Volume Graphics (2005)
E. Gröller, I. Fujishiro (Editors)

Introducing Topological Attributes for Objective-Based Visualization of Simulated Datasets

Y. Takeshima[1] S. Takahashi[2] I. Fujishiro[1] G. M. Nielson[3]

[1] Tohoku University, Japan [2] The University of Tokyo, Japan [3] Arizona State University, USA

Abstract

Recent development in the design of multi-dimensional transfer functions allows us to automatically generate comprehensible visualization images of given volumes by taking into account local features such as differentials and curvatures. However, especially when visualizing volumes obtained by scientific simulations, observers usually exploit their knowledge about the simulation settings as the clues to the effective control of visualization parameters for their own specific purposes. This paper therefore presents an objective-based framework for visualizing simulated volume datasets by introducing a new set of topological attributes. These topological attributes are calculated from the level-set graph of a given volume dataset, and thus differ from the conventional local attributes in that they also illuminate the global structure of the volume. The present framework provides a systematic means of emphasizing the underlying volume features, such as nested structures of isosurfaces, configuration of isosurface trajectories, and transitions of isosurface's topological type. Several combinations of the topological attributes together with the associated transfer function designs are devised and applied to real simulated datasets in order to demonstrate the feasibility of the present framework.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Display algorithms

1. Introduction

The recent development of high-performance computer system has made it easier for us to obtain large simulated volume datasets. Direct volume rendering is a standard technique for projecting all optically-encoded samples onto the screen at once to peer into the inner structures involved in a volume data. Data-centric approaches to the design of *transfer functions* (*TFs*) have recently been well-established, which perform mathematical analysis of the data prior to pertinent rendering.

The advent of *multi-dimensional* TFs is one of the latest major achievements in the volume visualization research. As opposed to the traditional one-dimensional TFs that only consider a voxel's scalar field value, the multi-dimensional TFs assign auxiliary attributes to the voxels to construct their sophisticated parametric domains. For example, if we take into account additional attributes, such as higher-order gradient fields, as well as the original scalar field, we can generate more comprehensible visualization images, where the relative geometric positions and subtle differences among the volumetric features are revealed in an accentuated manner. Such multi-dimensional TFs have played an important role in gaining clear insights into a target volume data, especially for the cases without any prior knowledge about the data.

In actual situations, however, observers usually might want to exploit their background knowledge about a target volume data as the clues to perform detailed analysis of the data. For example, when visualizing volumes obtained by scientific simulations, they can utilize their own knowledge about the simulation settings to extract the global characteristics of the volumes and to locate regions of particular interest. If they are allowed to design multi-dimensional TFs using proper attributes so as to encapsulate such advance knowledge, they can readily yield visualization results to fulfill their purposes.

Nevertheless, nearly all attributes for the conventional multi-dimensional TFs are based on local features, such as differentials and curvatures, and cannot capture the global structure of the volume contrary to the observer's purposes. As such, the observers are forced to design their TFs by *juxtaposing* the explicitly-described local features together with the global structure in their mind, and thus leading to cogni-

tive load as well as qualitative limitations of resultant visualization images.

This paper therefore introduces a new set of topological attributes to establish a new framework that is intended to realize objective-based assistance, especially for visualizing simulated volume datasets. In our framework, transfer functions are designed so that feature isosurfaces which represent topological features of a given volume dataset are emphasized. However, only the feature isosurfaces selected according to observer's purposes instead of all the feature isosurfaces are emphasized using topological attributes. Topological attributes proposed herein are derived from the level-set graph of a given volume, which delineates the topological evolution of an isosurface with respect to the scalar field. The level-set graph can capture not only the local features but also the global structure of the volume because each node locates the local topological change in the evolving isosurface, and its link the global connection in between. To the best of the authors' knowledge, topological attributes derivable from the level-set graph can be viewed as the only attributes that can reflect the global structure as well as local features of the volume, and can be used to encapsulate the observer's intentions fully into the multi-dimensional TFs. Needless to say, these topological attributes are expected to provide more valuable analysis clues than the conventional local feature-based attributes when visually exploring unknown volume datasets along with background knowledge.

In our framework, a specific kind of level-set graph, called *volume skeleton tree* (*VST*), is extracted robustly through a method, called *topological volume skeletonization*, even from practical discrete volume samples involving high-frequency noise and/or zero-gradients [TTF04, TNTF04]. Topological attributes based on the VST can provide a systematic means of emphasizing the underlying volume features, including nested structures of isosurfaces, configuration of the isosurface trajectories, and transitions of isosurface's topological type. This paper develops several combinations of these topological attributes together with the associated TF designs, and applies them to real simulation datasets in order to demonstrate the feasibility of the present framework.

The remainder of this paper is organized as follows. Section 2 describes previous work related to ours. Section 3 outlines the construction and notation of the VST, upon which we will build to come up with topological attributes. Section 4 proposes a set of topological attributes for constructing more powerful multi-dimensional TFs, and describes how these attribute values are calculated from the VST. Section 5 evaluates the effectiveness of the designs of topological multi-dimensional TFs empirically with applications to several real simulated datasets. Lastly, Section 6 concludes the paper with possible future research themes.

2. Related Work

Visualization design is generally divided into the consecutive three steps, i.e., conceptual design, articulation [KASS91], and parameter tweaking. *Transfer functions* (*TFs*) are used in the third and last step for parameter tweaking as the final determinant of effects obtained by visualizing a single 3D scalar field according to the observers' purposes. For the sake of enhanced expressiveness, much energy has been devoted to the invention of *multi-dimensional* TFs.

The idea of multi-dimensional TFs can be traced back to Levoy's pioneering work [Lev88], where he used gradient magnitude proficiently as an auxiliary attribute value for designing an opacity TF which can capture some features of a volume data. Kindlmann and Durkin [KD98] were inspired by the work, and formalized a data structure called *histogram volume*, that is spanned by a scalar field value, and its first and second partial derivatives, in order to design TFs that effectively emphasize boundaries between different neighboring materials within the volume data. The idea was generalized later by Kniss et al. [KKH02] to propose the design of three-dimensional TFs. Meanwhile, Hladůvka et al. [HKG00] proposed another type of multi-dimensional TF that employs as its attributes, the maximum and minimum principal curvatures of the volume function. Recently, this framework has been further sophisticated by Kindlmann et al. [KWTM03]. However, all of these attributes employed in multi-dimensional TFs capture only local features of a volume, and are expected to be effective to strike up object boundaries embedded in unknown datasets, but having their limit to convey any intentions of the observers. On the contrary, Tenginakai et al. [TLM01] proposed a global attribute value that is derivable from statistical analysis of the volume data. However, this attribute can be defined independently of volumetric local properties, and cannot be used in the full-fledged volume visualization.

This becomes a strong incentive for us to rivet our focus on level-set graphs. While Fujishiro et al. [FAT99, FATT00, FTTY02] and Weber et al. [WSHH02] presented methods for locating topological changes of isosurfaces for this purpose, they still considered local features around the critical points only and ignored their associated global connectivity. Our approach to TF design [TTFN05] made it possible to emphasize such global structures efficiently by extracting the topological transitions of isosurfaces from input volumes. It is still difficult to generate visualization images with observer's purposes only from one topological attribute. The observer therefore should use more attributes effectively. Although the specific set of optimal parameter values cannot be uniquely determined for each volume dataset, the parameter values which are calculated automatically from our method play a role to obtain the initial guess for the further analysis.

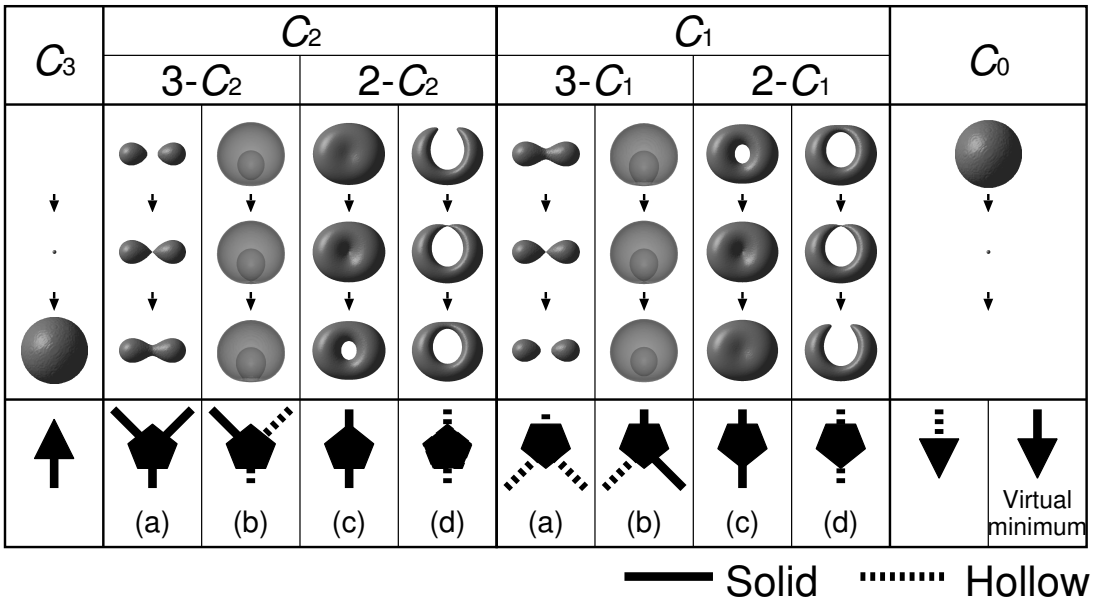


Figure 1: *The connectivity of a critical point of each type in the VST.*

3. Volume Skeleton Tree

Prior to calculating topological attributes, we extract a level-set graph of a given volume dataset using the topological volume skeletonization algorithm [TNTF04]. The level-set graph, called volume skeleton tree (VST) [TTF04], allows us to evaluate the topological attributes of each voxel by illuminating both the global and local features of the volume dataset.

A node of the VST represents *critical point* that has the change either in the number of connected isosurface components or in the genus of each of the isosurface components. Critical points are classified into four groups: maxima (C_3), saddles (C_2, C_1), and minima (C_0), which represent isosurface appearance, merging, splitting, and disappearance, respectively, as the scalar field value reduces. Here, an index of a critical point represents the number of negative eigenvalues of the Hessian matrix there. A link of the VST represents an topology-preserving transition of an isosurface connected component. A link is defined *solid* if its isosurface component expands as the scalar field value reduces while *hollow* if it shrinks.

The isosurface merging at C_2 and splitting at C_1 have both four topological transition paths with different isosurface spatial configurations, as shown in Figure 1. In what follows, the VST uses the notation for the critical points with its own connectivity as illustrated in Figure 1, where the solid incident link represents a solid isosurface while the broken link represents hollow. Saddle points of $C_i (i = 1,2)$ are classified into 3-C_i and 2-C_i, according to their degree (valence). For later convenience, all the boundary voxels are assumed to be connected to the virtual minimum having $-\infty$ as its scalar field value [TTF04]. Note that the link incident to the C_0 node is solid when the node is the virtual minimum as shown in Figure 1. An example of the VST together with isosurface transitions is illustrated in Figure 2. In our implementation, a node has its coordinates and scalar field value, and a link has its genus and index of adjacent nodes.

The VST has much in common with the contour tree [BPS97] in the mathematical sense. However, the VST decomposes a multiple (degenerate) critical node into simple ones to extract the global structures such as nested structure of isosurfaces, whereas the contour tree keeps critical points of multiple degrees directly without resolving them into simple ones.

4. Formulation of Topological Attributes

This section introduces a new set of visualization parameters called *topological attributes* to incorporate user's objectives into volume visualization. In our framework, the topological attributes are formulated using the VST, and thus successfully inherit both global and local features of the volume from the VST. This means that the attributes can serve as auxiliary variables for multi-dimensional TFs that illuminate volume features according to the user's objectives. In the remainder of this section, these topological attributes are classified into three categories: inclusion levels (Section 4.1), isosurface-trajectory distances (Section 4.2), and isosurface genera (Section 4.3).

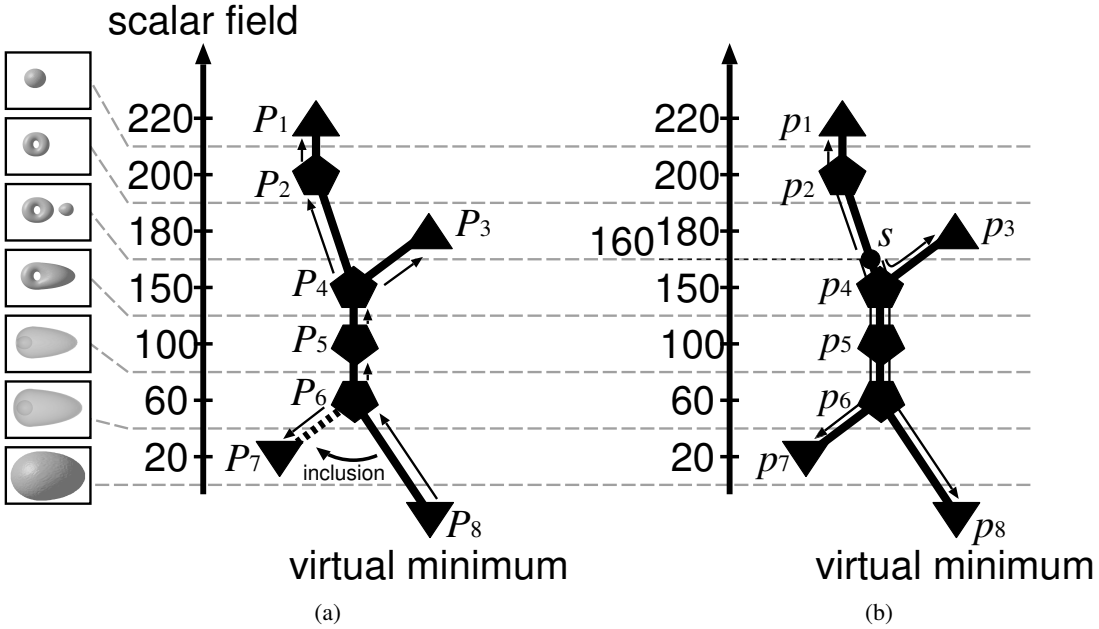


Figure 2: *The process of calculating (a) the inclusion levels and (b) isosurface-trajectory distances.*

4.1. Inclusion levels

A volume dataset often includes a complicated nested structure where one feature subvolume completely encloses others within some range of the scalar field. Conventional single-dimensional TFs cannot discriminate between such inner and outer feature subvolumes because they depend only on the scalar field, and thus assign the same color and opacity values to both of the subvolumes. This motivates us to introduce an *inclusion level* together with multi-dimensional TFs. Here, the inclusion level of a voxel represents the depth of its associated isosurface in the nested structure at the corresponding scalar field value, and serves as an additional variable for the multi-dimensional TFs that emphasize such nested structures. For computing the inclusion levels, our framework employs the algorithm described in [TTFN05].

It is clear from Figure 1 that isosurface nested structures originate only from the transition paths in C_2(b) and C_1(b). This suggests that we can locate such isosurface inclusions directly from the VST if we can identify all the nodes that correspond to the above transition paths. Indeed, this can be done by identifying all the links of the VST with solid or hollow while tracing the VST thoroughly.

Let us calculate the inclusion level for each link of the VST shown in Figure 2(a) by detecting isosurface nested structures. The tracing process starts from the virtual minimum P_8 because its incident link $\overline{P_6P_8}$ is known to be solid (See Section 3). The first node to be traversed is P_6, which can be identified with the node of the 3-C_1(b) type in Figure 1. This is because the node has one upward and two downward incident links and one of the downward links (i.e. $\overline{P_6P_8}$) has already turned out to be solid. Simultaneously, it becomes apparent from Figure 1 that the links $\overline{P_5P_6}$ and $\overline{P_6P_7}$ are solid and hollow, respectively, and the isosurface component of the link $\overline{P_6P_8}$ includes that of $\overline{P_6P_7}$ within the scalar field range from 20 to 60. Repeating this process makes it possible to identify all the links with solid and hollow isosurfaces by referring to the connectivity patterns in Figure 1, and thus to locate all the nested structures in the volume. In this example, the inclusion level of the link $\overline{P_6P_7}$ is 1 while those of the other links are 0. Note that our framework can assign an appropriate inclusion level to each voxel because a link of the VST possesses a list of voxels associated with it.

4.2. Isosurface-trajectory distances

Considering the transition of each connected isosurface component according to the scalar field, it is preferable to formulate a measure of closeness between any two of the isosurface components in the volume. In our framework, the distance between two isosurface components is defined as the length of the shortest path between the two corresponding points on the VST. This new topological attribute is referred to as *isosurface-trajectory distance* in this paper.

Let us calculate the distance of two isosurface components along the isosurface trajectory. Suppose that the points p and q on the VST correspond to the two isosurface components, respectively. The distance $D(p,q)$ is defined as the

difference in the scalar field between p and q along the shortest path on the VST.

For example, as shown in Figure 2(b), we define the isosurface component on the link $\overline{P_2P_4}$ with the scalar field value 160 as a starting point s. In this case, the shortest path of each critical point from s is indicated by the corresponding arrow in the figure. The distance from s to P_2, which is denoted as $D(s,P_2)$, amounts to be the difference in the scalar field value between s and P_2 along the link $\overline{P_2P_4}$ because s lies on that link, which gives the distance as $D(s,P_2) = |200-160| = 40$. The shortest path between s and P_3 is the route s–P_4–P_3, and thus the corresponding distance can be obtained as $D(s,P_3) = D(s,P_4)+D(s,P_3) = |160-150|+|150-180| = 40$. Using the same calculation process, we obtain $D(s,P_1) = 60$, $D(s,P_4) = 10$, $D(s,P_5) = 50$, $D(s,P_6) = 100$, and $D(s,P_7) = 140$.

The formulation of this distance inspires the definition of another topological attribute called *integral of the isosurface-trajectory distance function*, which represents the balance of the isosurface transitions along its trajectories. We can employ it to emphasize the topological changes (isosurface appearance and disappearance) around maxima and minima. The new attribute is obtained by calculating the integral of the trajectory distance between two points by moving one point over the VST while fixing the other. Its mathematical definition can be written as:

$$I(p) = \int_{q\in \mathrm{G}} D(p,q)dG,$$

where G represents the domain of the VST. In practice, we take a set of uniform sample points on the VST as Q, and approximate the integral by

$$I(p) = \sum_{q\in Q} D(p,q).$$

The idea of this topological attribute originates from the histogram of the shortest distance between two random points on the 3D surface by Osada et al. [OFCD01], and the geodesic distance between two points along the surface by Hilaga et al. [HSKK01]. An important property of this integral is that the global minimum represents the center of the isosurface trajectory with respect to scalar field.

Assume that the virtual minimum has 0 as its scalar field value. By sampling the points over the VST with an interval of 1, the integrals of P_1 in Figure 2(b) for instance can be calculated as

$$\begin{aligned} I(P_1) &= \sum_{t\in\overline{P_1P_2}} D(P_1,t) + \sum_{t\in\overline{P_2P_4}} D(P_1,t) + \sum_{t\in\overline{P_3P_4}} D(P_1,t) \\ &\quad + \sum_{t\in\overline{P_4P_5}} D(P_1,t) + \sum_{t\in\overline{P_5P_6}} D(P_1,t) \\ &\quad + \sum_{t\in\overline{P_6P_7}} D(P_1,t) + \sum_{t\in\overline{P_6P_8}} D(P_1,t) \\ &= \sum_{n=1}^{D(P_1,P_2)} n + \sum_{n=D(P_1,P_2)+1}^{D(P_1,P_4)} n + \sum_{n=D(P_1,P_4)+1}^{D(P_1,P_3)} n \\ &\quad + \sum_{n=D(P_1,P_4)+1}^{D(P_1,P_5)} n + \sum_{n=D(P_1,P_5)+1}^{D(P_1,P_6)} n \\ &\quad + \sum_{n=D(P_1,P_6)+1}^{D(P_1,P_7)} n + \sum_{n=D(P_1,P_6)+1}^{D(P_1,P_8)} n \\ &= 34,095. \end{aligned}$$

Using the same calculation process, we obtain $I(P_2) = 28,695$, $I(P_3) = 29,270$, $I(P_4) = 18,695$, $I(P_5) = 16,695$, $I(P_6) = 18,695$, $I(P_7) = 28,695$, and $I(P_8) = 30,685$. The global minimum is found at the point on the link $\overline{P_4P_5}$ with the scalar field value 105.

In our framework, integral of the isosurface-trajectory distance is assigned to each voxel just like inclusion level.

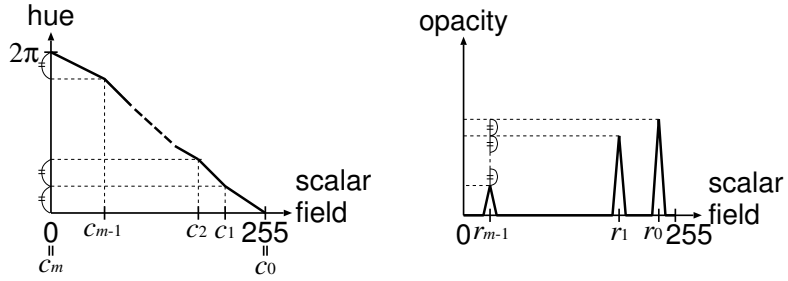
4.3. Isosurface genera

Another topological attribute in our framework is an *isosurface genus*, which is equivalent to the number of holes on each of the isosurface components. Actually, the change in this number often outlines some distinctive feature subvolume embedded in the given dataset. This attribute is already accessible in our framework because our framework computes the attribute value for each voxel in the process of the topological volume skeletonization [TNTF04]. Employing this attribute as an auxiliary variable for multi-dimensional TFs allows us to emphasize isosurface components according to the number of holes on them in our visualization process.

5. Designing Multi-Dimensional Transfer Functions with Topological Attributes

In this section, we illustrate that adaptive usage of those topological attributes, which were formulated in the previous section, makes it possible to obtain visualization results emphasizing various structures of volumes effectively. As described in Section 2, in the modern volume visualization, local feature-based attributes usually have been specified as auxiliary variables for multi-dimensional TFs. It should be urged here again that unlike those local traditional attributes, our topological attributes can capture the global structure of the volume as well.

First, we describe how we make TFs depend on a single scalar field, which is an indispensable variable for any volume visualization (Section 5.1). Then, we will design objective-based multi-dimensional TFs which use topological attributes as their auxiliary variables in addition to the scalar field for real simulated datasets (Sections 5.2–5.4).



(a) Color (b) Opacity

Figure 3: *Basic design principle of 1D transfer functions (TFs).*

5.1. Basic design of 1D transfer functions

Our basic principle to design a 1D TF depending on a scalar field is to accentuate the topological change of an isosurface within the volumetric domain [TTF04].

The scalar field value corresponding to a critical point is referred to as *critical field value* here. In our implementation, a scalar filed value is transformed affinely into an unsigned 8 bit integer value in $[0,255]$. Here, $m-1$ critical field values are denoted with $c_1, c_2, \cdots, c_{m-1}$ in a descending order of the scalar field, and let c_0 and c_m be 255 and 0, respectively. Then the 1D TFs can be designed as in Figure 3.

As shown in Figure 3(a), the color TF is defined piecewise using the range of the HSV hex-cone $[0,2\pi]$ so that the function value decreases linearly over an evenly-divided hue interval for each of the field intervals $[c_{i+1}, c_i]$. This setting allows us to assign a steep color gradation to the regions where the consecutive critical field values are located in closer proximity.

For the opacity TF, as shown in Figure 3(b), we define the mid-value of each of the intervals $[c_{i+1}, c_i](i = 0, \ldots, m-1)$ bounded by consecutive critical field values as the *representative field value* $r_i = (c_{i+1} + c_i)/2$. The actual opacity TF accentuates the *representative isosurfaces* corresponding to the m representative filed values $r_0, \ldots, r_{m-1}$. Furthermore, in order to accentuate each of the representative isosurfaces individually, we assign the zero value to all the field domain, except for local hat functions centered at the representative field values $r_0, \ldots, r_{m-1}$. In our framework at the design of 1D TFs, since the outermost solid isosurface does not shrink as the field value decreases, we minimize the occlusion artifacts induced by the isosurface nested structure by decreasing the height of the hat functions for r_{m-1} through r_0 by a fixed amount.

Generally, the level-set graph including the VST is too sensitive to high-frequency noise in the volume datasets, and thus often detects many minor critical points. Consequently, the VST should be simplifyed by removing such unnecessary critical points to reduce the number of feature isosurfaces until the isosurfaces can be distinguished each other [TNTF04, CSvdP04]. In our framework, the number of isosurfaces is reduced to ten or less empirically [TNTF04].

5.2. Revealing isosurface nested structures

Some volume datasets may be characterized with their isosurface nested structures. Figure 4 visualizes a snapshot volume for 3D fuel density distribution excerpted from a time-varying dataset simulating the process of implosion in laser fusion [SMSY02], where small bubble-spike structures evolve around a contact surface between a fuel ball (inner) and pusher (outer) during the stagnation phase. The fuel-pusher contact surface can be identified with an isosurface extracted by observing the rapid gradients of the fuel density field. We can learn from the specific simulation setting that the extracted isosurface has two nested connected components, and the contact surface of our interest is occluded by the other outer component residing in the pusher domain, which is nothing but a phantom surface induced by the action-reaction effect [SMSY02].

Figure 4(a) shows the VST for the implosion dataset, where the skeletal structure of the complex fuel density distribution has been extracted with an intentional control of VST simplification. A glance at the VST around the field interval $[14,176]$ finds a nested structure where connected isosurface components corresponding to the links $\overline{P_2P_3}$, $\overline{P_3P_4}$, and $\overline{P_3P_5}$ are included by another connected isosurface component corresponding to the link $\overline{P_2P_6}$.

A volume-rendered image is shown with the topologically-accentuated 1D opacity TF in Figure 4(b), from which we can see that after the scalar field itself has been topologically-accentuated, we still suffer from a problem that the inner isosurface components of interest for the observer are indeed occluded by the outer spherical isosurface component. Contrary to that, as shown in Figure 4(c), if we devise the 2D opacity TF which depends on the inclusion level as well to assign a lower opacity value to voxels on the outer isosurface component than to voxels on the inner ones, we can observe the optically-deeper bubble-spike structures more clearly than in Figure 4(b). Furthermore, by assigning

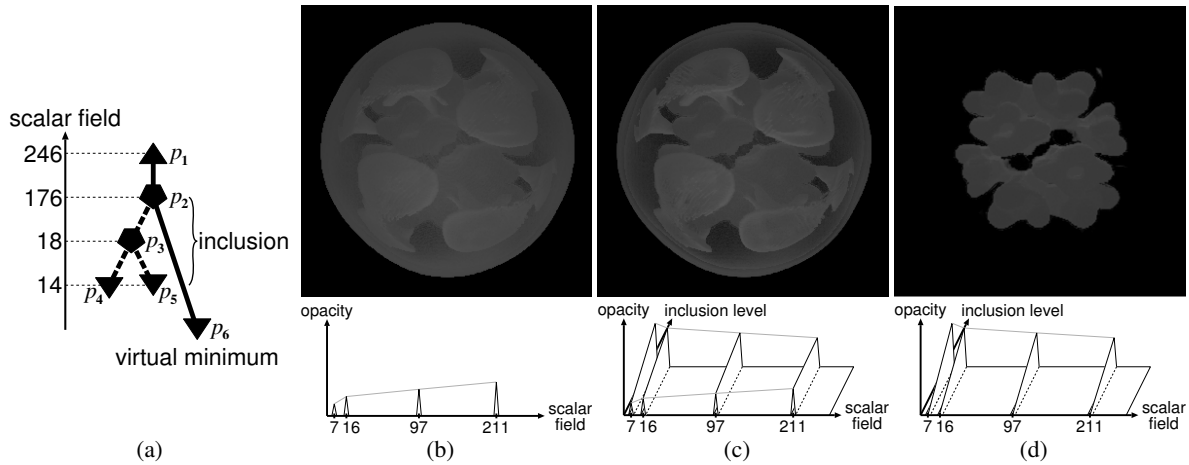


Figure 4: *Visualizing simulated implosion in laser fusion: (a) The corresponding VST, (b) with topologically-accentuated 1D opacity TF, (c) with 2D opacity TF depending also on the inclusion level, and (d) with 2D opacity TF that visually extracts inner structures.*

zero opacity to the outermost representative isosurfaces, we can visually extract the involved nested structure only as shown in Figure 4(d). These two visualization results surely reflect the above-mentioned simulation setting.

Note that throughout the three cases, we consistently use the 1D color TF, which was designed based on the basic principle described in Section 5.1.

5.3. Trailing symmetric isosurface trajectories

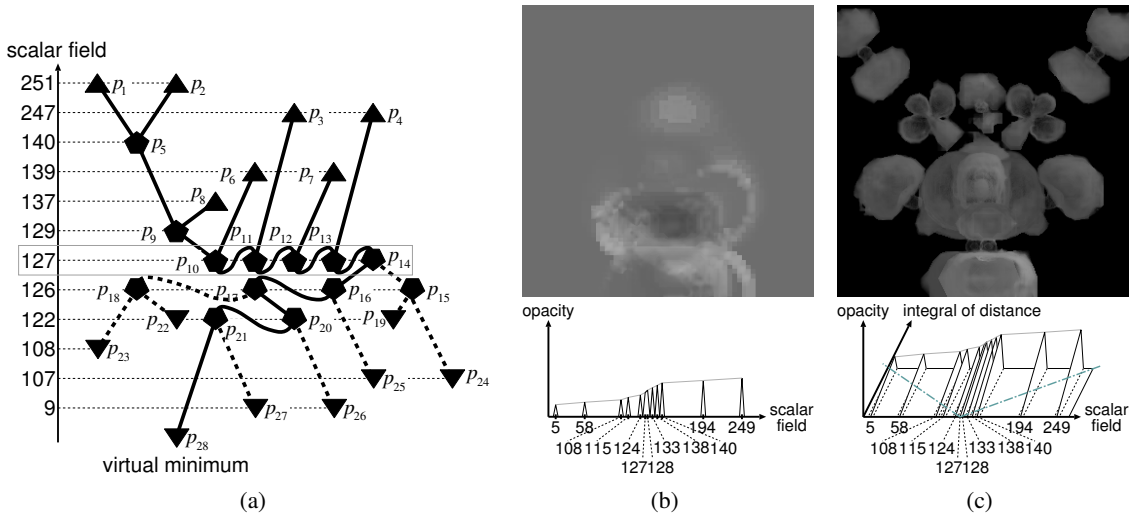
Simulated volume datasets such as distributions of energy functions often contain a symmetric isosurface trajectory with respect to the scalar field. In such dataset, the isosurface component at the center value of symmetry occupies the entire volume domain. This type of dataset is often difficult to visualize because we cannot discriminate interesting features such as maxima and minima that invoke the isosurface appearance and disappearance. However, our framework can effectively alleviate this problem using the integral of the isosurface-trajectory distance function described in Section 4.2, because this topological attribute successfully identifies the center of isosurface trajectory. This leads us to the idea for clearly illuminating the important interior features by lowering the opacity of the voxels belonging to the occluding isosurface.

For example, as shown in Figure 5, the High Potential Iron Protein (HIPIP) dataset has a symmetric wave function with respect to the scalar field, and thus the isosurface component around the center value of the isosurface trajectory covers up the entire volume. Figure 5(a) shows the VST of the HIPIP dataset, and Figure 5(b) shows a visualization result obtained using the topologically-accentuated 1D opacity TF. As seen in Figure 5(a), the VST is almost symmetric and it has many critical points around the mean scalar field value 127. However, the corresponding isosurface component actually occludes many significant features as shown in Figure 5(b) if we assign a large opacity value to each representative field value uniquely. This observation lets us improve the result as shown in Figure 5(c) by lowering the opacity values of the voxels that have the small integral values. Indeed, this allows us to eliminate the occluding isosurface component from the important structures inside the volume. Note that since most voxels have the scalar field values and integral values near the dashed line overlaid in the TF definition in Figure 5(c), the opacity value gets substantially higher as the isosurface components shrink with scalar field value less than the mean 127.

5.4. Accentuating specific isosurface genera

The change in genus of each component of an isosurface may provide an important clue which allows us to visually understand the complexity of the structures embedded in a volume dataset. For example, Figure 6 visualizes the half domain of positive charge distribution simulated around two ^{16}O nucleons [Mei], where the other nucleon is located above the visualized domain.

Figure 6(a) shows the VST for this dataset, where the number on the left side of each link represents its genus. From the VST, we can see that two isosurface components corresponding to the links $\overline{P_6P_9}$ and $\overline{P_7P_8}$ are included by the outer isosurface component. By assigning higher opacity values to the voxels on the two included isosurface components, we can depict the included isosurface structures in an accentuated manner. The visualization result in Figure 6(b) certainly provides useful information for us to understand the nested structure, though the image cannot be said to pro-



(a) (b) (c)

Figure 5: *Visualizing the HIPIP dataset: (a) The corresponding VST, (b) with topologically-accentuated 1D opacity TF, and (c) with 2D opacity TF depending also on the integral of the isosurface-trajectory distance function.*

vide sufficient information for us to realize the complex interaction between the two nucleons. This motivates us to use the genus of isosurface component instead of the inclusion level as a topological attribute to design the new 2D TF that is intended to accentuate the regions topologically equivalent to a torus. A resultant image rendered with the new 2D TF is shown in Figure 6(c), where voxels which belong to isosurface components of genus 1 (corresponding to the links $\overline{P_2P_4}$ and $\overline{P_4P_5}$) are emphasized. In fact, the region topologically equivalent to a torus coincides with the subspace having complex interactions between the two nucleons, and attracts much attention from the observers. Furthermore, the visualization result pinpoints the locations where the change in genus is invoked, and provides the observers with important visual cues about the detailed spatial configuration of each of the ^{16}O nucleons.

6. Conclusion

In this paper, we proposed a set of topological attributes derived from the level-set graph of a given volume, and presented the basic design principles of multi-dimensional TFs depending on these attributes.

Since the present topological attributes can encapsulate the global structure as well as local features of the volume data, they make it much easier for observers to express their objectives than using the traditional local feature-based TFs, and thus resulting in comprehensible images so as to richly convey the structures embedded in the data. The feasibility of the present framework was proven empirically with applications to several scientific simulation datasets.

Remaining issues for our future research are as follows. First is to enrich the set of topological attributes towards more powerful analysis not only for real simulated datasets but for measured ones. Second is to identify the mutual relationships between the topological attributes and the traditional local features such as gradients and curvatures for realizing more advanced visualization operations. The last is to develop an intuitive tool for assisting TF design, where the observer's objectives are semi-automatically interpreted to suggest the best multi-dimensional TFs involving proper topological attributes.

Acknowledgements

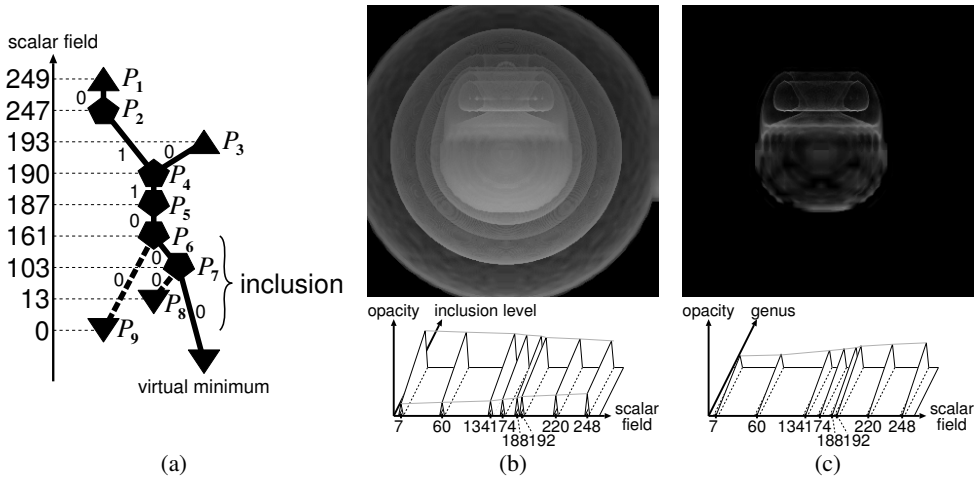
The authors would like to thank Prof. Hitoshi Sakagami at University of Hyogo to let us use the implosion datasets in Section 5.2 and provide many valuable comments. This work has been partially supported by Japan Society of the Promotion of Science under Grants-in-Aid for Young Scientists (B) No. 14780189, and the Ohkawa Foundation.

References

[BPS97] BAJAJ C. L., PASCUCCI V., SCHIKORE D. R.: The contour spectrum. In *Proc. IEEE Visualization '97* (1997), pp. 167–173.

[CSvdP04] CARR H., SNOEYINK J., VAN DE PANNE M.: Simplifying flexible isosurfaces using local geometric measures. In *Proc. IEEE Visualization 2004* (2004), pp. 497–504.

[FAT99] FUJISHIRO I., AZUMA T., TAKESHIMA Y.: Automating transfer function design for comprehensive rendering based on 3d field topology analysis. In *Proc. IEEE Visualization '99* (1999), pp. 467–470, 563.



(a) (b) (c)

Figure 6: *Visualizing the ^{16}O nucleon dataset: (a) The corresponding VST, (b) with topologically-accentuated 1D opacity TF, and (c) with 2D opacity TF depending also on the isosurface genus.*

[FATT00] FUJISHIRO I., AZUMA T., TAKESHIMA Y., TAKAHASHI S.: Volume data mining using 3D field topology analysis. *IEEE CG&A 20*, 5 (2000), 46–51.

[FTTY02] FUJISHIRO I., TAKESHIMA Y., TAKAHASHI S., YAMAGUCHI Y.: Topologically-accentuated volume rendering. In *Data Visualization: The State of the Art*, Post F. H., Nielson G. M.,, Bonneau G.-P., (Eds.). Kluwer Academic Publishes, 2002, pp. 95–108.

[HKG00] HLADŮVKA J., KÖNIG A., GRÖLLER E.: Curvature-based transfer functions for direct volume rendering. In *Proc. Spring Conference on Computer Graphics 2000* (2000), pp. 58–65.

[HSKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T. L.: Topology matching for fully automatic similarity estimation of 3D shapes. *Computer Graphics (Proc. SIGGRAPH 2001)* (2001), 203–212.

[KASS91] KOCHEVAR P., AHMED Z., SHADE J., SHARP C.: Cooperative, computer-aided design of scientific visualization. In *Proc. IEEE Visualization '91* (1991), pp. 306–313.

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

[KKH02] KNISS J., KINDLMANN G., HANSEN C.: Multidimensional transfer functions for interactive volume rendering. *IEEE TVCG 8*, 3 (2002), 270–285.

[KWTM03] KINDLMANN G., WHITAKER R., TASDIZEN T., MÖLLER T.: Curvature-based transfer function for direct volume rendering: Methods and applications. In *Proc. IEEE Visualization 2003* (2003), pp. 513–520.

[Lev88] LEVOY M.: Display of surfaces from volume data. *IEEE CG&A 8*, 5 (1988), 29–27.

[Mei] MEISSNER M.: `[http://www.volvis.org/]`.

[OFCD01] OSADA R., FUNKHOUSER T., CHAZELLE B., DOBKIN D.: Matching 3d models with shape distribution. In *Proc. Shape Modeling International 2001* (2001), pp. 154–166.

[SMSY02] SAKAGAMI H., MURAI H., SEO Y., YOKOKAWA M.: 14.9 tflops three-dimensional fluid simulation for fusion science with HPF on the Earth Simulator. In *Proc. Supercomputing 2002* (2002).

[TLM01] TENGINAKAI S., LEE J., MACHIRAJU R.: Salient iso-surface detection with model-independent statistical signatures. In *Proc. IEEE Visualization 2001* (2001), pp. 231–238.

[TNTF04] TAKAHASHI S., NIELSON G. M., TAKESHIMA Y., FUJISHIRO I.: Topological volume skeletonization using adaptive tetrahedralization. In *Proc. Geometric Modeling and Processing 2004* (2004), pp. 227–236.

[TTF04] TAKAHASHI S., TAKESHIMA Y., FUJISHIRO I.: Topological volume skeletonization and its application to transfer function design. *Graphical Models 66*, 1 (2004), 24–49.

[TTFN05] TAKAHASHI S., TAKESHIMA Y., FUJISHIRO I., NIELSON G. M.: Emphasizing isosurface embeddings in direct volume rendering. To appear in a book for Dagstuhl Seminar on Scientific Visualization 2003, 2005.

[WSHH02] WEBER G. H., SCHEUERMANN G., HAGEN H., HAMANN B.: Exploring scalar fields using critical isovalues. In *Proc. IEEE Visualization 2002* (2002), pp. 171–178.