residue flow model based on Darcy's Law is employed to control distributions of opacity between branches of the contour tree. Topological attributes are also used to control color selection in a perceptual color space and create harmonic color transfer functions. The generated transfer functions can depict inclusion relationship between structures and maximize opacity and color differences between them. The proposed approach allows efficient automation of transfer function generation on the data to be carried out based on controlling of opacity residue flow rate instead of complex low-level transfer function parameter adjustments. Experiments on various data sets demonstrate the practical use of our approach in transfer function generations.

Index Terms—Volume Rendering, Transfer Function, Contour Tree, Residue Flow, Harmonic Color.

1 INTRODUCTION

The ultimate goal of volume visualization is to provide useful insights into volumetric data. Direct volume rendering is one of effective and flexible visualization methods for three-dimensional volumetric data. Despite the proliferation of volume rendering and manipulation techniques, the key to comprehensible volume rendering still lies in the design of effective transfer functions, which map scalar values to specific colors and opacities. Transfer functions assume that scalar values map directly to physical properties such as tissue types. Thus, they are crucial in the understanding of the overall volumetric data and individual features contained within the volume space. Various approaches have been developed to automate or ease specification of transfer functions.

Conventional approaches often use boundary information to automate transfer function specifications [11, 12]. Despite various data features (e.g. derivatives, curvatures, texture, size) being used to define transfer functions, most of current volume rendering methods highly depend on manual specification (at least in some degree) of transfer functions to reveal various structures in a volume. This process is time-consuming and comes short in repeatable results. In addition, volumetric data often contains nested inner structures. This commonly seen inclusion relationship heavily affects understanding of volumetric data. Conventional transfer functions cannot automatically fully clarify such inner structures. Thus, a systematic and automatic scheme to specify transfer functions while revealing inclusion relationships is highly desirable to greatly facilitate the volume rendering process.

The contour tree is a data structure representing nesting relationships of connected components of contours [3]. In addition to provide topological features of a volume, the contour tree is also used in transfer function generations [8, 24, 29]. Topology controlled transfer function generations are reviewed in Section 2. The underlying weakness of these methods is that transfer functions are still specified manually through user interactions. They do not consider how to use the topology to automate transfer function generations. The principal technical challenges for these problems include how to generate transfer functions that are used to get a wide spread of dissimilar output renderings and to reveal inclusion relationship between subregions. Users also

 Authors are with School of Information Technologies, The University of Sydney, NSW 2006, Australia, and National ICT Australia (NICTA).
 E-mail: {zhou|masa}@it.usyd.edu.au.

Manuscript received 31 March 2009; accepted 27 July 2009; posted online 11 October 2009; mailed on 5 October 2009. For information on obtaining reprints of this article, please send email to: tvcg@computer.org. need to decide what measures are used to control transfer function differences between subregions based on the contour tree.

Moreover, any of different subregions in volumetric data may bear interesting intensity variations which need to be visually represented in a faithful manner. Color transfer functions allow aesthetically attracting rendering in order to improve data analysis efficiency. Conventional color transfer functions are parameterized by values that segment the data into different subranges, which are arbitrarily assigned with different colors based on experiences and personal preferences [13, 14]. Thus color is used to only identify subranges of the data, and not to convey any quantitative relations among the data. One popular design aspect of color in graphics and image is color harmony [28]. Wang et al. [27] presents a harmonic color design scheme for illustrative visualization. The scheme uses manually specified parameters such as importance factors to control color components (hue, vividness, lightness) selection for segmented data rendering. The principal weakness of this scheme is that the manually specified parameters have no direct relations with features derived from the data set itself. Vividness and lightness are still selected arbitrarily based on users' preferences in some degree. Thus, an automatic scheme to specify color transfer functions by incorporating data features (e.g. topological features) is highly desirable, and expected to greatly improve the visual aesthetic attraction of rendering.

This paper presents an approach for automating transfer function generations by utilizing topological attributes derived from the contour tree of a volume. The contour tree acts as a visual index to volume segments, and captures associated global topological attributes involved in volumetric data in the pipeline. The objective of this work is to deliver a new transfer function generation paradigm based on the contour tree. The generated transfer functions can depict inclusion relationship between structures and maximize differences between them. The advantage of the proposed approach is that it allows more efficient automation of transfer function generations, and exploration on the data to be carried out based on controlling of opacity residue flow rate instead of complicated low-level transfer function parameter adjustments. The contributions of the paper are follows:

- A residue flow model based on Darcy's Law is proposed to control opacity residue flow between branches in the contour tree.
- Based on the residue flow model, we design a comprehensive framework dedicated to automate transfer function generations controlled by topological attributes from the contour tree.
- Contour tree controlled harmonic color transfer function is proposed to get visual aesthetic attraction of rendering.

This paper is organized as follows. We first introduce the previous Published by the IEEE Computer Society

work on transfer functions in Section 2. We also briefly survey contour trees in Section 3. An overview of the proposed framework is described in Section 4. A residue flow model is presented in Section 5 to control opacity distribution in the contour tree. The residue flow model is used to automate the opacity transfer function generation in Section 6. An automatic color transfer function generation scheme is covered in Section 7. Section 8 describes experimental results and discussions. Finally, conclusions are drawn in Section 9.

2 RELATED WORK

Research on transfer functions is focused on making the parameter space of transfer functions easier to explore. Many methods have been proposed to that end ranging from 1D and multi-dimensional transfer function, semantic transfer function, to incorporate other data features in transfer functions. Typical opacity transfer function and color transfer function approaches are reviewed in this section.

2.1 Early Transfer Function Approaches

Early volume rendering approaches often use 1D transfer functions to explore volume data [16]. Fang et al. [7] defined a transfer function as a sequence of 3D image processing procedures, and allows users to adjust a set of qualitative and descriptive parameters to achieve their subjective visualization goals. Design galleries [17] generates all possible transfer functions simultaneously, each representing a different configuration of the transfer function. The satisfied transfer function is then selected from these representatives and optimized. Contour spectrum [2] visually summarizes the space of isosurfaces in terms of metrics like surface area and mean gradient magnitude, thereby guiding the choice of isovalue for isosurfacing, and also providing information useful for transfer function generations. These approaches do not provide any mechanism to automate transfer function generations and still apply transfer functions globally instead of locally to structures.

2.2 Multi-Dimensional Transfer Functions

2D transfer functions were firstly introduced by Levoy [16]. Kindlmann and Durkin [11] proposed a semi-automatic method both for 1D and 2D transfer functions. It uses a data structure named histogram volume to capture the relationship between data values and boundary representations. Kniss et al. [13, 14] designed an interface to construct multi-dimensional transfer functions interactively. This interface adds derivatives information as parameters to transfer functions, but continues to apply same transfer functions everywhere in the data. Kindlmann et al. [12] added the curvature of the function as a parameter into multi-dimensional transfer functions. This approach still lacks any spatial locality. Huang and Ma [9], and Tzeng et al. [26] generated 2D transfer functions by using results of partial region growing from a point selected in volume space. Automation of transfer function specifications are still not solved, but more parameters to be modulated by users are introduced in multi-dimensional transfer functions.

2.3 Semantic Transfer Functions

In order to ease the specification of transfer functions with complex parameter space, Rezk-Salama et al. [20] introduced an abstract semantic level for parametric models of transfer functions. The proposed semantic model is defined based on a set of reference data, a list of the relevant structures contained in the data, and transfer function templates for the reference data. Rautek et al. [19] presented a semantic layer mapping from volumetric attributes to multiple illustrative visual styles in volume rendering. The basic low level of semantics is still the various conventional transfer function specification methods as reviewed above. They still require excessive user interactions to create ideal renderings.

2.4 Topology Controlled Transfer Functions

Introducing topological attributes into the transfer function specification allows volume rendering to capture global characteristics of the data set while locating regions of particular interest. Fujishiro et al. [8] introduced topological attributes in transfer function generations. Takahashi et al. [23, 24] and Takeshima et al. [25] used topological attributes derived from the contour tree to define transfer functions. However, they applied transfer functions to a data set according to fixed topological indices (e.g. depth) and did not explore transfer function automation deeply. Conventional transfer function approaches cannot distinguish between distinct features that share the same scalar value. To overcome this problem, Weber et al. [29] used the contour tree to index various subregions of a volume and specify transfer functions locally for individual subregions. However, as there are often many branches in a contour tree, it is impractical and time-consuming to define transfer functions for each subregion corresponding to a branch one-by-one manually. The inclusion relationship is also not considered in [29].

2.5 Color Transfer Functions

Color transfer functions map scalar densities to colors in order to label different objects and create aesthetic attraction in volume rendering. Color transfer functions have in most cases been guided by personal preferences or even just by random assignments [28]. Furthermore, color harmony is one popular design aspect in terms of aesthetics. Cohen-Or et al. [5] introduced a framework of automated image color harmonization. Wang et al. [27, 28] extended this color harmonization to be used in volume visualization. They presented how color harmonization is used to semi-automate color definitions in volume rendering. Our work extends the color selection algorithm in [27] by incorporating topological features.

In addition, recent work on transfer functions introduced other features of volume data into the transfer function pipeline. For example, Selver and Güzelis [22] used a volume histogram stack to do semiautomatic transfer function initialization. Neural networks are used in this process. Correa and Ma [6] mapped a size of features to color and opacity. In our work, we consider affects not only of size (volume) of structures, but also of hypervolume and persistence of structures on transfer functions. Persistence, volume, and hypervolume are derived from the contour tree and more meaningful in representing importance of structures in transfer functions.

3 CONTOUR TREE REVISITED

Topology has been an important tool for analyzing scalar data and flow fields in visualization. The contour tree is one of data structures used to explicitly store topological features. This section reviews basics of the contour tree, and this forms the basis of contour tree controlled transfer function generation in this paper.

3.1 Definition of Contour Trees

Consider a continuous scalar field \mathbb{F} defined on a domain \mathbb{R}^d , $f : \mathbb{R}^d \to \mathbb{R}$. \mathbb{R}^d is assumed to be a simplicial complex. For a point inside a simplex, its function value is a linear interpolation of the values on the vertices. The *functional range* of the field \mathbb{F} is the interval between the minimum and maximum values of the function f, $[f_{min}, f_{max}]$. For a scalar value $h \in [f_{min}, f_{max}]$, the *level set* of the field \mathbb{F} at the value h is the subset $L(h) = \{(x) | f(x) = h\}$. While h scans monotonically through the entire range of \mathbb{F} from f_{min} to f_{max} , the topology of the level set of $L(h) = \{(x) | f(x) = h\}$, contours appear at local minima of f, join or split at saddles, and disappear at local maxima of f. If each contour is represented as a *node*, the evolution of the level set forms a tree called *contour tree*. It represents the nesting relationships of connected components of contours [4, 29].

Typically, the contour tree is represented as a list of *nodes* and a list of *arcs*, where each arc is defined as a node pair. The nodes of the contour tree correspond to critical points of a scalar field and are therefore associated with their critical values. Pascucci et al. [18] used an alternative *branch decomposition*, where a *branch* is defined as a monotone path in the graph traversing a sequence of nodes with non-decreasing (or non-increasing) value of the scalar field.

3.2 Contour Tree Simplification

The contour tree is vulnerable to noise and artifacts in the input data. Many real-world data sets produce unmanageably large contour trees because of noise and artifacts from a data acquisition process. This makes it difficult to recognize edges that correspond to objects of interest, and results in the contour tree be impractical in data analysis and visualization. The contour tree simplification would remove the noise and artifacts, while making the size of tree small enough for the user interaction and maintaining essential structure of the data.

Various approaches are proposed to simplify the contour tree. For example, Carr et al. [4] simplified the contour tree based on several metrics that are used for ranking the "importance" of an arc. The geometric measures used in the contour tree simplification include persistence, volume, and hypervolume. Persistence is equal to the absolute difference in scalar values of two critical points. Topological persistence is used to look at homological features that persist along a nested sequence of topological spaces. It refers to how long a topological feature last until it changes. Volume is the voxel count of the region enclosed by the isosurface. It refers to the size of objects. Hypervolume is the integral of the scalar field over the enclosed region. Pascucci et al. [18] used a data structure named branch decomposition for multi-resolution hierarchical representation of the contour tree.

3.3 Contour Tree Indexed Subregions

Since the concept of flexible isosurface [3, 4] supports independent manipulation of single contours, the contour tree can be used as a visual index to segment data set into different zones/subregions. Weber et al. [29] extended this idea in volume rendering and proposed a volume rendering framework which classifies volume data based on the contour tree and assigns unique transfer function to each subvolume corresponding to a branch of the contour tree. Our work uses the contour tree not only to index various subregions and define unique transfer functions to them, but also to control transfer function generations based on topological features derived from the contour tree.

4 FRAMEWORK OVERVIEW

Our proposed pipeline for automatic transfer function generations consists of several processes. An overview of the framework is shown in Fig. 1. Given a volume data set, the contour tree is created and simplified. The contour tree is used as a visual index of various subregions/structures of the data set, and different topological attributes are derived from the contour tree. These topological attributes are then used to control transfer function generations. Based on the contour tree, an opacity residue flow model is set up. The opacity residue flow model treats opacity transfer function generation as a kind of liquid (e.g. water) flow process between branches of the contour tree: opacity is treated as water, and topological attributes are used to control opacity flow between branches. Interfaces are provided to allow users to control the opacity flow rate and the shape of local transfer function in a branch, and thus decide the final transfer function. Meanwhile, topological attributes are also used to generate harmonic colors for color transfer functions. The generated opacity and color transfer functions are then used to render the data set.



Fig. 1. Flow chart showing the pipeline for automatic transfer function generations.

In this framework, users only need to determine the opacity flow rate, and thus generate transfer functions automatically. The approach generates transfer functions more easily and simply than traditional approaches which require complex interactions on various parameters. The opacity flow rate and local transfer function shape are used in the pipeline to keep topological relationships represented in the contour tree in the rendering. This paper focuses processes of how the contour tree controls transfer function generations in the pipeline. The details of each process will be covered in the later sections.

5 RESIDUE FLOW MODEL

In this paper, we model the distribution of opacity between branches in the contour tree as a multi-stream water flow process. When the water passes through a branch, some of it is re-directed by the branch and residues are delivered to child branches at the next depth level. This section presents Darcy's Law as a basic theory to get residues, and gives an overview of residue flowing between branches.

5.1 Darcy's Law



Fig. 2. Definition and direction for Darcy's law in one dimensional case.

Darcy's Law [1, 15] is a generalized relationship for fluid flow in porous medium. Fig. 2 shows the definition and direction for Darcy's law in one dimensional case, where the light red regions represent the fluid and blue arrows represent the flow direction of the fluid. Darcy's Law demonstrates that the volumetric flow rate Q of the fluid, through the porous medium, is proportional to both the head loss (i.e., fluid height difference) $(h_{out} - h_{in})$ between the ends of the cylinder, the cross-sectional area A of the medium, and inversely proportional to the packed length L of the cylinder:

$$Q = -KA\frac{\Delta h}{L},\tag{1}$$

where the minus sign on the right hand reflects that the hydraulic head always decreases in the direction of flow. We use absolute value in this paper. The constant K of proportionality, is referred to as the hydraulic conductivity and is a function of both the porous medium's permeability and the particular fluid. This can be seen by writing the hydraulic conductivity in terms of the permeability k of the medium:

$$K = k \frac{\rho g}{\mu},\tag{2}$$

where ρ and μ are the fluid's density and viscosity, and g is the acceleration due to gravity. [10] summarizes average hydraulic conductivity K for general materials. This paper uses sand as the porous medium, and the value of K for water flowing through sand is 300 [10]. The choice of materials does not affect rendering results except for the use of different flow rates in the rendering pipeline.

5.2 Residue Flow Model

We model the distribution of opacity between various branches of the contour tree as a water flow process in the tree structure. Opacity band on each branch is defined as the opacity range size, i.e., the difference between maximum opacity and minimum opacity applied to that branch. It is modeled as the flowing water. As shown in Fig. 3, we imagine that the contour tree is placed upsidedown with the root upside and leaves downside. At the same time, the water is poured at the root. The branch is modeled as a pipe structure with porous medium inside. Every branch in the tree absorbs some water passing through it



Fig. 3. Water is poured at the root of the tree and flows throughout the tree. Part of water is absorbed by each branch and residue is delivered to the next depth level. The depth level is encoded with colors.

and delivers residue to its children. The water absorbed by each branch is modeled as the opacity band allocated to each branch.

The Darcy's Law is adapted to control the opacity distribution between branches in the contour tree: For each branch, the packed length L of the cylinder in the Darcy's Law is modeled as the persistence p_i of the branch, i.e., $L = p_i$; the cross-sectional area A of the medium in the Darcy's Law is modeled as the number n_i^c of child branches, i.e., $A = n_i^c$; the volumetric flow rate Q of the fluid is modeled as the opacity flow rate in the branch; and Δh is modeled as the opacity residue factor in the branch. So given the opacity flow rate Q, the opacity residue factor Δh of each branch can be obtained from Equation 1. The opacity residue factor is used to modulate opacity band and get opacity residue for the next depth level.

In the residue flow model, the water begins to flow at the root of the contour tree as shown in Fig. 3. Some of the flowing water is absorbed by branches, and there is the residue that is not absorbed by branches on each depth level. We assume that the water initially absorbed by branches at each depth level is equal. This means that the water is initially evenly distributed between depth levels. After this, the Darcy's Law is applied to each parent branch sequently from the root branch in order to get residues and then deliver them to child branches at the next depth level. The residue flows down to the next depth level, and combined with the initially evenly distributed water together. This combined water is then partly absorbed by the branch and delivered the residue to the next depth level further. This process is continued until reaching leave branches.

Based on this model, the opacity band on each branch is allocated using following procedure: Initially, the opacity range used to render the overall data set is divided into equal-size bands based on the maximum depth value. This allows the opacity band be evenly distributed between each depth level. Then from the root of the contour tree, initial opacity band on each depth level is further processed in order to distribute it among siblings and deliver the opacity residue to the next depth level. The residue of opacity band on each depth level sinks down to the next depth level and contributes for the opacity band computation of that level. This process is illustrated in Algorithm 1.

Algorithm 1 shows that the residue is generated on the parent depth level and flows to the next depth level. The residue is also distributed between siblings on each depth level. The details of each step in Algorithm 1 are presented in later sections.

6 OPACITY TRANSFER FUNCTION GENERATION BASED ON RESIDUE FLOW MODEL

Objectives of opacity transfer function (OTF) in this paper include: maximize differences between various objects inside volume data, and represent corresponding inclusion relationship between objects at the same time. The branch decomposition is used to represent the topology of volume data in this paper. This section first presents how the contour tree is used to represent inclusion relationship of objects inside volume data. Then the residue flow model is used to control the flow of opacity band in the contour tree in order to specify OTF.

Algorithm 1: Residue flow in the contour tree.
Input: Contour tree
Output: Opacity distribution in the contour tree
① Get initial opacity band α_d for each depth level;
for each depth level i do
⁽²⁾ Get opacity residue factor Δh_i ;
⁽³⁾ Get basic opacity band α_i of the current depth level based
on residue $\Delta \alpha_{i-1}$ from parents;
for each child j do
(4) Distribute α_i based on scalar value s_{ij} as well as
importance measures;
(5) Get opacity band α_{ij} of the current branch b_{ij} ;
end
(6) Get residue $\Delta \alpha_i$ of the current depth level;
end

6.1 Representation of Inclusion Relationship

In 3D space, structures of interest are often included by other structures because of their spatial positions. The inclusion relationship refers to the situation of objects included by others in 3D space, which creates inner objects and outer objects respectively. Takahashi et al. [24] called this as isosurface embedding. Typical examples are often seen in medical or mechanical structures. Providing clear insight into inner structures involved in a volume data set has been a challenging task in the field of volume rendering.

Takahashi et al. [24] and Takeshima et al. [25] introduced inclusion level to analyze inclusion relationship based on the original contour tree. The inclusion level can be considered as the depth level in the branch decomposition. In this paper, we use the branch decomposition to analyze the inclusion relationship. The branch decomposition provides a direct way to track inclusion relationship based on branches. Because a branch is usually a concatenation of a sequence of arcs in the contour tree, the subregion corresponding to a branch is a combination of subregions corresponding to arcs. We assume that there exists an outermost surface which includes objects of interest in a data set. So we track the inclusion relationship in the branch decomposition from the branch corresponding to the outer surface. Usually, the outermost surface corresponds to the global minimum of the data set which is one of the critical nodes of the root of the contour tree (see Fig. 3). If the global minimum does not corresponds to the outermost surface, a virtual minimum needs to be applied to set up the contour tree [24]. The inclusion relationship analysis begins from the root and tracks down to higher depth branches as shown in Fig. 3. The further tracking of inclusion relationship is performed along every child branch. The parent branch includes its child branch. In order to emphasize inner structures and deemphasize outer structures, branches with lower depth value are applied with lower opacity and branches with higher depth value are applied with larger opacity. This forms the basis of the opacity residue flow presented in later subsections.

Compared with approaches in [24, 25], our method analyzes inclusion relationships based on the branch decomposition. The branch decomposition overcomes the disadvantage of "over-segmentation" by the general contour tree, and thus avoids the unnecessary inclusion analysis. Furthermore, our method applies transfer functions of different subregions corresponding to branches locally but not globally.

6.2 Residue Flow from Parent to Children

In order to distribute opacity between branches and thus generate OTF of volumetric data, we first divide the opacity range applied to volume data into equal segments based on the maximum depth of the contour tree. The initial value of each opacity band is computed as follows:

$$\alpha_d = \frac{\alpha_r}{d_{\max}},\tag{3}$$

where α_d is the initial opacity band used to render a subregion corresponding to a branch on each depth level, α_r is the opacity range

used to render the overall volume data set, d_{max} is the maximum depth value of the contour tree.

As mentioned in the residue flow model in the previous section, the opacity band applied to each branch is modeled as water and the branch is modeled as a pipe with porous medium inside. According to the Darcy's Law, the absolute head loss value Δh_i between the ends of the current branch in the contour tree is computed by Equation 4:

$$\Delta h_i = \frac{1}{K} \cdot \frac{Q \cdot p_i}{n_i^c},\tag{4}$$

where $\Delta h_i \in [0.0, 1.0]$, p_i is the persistence of the current branch, n_i^c is the number of children of the current branch.

In this paper, Δh_i is used as a factor to evaluate opacity band absorbed by the branch and residue delivered to the next depth level. The opacity band and residue on the *i*th depth level are computed by Equation 5 and Equation 6 respectively:

$$\boldsymbol{\alpha}_{i} = \left(\boldsymbol{\alpha}_{d} + \Delta \boldsymbol{\alpha}_{i-1}\right) \cdot \left(1 - \Delta h_{i}\right), \tag{5}$$

$$\Delta \alpha_i = (\alpha_d + \Delta \alpha_{i-1}) \cdot \Delta h_i, \tag{6}$$

where $i \ge 1$, $\alpha_0 = \alpha_d \cdot (1 - \Delta h_0)$, and $\Delta \alpha_0 = \alpha_d - \alpha_0$. α_i is the opacity band applied to branches on the *i*th depth level, and $\Delta \alpha_i$ is the residue generated on the *i*th depth level (see Fig. 3).

The opacity residue $\Delta \alpha_i$ flows from lower depth level to higher depth level in the contour tree. As mentioned, branches on the lower and higher depth level correspond to outer and inner structures respectively. The flow of the opacity residue in the contour tree allows users to apply larger opacity band to inner structures, and thus emphasize inner structures in volume rendering. The higher the residue is, the bigger the difference between inner and outer structures through modulating flow rate according to Equation 4.

6.3 Residue Flow Among Siblings

The opacity band and residue flowing to branches on each depth level need to be distributed further between sibling branches on that depth level. One of objectives of the further distribution of opacity band and residue is to maximize differences between sibling branches. Various sibling branches have different extreme and saddle values. The saddle value of each branch decides its exact location on the depth level it resides. Importance values of persistence, volume, and hypervolume of each branch are also contributed to enhance differences between sibling branches. Considering these factors, the final opacity band applied to each branch is computed by Equation 7:

$$\alpha_{ij} = \alpha_i \cdot g_{sb} \left(p_{ij}, v_{ij}, h v_{ij} \right) \cdot g_{sd} \left(s_{ij} \right), \qquad \left(j = 1, \dots, n_i^s \right), \tag{7}$$

where α_{ij} is the opacity band applied to the *j*th sibling branch on the *i*th depth level (see Fig. 3). α_i is the basic opacity band on the *i*th depth level and computed by Equation 5. p_{ij} , v_{ij} , and hv_{ij} are persistence, volume, and hypervolume respectively of the *j*th sibling branch on the *i*th depth level. g_{sb} is the function used to control distribution of opacity band between siblings based on importance values p_{ij} , v_{ij} , and hv_{ij} . n_i^s is number of siblings on the *i*th depth level. g_{sd} is the function used to control distribution of saddle value s_{ij} of each branch.

In order to evaluate influences of various importance values of a branch on the opacity band and residue flow, we use a concept of importance triangle [30] to combine various importance values together in order to modulate opacity band. The size (area) of the importance triangle is used as the solution to the function g_{sb} in Equation 7 and computed by Equation 8:

$$g_{sb}(p_{ij}, v_{ij}, hv_{ij}) = I_{ij} = \frac{1}{2}\sqrt{(hv_{ij} \cdot p_{ij})^2 + (v_{ij} \cdot p_{ij})^2 + (hv_{ij} \cdot v_{ij})^2},$$
(8)

where p_{ij} , v_{ij} , and hv_{ij} are firstly normalized by corresponding local maximum within sibling branches respectively on the *i*th depth level.

On the *i*th depth level, sibling branches have different saddle values. Branches with lower saddle values are close to the start of the root of the contour tree, and branches with larger saddle values are far away from the start of the root of the contour tree. Because a branch is a concatenation of a sequence of arcs in the general contour tree, we apply smaller opacity band to branches with lower saddle values and larger opacity band to branches with larger saddle values, in order to emphasize branches with larger saddle values and represent inclusion relationship. Based on this observation, the solution to the function g_{sd} in Equation 7 is computed by Equation 9:

$$g_{sd}\left(s_{ij}\right) = \frac{s_{ij} - s_i^{\min}}{\Delta s_i},\tag{9}$$

where $\Delta s_i = s_i^{\max} - s_i^{\min}$, $s_i^{\max} = \max(s_{i0}, s_{i1}, \dots, s_{i,n_i^s})$, $s_i^{\min} = \min(s_{i0}, s_{i1}, \dots, s_{i,n_i^s})$.

Finally, the actual opacity range applied to the *j*th branch on the *i*th depth level is:

$$\alpha_{ij}^{l} = \alpha_{\min} + \sum_{m=0}^{i-1} \alpha_{m}, \qquad (10)$$

$$\alpha_{ij}^{h} = \alpha_{\min} + \sum_{m=0}^{i-1} \alpha_m + \alpha_{ij}, \qquad (11)$$

where α_{ij}^l is the lower opacity value applied to the current branch. α_{ij}^h is the upper opacity value applied to the current branch. α_{\min} is the minimum opacity value applied to the overall data set.

After getting the opacity range of each branch, one of transfer function shapes is selected to generate actual opacity values. In this paper, we provide five typical transfer function shapes as shown in Fig. 4. Each transfer function shape has its own features: the hat-like shape aims to reveal isosurface-like shapes in the data set; the linear shape can show other structures except isosurfaces. Other transfer function shapes have similar features with the linear shape and hat-like shape. The choice of the transfer function shape depends on the complexity of volumetric data. If a data set contains complex structures inside, the hat-like shape is more effective to reveal isosurfaces of structures and see-through isosurfaces to show inner structures. Otherwise, the linear shape and other shapes can also reveal inner structures effectively.



Fig. 4. Various transfer function shapes used in a branch.

7 COLOR TRANSFER FUNCTION GENERATION

In volume rendering, a color transfer function (CTF) is used to map data features and scalar values to colors in order to label various structures and create aesthetic attraction. It is often defined arbitrarily by trial and error based on personal preferences. Even if some transfer function approaches provide widgets to assign colors to structures, there are no theoretical relations between the selected color and the object [14]. The CTF in this paper aims to incorporate topological features derived from the contour tree into the CTF specification, and focus attention on inner structures. It also aims to label different structures with various colors, and automate CTF generation to create harmonic renderings. Thus the CTF specification becomes a meaningful but not arbitrary process.

When viewing a color object, human visual system characterizes it by its brightness and chromaticity. The latter is defined by hue and saturation. Brightness is a subjective measure of luminous intensity. It embodies the achromatic notion of intensity. Hue is a color attribute and represents a dominant color. Saturation is an expression of the relative purity. A perceptual color space (e.g. CIELAB space) includes all these three perceptual components: hue, saturation, and brightness. In our approach, we use CIELAB to computer colors. HSV color space is used as an interface between color computation and the CTF specification. Thus we select a color triple of hue, lightness, and vividness for each branch of the contour tree. The hue is selected in HSV space, and lightness as well as vividness are selected in the CIELAB space. Specifically, the absolute chroma of a color refers to colorfulness and can be computed in CIELAB space from Equation 12, and vividness is defined as color's relative purity (chroma) and got from Equation 13 [27]:

$$chrom(h,s,v) = \sqrt{a^2 + b^2},$$
(12)

$$vivid(h, s, v) = chrom(h, s, v)/chrom(h, 1, 1),$$
(13)

where a and b are color components in CIELAB color space.

Our work extends Wang et al.'s approach [27] in following ways: various hues are selected for each branch, and topological attributes derived from the contour tree instead of specified manually as in [27] are used in lightness and vividness selection. The contour tree is used to automate the overall color selection process. The color selection process for the CTF of each branch is presented as follows:

- **Step 1** Hue selection. There are *N* branches in the contour tree, which correspond to various subregions/structures in the data set. This step creates various hues for each branch in order to increase contrast between structures. Different *N* hues may be evenly located on the hue wheel in HSV space. We may also use a harmonic color template [5] to select a hue type and limit hue positions inside the harmonic hue range on a harmonic hue wheel. We start to specify colors to each branch from the root of the contour tree. The choice of the color hue type depends on complexity of volumetric data. If a volume data contains large number of objects inside, users may use a hue type with large hue range or even the full hue range in order to increase differences between objects.
- Step 2 Lightness selection. Good global lightness contrast helps to discriminate different features. It also helps in the depth ordering task to discriminate inner structures. The lightness of each branch is computed based on its depth and subtree size. Depth of the branch helps to highlight contrast of inner structures. Branches with lower depth are applied with higher lightness values, and branches with higher depth are applied with lower lightness values. The subtree size allows to consider affects of number of objects on the overall lightness. Branches with larger subtree size are applied with higher lightness in order to get higher contrast between structures and highlight inner structures. The lightness is obtained by Equation 14:

$$L_{ij} = L_{\max} - \frac{d_{ij}}{d_{\max}} \cdot \left(1 - \frac{n_{ij}^s}{n_{\max}^s}\right) \cdot \left(L_{\max} - L_{\min}\right), \quad (14)$$

where L_{ij} is the lightness applied to the current branch, n_{ij}^s is the size of subtree beginning from the current branch, n_{max}^s is the whole contour tree size.

Step 3 Vividness selection. Vividness can be a strong depth cue because decreasing vividness gives the effect of seeing in a fog [21]. It guides the observer to the most important features of the data. The vividness of each branch is obtained based on its depth and importance value. For the former, we apply lower vividness values to branches with lower depth in order to focus users' attention to inner structures. The importance value of each branch is also considered and more important branches are colored with a higher vividness. So this specification can be represented with Equation 15:

$$V_{ij} = V_{\min} + \frac{d_{ij}}{d_{\max}} \cdot I_{ij} \cdot (V_{\max} - V_{\min}), \qquad (15)$$

where V_{ij} is the vividness applied to the current branch, d_{ij} and d_{max} are the depth of the current branch and maximum depth of the current tree, I_{ij} is the importance value of the current branch computed by Equation 8, V_{min} and V_{max} are the minimum and maximum vividness specified by the user which determine the overall vividness of the visualization.

After the hue, lightness, and vividness have been selected, we convert these (h,L,V) triples to their (h,s,v) equivalents and finally to RGB for the transfer function. To obtain (h,s,v) from a given (h,L,V), we use the binary search in a given hue slice in HSV space [27].

8 EXPERIMENTAL RESULTS AND DISCUSSIONS

We conducted experiments on various data sets to demonstrate the effectiveness and utility of the proposed approach in volume rendering. Our system was run on Ubuntu platform on a Dell machine (Intel Core2Duo CPU E4400, 3G RAM) equipped with an NVIDIA GeForce 8300GS graphics card.



Fig. 5. Volume rendered nucleon data set using harmonic hue T-Type and linear transfer function shape.



Fig. 6. Transfer function set generated in Fig. 5.

We first used the proposed approach to automate the transfer function generation for the "nucleon" data set, a $41 \times 41 \times 41$ voxel data set resulting from a simulation of the probability distribution of a nucleon in the atomic nucleus ¹⁶O, see http://www.volvis.org/. Fig. 5 shows the result of the experiment. In this figure, the harmonic color hue T-Type and linear transfer function shape are used as shown in Fig. 5(b). Fig. 6 shows the transfer function set generated in this experiment. By observing the result, the nested inner structures and outer layers are clearly revealed. This figure clearly shows various layers from outside to inner side with increased opacity, and increased vividness and decreased lightness. Harmonic colors give users aesthetic attraction and enhance understanding of structures.

Fig. 7 shows the rendering result of the "fuel" data set, a $64 \times 64 \times 64$ voxel data set resulting from a simulation of fuel injected into a combustion chamber, see http://www.volvis.org/. Using the contour tree controlled residue flow model, it is possible to automatically reveal inner structures while rendering surrounding layers using a low opacity. Colors of various structures are differentiated, and inner



Fig. 7. Volume rendered fuel data set using full hue range and linear transfer function shape.

structures are emphasized and applied with colors with high vividness and low lightness. Thus colors of structures controlled by topological attributes enhance understanding of structures.



Fig. 8. Volume rendered knee data set using full hue range and hat-like transfer function shape.



Fig. 9. Volume rended foot data set using harmonic hue T-Type and hat-like transfer function shape.

To demonstrate how our proposed method performs on real-world medical data sets, we conducted an experiment on a moderately sized $379 \times 229 \times 305$ voxel CT knee data set (see http://www9.informatik.uni-erlangen.de/External/vollib/) as shown in Fig. 8, and a $256 \times 256 \times 256$ voxel CT foot data set (see http://www.volvis.org/) as shown in Fig. 9. The full hue range and hat-like transfer function shape are used in Fig. 8, and the harmonic color hue T-Type and hat-like transfer function shape are used in Fig. 9. The inner bone structures and outer surface layers are clearly revealed with different colors at the same time. Inner structures are rendered using high opacity, high vividness, and low lightness, while outer surfaces are rendered using low opacity, low vividness, and high lightness. The transfer function for each structure is automatically and locally defined, in order to highlight internal structures while revealing outer layers.

The proposed approach is also applied to a more complicated data set, an MR head data set with brain tumors inside (data courtesy of B Terwey, Bremen). In the MR head data set, brain tumors are often difficult to be visualized because of inclusions of tumors inside the brain and complicated brain structures. The full hue range and hat-like transfer function shape are used in this experiment. In this figure, because opacity residues flow from outer structures to internal structures, the brain tumor is highlighted using high opacity, high vividness, and low lightness. Meanwhile, inner brain shape and outer head surfaces are clearly depicted using low opacity to show context. The rendering result enhances users' understanding of the tumor in the brain data.

The effectiveness of the proposed approach can also be observed



Fig. 10. Volume rendered tumor head data set using full hue range and hat-like transfer function shape.

in movies accompanying this paper, which show how rendering results are changed when the opacity flow rate Q is increased for the "fuel" and "nucleon" data sets. From the movies, we see that outer surfaces become more transparent and then peeled off, while inner structures become more clear with the increasing of the opacity flow rate Q. This is because that more opacity residues flow to inner structures when the opacity flow rate is increased. The movies give users better understanding of the effectiveness of the residue flow model in transfer function generations.

The volume rendering pipeline in this paper is implemented based on graphics processing unit (GPU) fragment programs. The performance of the system allows real-time exploration of the volumetric data. The processing time of automatically generating transfer functions for a data set depends on branch number of the contour tree and size of the data set. For small data sets, such as the "fuel" data set, the branch number is usually within a manageable size (e.g. less than 90 before simplification and around or less than 20 after simplification). The transfer function can be interactively generated for such data sets. We created movies for "fuel" and "nucleon" data sets to show how transfer functions are interactively generated when changing the opacity flow rate in this paper. The frame rate (the size of the view port is 400×374 in this paper) for the "fuel" data set is 36.5 fps (20 branches, and original branch number is 86), and the frame rate for the "nucleon" data set is 55.8 fps (8 branches, and original branch number is 40). However, frame rates for larger data sets are still sufficiently high for automatically transfer function generations. For example, the processing time of generating transfer functions for the CT knee data set is 1.27s (10 branches, and original branch number is 954968). After generating transfer functions, users can interactively explore data sets.

From the experiments, we can see that the proposed approach can effectively automatically generate transfer functions. Compared with conventional transfer function methods, our approach has following advantages: 1) It only requires users to simply control the opacity residue flow rate, instead of time-consuming interactions for tweaking complex transfer function parameters to explore volume data. 2) The generated transfer function automatically reveals topological relations (such as inclusion relationships), instead of manually controlling parameters to depict topological relations. Furthermore, it is even impossible for conventional transfer function methods to carry out a such task if scalar ranges of various structures are overlapped in a data set. 3) It only requires users to understand the concepts of flow rate and transfer function shape, instead of complex visualization expert knowledge as required by conventional methods. Although some tried to overcome this problem by introducing semantic layers to ease transfer function specifications [20, 19], they still need complex decisions and interactions. 4) The automatically generated transfer functions also provide initial estimates even if users are to manually fine tune them. 5) Because the proposed approach does not require much involvement from users, it is even effective for an unknown volume data set. 6) The advantage of the residue flow model is that it allows the opacity residue sinks down to leaves inside an upsidedown contour tree, and results in larger opacities of branches on the higher depth level. This means that inner structures get larger opacity than outer structures, and thus emphasize inner structures. In summary, our new approach greatly improved the transfer function generation process,

and it would be significantly beneficial to users who are not visualization expert.

Various opacity transfer function shapes are provided in the framework. Opacity transfer functions with hat-like shape try to capture isosurfaces in the data set. This results in the possibility that users may see through outer surfaces to perceive inner structures in volume rendering. However, other transfer function shapes are also powerful in revealing structures in volumetric data. For example, the linear shape used in Fig. 5 and Fig. 7 is effective in depicting inner structures of data sets. Compared with Wang et al.'s method [27] which uses parameters such as importance levels specified manually by users to control color generations, our approach controls color generations based on topological attributes directly derived from the contour tree. This is more meaningful and effective. We used both full hue range color and harmonic color in the experiments. Full hue range color provides high color contrast of structures, while harmonic color creates aesthetic attraction in the rendering.

The effectiveness of our method depends on the quality of the contour tree, which indicated by signal-to-noise ratio (SNR) and contrast of volumetric data. Data sets with higher SNR and higher contrast create higher quality contour trees which can more accurately represent topology of data sets, and thus generate transfer functions more effectively. Fortunately, with the improvement of scanning techniques, scientists may get volumetric data with higher quality, and thus our method becomes more effective in transfer function generations. Currently, we may improve the quality of the contour tree through preprocessing of volumetric data (e.g. filtering).

9 CONCLUSIONS AND FUTURE WORK

This paper presented a new paradigm for automating transfer function generations in volume rendering. In the proposed approach, a residue flow model based on Darcy's Law was set up to differentiate the distribution of opacity between branches of the contour tree. Topological attributes were also used to control color selection in a perceptual color space and create harmonic color transfer functions. The generated transfer functions depicted inclusion relationship between structures and maximized differences between them. Experiments on various data sets showed effectiveness of our approach in automation of transfer function generations. In summary, the proposed approach allows more efficient automation of transfer function generation, and data exploration through the control of opacity residue flow rate rather than complicated low-level transfer function parameters.

In the future, more choices of local transfer function shapes for each branch will be developed. For example, multi-dimensional transfer functions and other approaches reviewed in Section 2 can be adapted for each branch to automate and improve the overall effectiveness of transfer function generations. We also plan to automate the selection of local transfer function shapes based on topological attributes and other data features.

ACKNOWLEDGMENTS

The authors wish to thank Scott Dillard for some basic contour tree codes, and National ICT Australia (NICTA) and China Scholarship Council (CSC) for their support for this research. We also thank the anonymous reviewers for their valuable comments.

REFERENCES

- C. A. Aumann and E. D. Ford. Modeling tree water flow as an unsaturated flow through a porous medium. *Journal of Theoretical Biology*, 219(4):415–429, 2002.
- [2] C. Bajaj, V. Pascucci, and D. Schikore. The contour spectrum. In *IEEE Visualization* '97, pages 167–173, 1997.
- [3] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75–94, 2003.
- [4] H. Carr, J. Snoeyink, and M. van de Panne. Simplifying flexible isosurfaces using local geometric measures. In *Proceedings of IEEE conference* on Visualization '04, pages 497–504, 2004.
- [5] D. Cohen-Or, O. Sorkine, T. L. R. Gal, and Y.-Q. Xu. Color harmonization. ACM Transactions on Graphics, 25(3):624–630, 2006.

- [6] C. Correa and K.-L. Ma. Size-based transfer functions: A new volume exploration technique. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1380–1387, 2008.
- [7] S. Fang, T. Biddlecome, and M. Tuceryan. Image-based transfer function design for data exploration in volume visualization. In *IEEE Visualization* '98, pages 319–326, 1998.
- [8] I. Fujishiro, T. Azuma, and Y. Takeshima. Automating transfer function design for comprehensible volume rendering based on 3d field topology analysis. In *IEEE Visualization*'99, 1999.
- [9] R. Huang and K.-L. Ma. Rgvis: Region growing based techniques for volume visualization. In *Proceedings of Pacific Graphics Conference* 2003, pages 355–363, Washington, DC, USA, 2003.
- [10] M. Kasenow. Determination of Hydraulic Conductivity from Grain Size Analysis. Water Resources Publication, 2002.
- [11] G. Kindlmann and J. W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *IEEE Symposium on Volume Visualization*, pages 79–86, 1998.
- [12] G. L. Kindlmann, R. T. Whitaker, T. Tasdizen, and T. Möller. Curvaturebased transfer functions for direct volume rendering: Methods and applications. In *Proceedings of IEEE Visualization*, pages 513–520, 2003.
- [13] J. Kniss, G. Kindlmann, and C. Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of IEEE Visualization 2001*, pages 255–262, 2001.
- [14] J. Kniss, G. Kindlmann, and C. Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):270–285, 2002.
- [15] N. Kreŝić. Hydrogeology and Groundwater Modeling. CRC Press, 2007.
- [16] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(5):29–37, May 1988.
- [17] J. Marks and et al. Design galleries: A general approach to setting parameters for computer graphics and animation. In ACM SIGGRAPH Conference on Computer Graphics '97, pages 389–400, 1997.
- [18] V. Pascucci, K. Cole-McLaughlin, and G. Scorzelli. Multi-resolution computation and presentation of contour trees. In *Proceedings of the IASTED conference on Visualization, Imaging, and Image Processing*, pages 452–290, 2004.
- [19] P. Rautek, S. Bruckner, and M. E. Gröller. Semantic layers for illustrative volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1336–1343, 2007.
- [20] C. Rezk-Salama, M. Keller, and P. Kohlmann. High-level user interfaces for transfer function design with semantics. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1021–1028, 2006.
- [21] P. Sabella. A rendering algorithm for visualizing 3d scalar fields. In Proceedings of the 15th annual conference on Computer graphics and interactive techniques (SIGGRAPH'88), pages 51–58, 1988.
- [22] M. A. Selver and C. Güzelis. Semi-automatic transfer function initialization for abdominal visualization using self generating hierarchical radial basis function networks. *IEEE Transactions on Visualization and Computer Graphics*, 15(3), 2009.
- [23] S. Takahashi, Y. Takeshima, and I. Fujishiro. Topological volume skeletonization and its application to transfer function design. *Graphical Models*, 66(1):24–49, 2004.
- [24] S. Takahashi, Y. Takeshima, I. Fujishiro, and G. M. Nielson. Scientific Visualization: The Visual Extraction of Knowledge from Data, chapter Emphasizing Isosurface Embeddings in Direct Volume Rendering, pages 185–206. Springer-Verlag, 2005.
- [25] Y. Takeshima, S. Takahashi, I. Fujishiro, and G. M. Nielson. Introducing topological attributes for objective-based visualization of simulated datasets. In *Volume Graphics*, pages 137–145, 2005.
- [26] F. Tzeng, E. Lum, and K. Ma. A novel interface for higher-dimensional classification of volume data. In *IEEE Visualization 2003*, 2003.
- [27] L. Wang, J. Giesen, K. T. McDonnell, P. Zolliker, and K. Mueller. Color design for illustrative visualization. *IEEE Transactions on Visualization* and Computer Graphics, 14(6):1739–1754, 2008.
- [28] L. Wang and K. Mueller. Harmonic colormaps for volume visualization. In Proceedings of IEEE/EG Symposium on Volume and Point-Based Graphics, pages 322–325, Los Angeles, USA, 2008.
- [29] G. H. Weber, S. E. Dillard, H. Carr, V. Pascucci, and B. Hamann. Topology-controlled volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):330–341, 2007.
- [30] J. Zhou and M. Takatsuka. Contour tree simplification based on a combined approach. Technical Report TR 624, School of Information Engineering, The University of Sydney, Australia, 2008.