

Topological Volume Skeletonization Using Adaptive Tetrahedralization

Shigeo Takahashi

Graduate School of Arts and Sciences
The University of Tokyo, Tokyo, Japan
takahashis@acm.org

Yuriko Takeshima

Graduate School of Humanities and Sciences
Ochanimizu University, Tokyo, Japan
yuriko@imv.is.ocha.ac.jp

Gregory M. Nielson

Department of Computer Science and Engineering
Arizona State University, Tempe, USA
nielson@asu.edu

Issei Fujishiro

Graduate School of Humanities and Sciences
Ochanimizu University, Tokyo, Japan
fuji@is.ocha.ac.jp

Abstract

Topological volume skeletons represent level-set graphs of 3D scalar fields, and have recently become crucial to visualizing the global isosurface transitions in the volume. However, it is still a time-consuming task to extract them especially when input volumes are large-scale data and/or prone to small-amplitude noise. This paper presents an efficient method for accelerating the computation of such skeletons using adaptive tetrahedralization. The present tetrahedralization is a top-down approach to linear interpolation of the scalar fields in that it selects tetrahedra to be subdivided adaptively using several criteria. As the criteria, the method employs a topological criterion as well as a geometric one in order to pursue all the topological isosurface transitions that may contribute to the global skeleton of the volume. The tetrahedralization also allows us to avoid unnecessary tracking of minor degenerate features that hide the global skeleton. Experimental results are included to demonstrate that the present method smoothes out the original scalar fields effectively without missing any significant topological features.

1 Introduction

Direct volume rendering is a powerful tool for visualizing complicated inner structures in a volume, and thus helpful to computer-aided modeling and testing as well as medical diagnosis and scientific simulation. However, it still requires significant features to be identified so that it can emphasize them individually in the final visualization images. As the key to the analysis of such significant features, *topological volume skeletons*, which are *level-set graphs* of 3D scalar fields, have recently received much attention in

the fields of volume visualization and computational geometry. This is because the level-set graph tracks topological transitions of isosurface components according to the scalar field, and thus serves as a landmark for exploring underlying inner structures. Examples can be found in [1, 13, 14].

Although the level-set graph is helpful, its complexity depends on the number of *critical points* that invoke topological changes of isosurfaces. This means that the level-set graph may become too complicated if the resolution of the input dataset exceeds some limit because it represents the critical points as its nodes. The level-set graph can also capture minor features such as *degenerate* critical points, which arise from the object interiors due to the small-amplitude noise or zero-gradient scalar fields. This is more likely to occur if the scalar field values are quantized to a small number of bits because scalar fields of small gradients are reduced to stepwise scalar fields in this case. While Takahashi et al. [13] presented a method for simplifying the complicated level-set graph to distinguish its global structure, the method still requires considerable computation time to extract an initial level-set graph if the input dataset is too complicated.

This paper presents a fast and robust method for computing the topological volume skeletons by introducing an adaptive tetrahedralization stage prior to tracking the skeletons. The present method introduces a top-down approach to adaptive tetrahedralization, because we have to extract the global topological skeleton of the entire volume without adding unnecessary tetrahedra for an appropriate interpolation of the scalar field. For selecting tetrahedra to be subdivided, the method employs a criterion that takes into account topological errors, in addition to the conventional criterion based on geometric errors. These criteria generate an interpolation of the 3D scalar field in such a way that we can track all the necessary topological features to constitute the

global skeleton of the volume. Furthermore, the adaptive subdivision scheme also prevents us from worrying about the minor degenerate critical points that have little influence on the underlying global skeleton. This is accomplished by assigning larger tetrahedra to small-amplitude noise and zero-gradient scalar fields for the approximation. Experimental results are demonstrated to show that the present method generates a smooth interpolation of a 3D scalar field while preserving its global topological skeleton.

This paper is organized as follows: Section 2 describes an algorithm for extracting the topological volume skeletons and mentions the requirements for our adaptive tetrahedralization scheme. Section 3 presents the details of the adaptive tetrahedralization method employed in our framework. After demonstrating several experimental results together with the feasibility of the present method in Section 4, Section 5 concludes this paper and refers to future work.

2 Topological Volume Skeletonization

The level-set graphs of 3D scalar fields were first introduced to the visualization community by Bajaj et al. [1]. Actually, they developed a fast algorithm for extracting level-set graphs called the *contour trees* (CTs) [15] that track the change in the number of connected components of isosurfaces. This algorithm was further extended to objects of any dimension by Carr et al. [3] in such a way that it has $O(n \log n + t\alpha(t))$ time complexity. Here, the algorithm is based on the assumption that all the volume cells of the input dataset are linearly interpolated by tetrahedralization, and n and t denote the numbers of vertices and tetrahedra there, respectively. One of the problems with the CTs is that the original CTs cannot represent the topological type (i.e. genus) of an isosurface. However, this problem has recently been solved by Pascucci et al. [10], where they calculate the changes in the Euler characteristics of isosurfaces.

Our algorithm first extracts a topological volume skeleton including relatively insignificant features, and then simplifies the skeleton to obtain the underlying global structure by analyzing the skeleton itself. In fact, the skeletonization algorithm we use here is newly developed by combining the algorithms of Carr et al. [3], Pascucci et al. [10] and Takahashi et al. [13]. The remainder of this section describes each step of the skeletonization algorithm. The tetrahedralization step prior to this skeletonization step will be described in Section 3.

2.1 Constructing Join and Split Trees

Our topological volume skeletonization begins with constructing CTs by tracking the change in the number of connected isosurface components as the scalar field value re-

duces. For this purpose, we use the algorithm of Carr et al. [3] in order to construct two graphs individually, which are the *join tree* (JT) that represents the appearance and merging of isosurface components as the scalar field value decreases, and *split tree* (ST) that represents the disappearance and splitting of isosurface components. Suppose that we have already generated a linear interpolation of the input 3D scalar field by using tetrahedralization. As shown in Figures 1(a) and (b), the JT and ST have voxels as nodes if they serve as vertices in the tetrahedralization. Note that, throughout this paper, the nodes of the graph are arranged from top to bottom according to the scalar field values. Since the JT and ST are dual if we reverse the axis of the scalar field, we will consider how to construct the JT only.

Before constructing the JT, the list of voxels associated with the tetrahedralization is sorted in a descending order according to the scalar field. If two voxels have the same scalar field value, they are compared according to the given indices to make the total ordering of the voxels. The first voxel is then removed from the list and added to the JT, which can be described as follows.

Suppose that n is the first voxel we have just pulled out of the list. From a set of voxels adjacent to n in the tetrahedralization, we extract voxels that have larger scalar field values than n as $\{u_1, \dots, u_l\}$. Since the voxel u_i ($1 \leq i \leq l$) has been already handled earlier than n , it belongs to some connected component C_i of the existing JT. Let us denote the node having the smallest scalar field value in the connected component C_i by r_i . If r_i is identical with the node n itself, we can skip the current node u_i and turn our attention to the next neighboring voxel u_{i+1} . Otherwise, we connect r_i and n with a link in the JT. If the target node n has no adjacent nodes that are larger in the scalar field, it will be inserted to the existing JT independently as a new connected component. This process allows us to construct the JT as shown in Figure 1(a), which represents the appearance and merging of isosurface components when the scalar field value decreases. In the same way, our skeletonization algorithm constructs the ST as shown in Figure 1(b) to locate the disappearance and splitting of isosurface components.

2.2 Constructing Augmented Contour Trees

Our next step is to construct a graph called the *augmented contour tree* (ACT) as an earlier representation of the CT. This graph also contains all the voxels that are involved in the tetrahedralization as the JT and ST do.

The ACT is defined to be a graph that tracks the topological transitions of isosurface components while passing through all the voxels as the scalar field value decreases. Carr et al. [3] proved that the ACT can be constructed from the JT and ST because the JT captures the top ends and upward branches of the ACT while the ST keeps its bottom

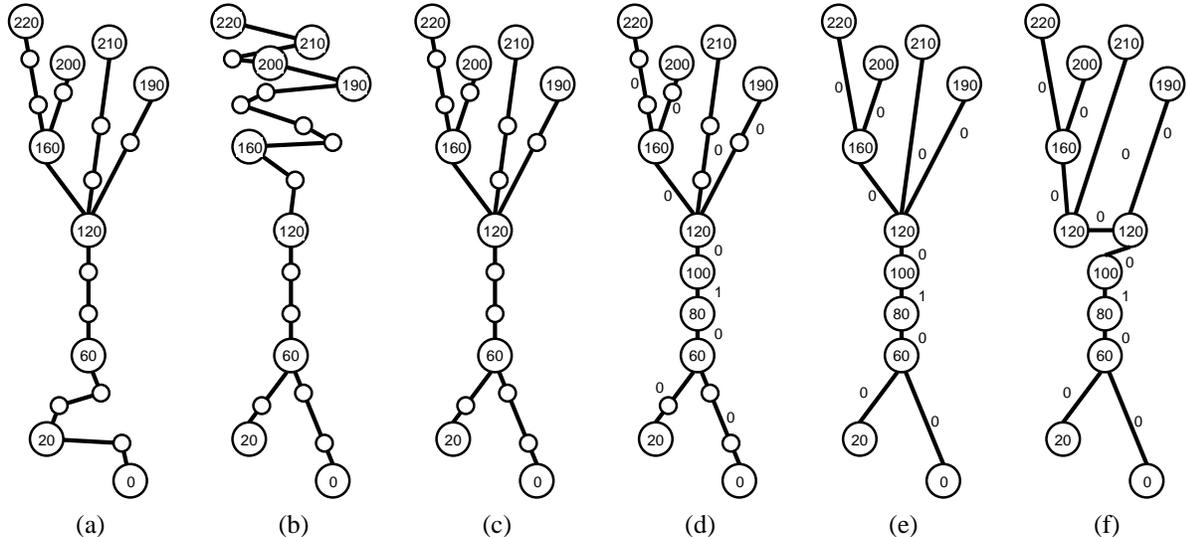


Figure 1. Steps for the skeletonization algorithm: (a) A join tree (JT), (b) a split tree (ST), (c) an augmented contour tree (ACT), (d) an augmented contour tree (ACT) with isosurface genera, (e) a contour tree (CT) with isosurface genera, and (f) a volume skeleton tree (VST).

ends and downward branches. Actually, our algorithm constructs the ACT by identifying its ends and branches from both its top and bottom while referring to the JT and ST. For example, suppose a node in the JT is a top end and its corresponding node in the ST has only one downward link. In this case, the node and its downward link in the JT is moved to the ACT, and the corresponding node in the ST and its incident links are removed. If the node in the ST has both upward and downward incident links, we connect its upper and lower adjacent nodes directly with a link in the ST. The same process can be carried out for the node that corresponds to a bottom end node in the ST and has only one upward link in the JT.

In this way, we can construct the ACT by reducing its undetermined part step by step because the end nodes in the JT and ST also stand for the end nodes in the undetermined part of the ACT. Figure 1(c) shows the final ACT constructed from the JT (Figure 1(a)) and ST (Figure 1(b)).

2.3 Extracting Changes in Isosurface Genus

So far we have constructed the ACT that tracks the change in the number of connected isosurface components. However, it is possible that we have critical points that invoke only the change in the isosurface topological type (i.e. genus) without changing the number of its connected components. Example include the transition from a sphere to a torus and also the reverse transition. This type of critical point is also very important when extracting the global

topological skeleton from the input volume.

Our algorithm extracts such critical points by taking advantage of the algorithm of Pascucci et al. [10]. In fact, their algorithm calculates the change in the Euler characteristic of isosurface components when they go through each voxel, by calculating the change in the Euler characteristic associated with simplices around the voxel. Here, the Euler characteristic χ is defined as

$$\chi = \#\{\text{vertices}\} - \#\{\text{edges}\} + \#\{\text{triangles}\} - \#\{\text{tetrahedra}\}, \quad (1)$$

where $\#\{X\}$ represents the number of X's. The change in the Euler characteristic of simplices around the voxel is calculated by finding its incident edges, triangles, and tetrahedra. Suppose that χ_l and χ_s are the Euler characteristics of the simplices, where the two values correspond to the isosurface components just before and after passing through the target voxel, respectively. To calculate χ_l , our algorithm counts simplices that have the target voxel as the vertex having the smallest scalar field value of all the corner vertices. The algorithm then applies Equation (1) for finding χ_l while setting $\#\{\text{vertices}\} = 1$. The other Euler characteristic χ_s is obtained by finding simplices where the target voxel is the largest in the scalar field. At last, the change in the Euler characteristic when each voxel is swept by the isosurface is calculated as $\chi_l - \chi_s$.

Indeed, this value allows us to detect the change in the genus of each isosurface component. For example, if the voxel causes the change in the Euler characteristic while

| C_3 | C_2 | | C_1 | | C_0 |
|---|---|---|---|---|---|
|  |  |  |  |  |  |
| C_3 | $3-C_2$ | $2-C_2$ | $3-C_1$ | $2-C_1$ | C_0 |
| (a) | (b) | (c) | (d) | (e) | (f) |

Figure 2. Connectivity of the critical points in the volume skeleton tree.

the number of isosurface components is left unchanged, it definitely affects the genus of the corresponding isosurface component. In this way, our algorithm can extract a specific type of critical point that changes only the genus of an isosurface component, as well as a critical point that changes the number of isosurface connected components. Figure 1(d) shows how the topological type (i.e. genus) of each isosurface component changes on the ACT.

2.4 Constructing the Contour Tree

Constructing the CT from the ACT just requires us to remove non-critical nodes from the ACT. Figure 1(e) shows an example of the CT, which is extracted from the ACT by removing the non-critical nodes represented by the small circles in Figure 1(d).

In this process, our algorithm transfers the non-critical nodes from the ACT to the remaining links in the CT. In other words, a link of the CT possesses a list of non-critical voxels that originally serve as nodes in the ACT. This helps us simplify a complicated volume skeleton for finding the global structure of the input volume, which will be described in Section 2.6.

2.5 Constructing the Volume Skeleton Tree

This step is devoted to finding a topological volume skeleton having only simple critical points, by resolving all the multiple critical points of the resultant CT into simple ones. In this paper, we call this type of level-set graph a *volume skeleton tree* (VST) [13]. The simple critical points of the VST have connectivities as shown in Figure 2, where each connectivity is classified according to the type and degree of the corresponding node.

Figure 2 suggests that isosurface transitions around simple critical points are classified into six types; appearance of a new isosurface component (Figure 2(a)), merging two isosurface components into one (Figure 2(b)), increment of the genus of an isosurface component (Figure 2(c)), splitting one isosurface component into two (Figure 2(d)), decrement of the genus of an isosurface component (Figure 2(e)), and

disappearance of an existing isosurface component (Figure 2(f)). Here, the subscript of the symbol C represents the number of negative eigenvalues of the Hessian matrix at the corresponding critical point. Since the critical points C_2 and C_1 have different degrees, we distinguish between them by indicating the degree of each critical node such as $3-C_2$, $2-C_2$, $3-C_1$, and $2-C_1$. According to this classification, we can obtain the VST from the CT by resolving a multiple critical point into simple ones. When resolving the multiple critical points, we assign an empty voxel list to a newly created link. For example, in Figure 1(e), the node at the scalar field value 120 has three upward links in the CT. This means that we can resolve this multiple node into two nodes as shown in Figure 1(f) because the multiplicity of the corresponding critical point is 2.

2.6 Simplifying the Volume Skeleton Tree

For analyzing the global topological structure from the input volume, our algorithm first extracts a VST in such a way that it still contains rather minor critical points. In fact, these minor critical points themselves are less important, but still necessary for simplifying the VST appropriately. This is because it is impossible to evaluate how each critical point contributes to the global structure only by looking at its local features. Note that while an adaptive tetrahedralization scheme is also introduced to reduce the complexity of the extracted VST, it only tries to eliminate degenerate critical points that have little influence on the global structure of the input volume and thus are completely negligible.

In our framework, by following the scheme presented in [13], we intend to reduce the complexity of the extracted VST until the VST becomes simple enough to express the underlying global structure of the volume. This is accomplished by assigning a weight value to each link of the VST and then removing the link having the smallest weight value one by one. Takahashi et al. [13] suggest that three patterns in the VST can be candidates for the removal as shown in Figure 3. In this figure, the third pattern can contain other critical points between the two end critical points while in the first and second patterns the two critical points are immediate neighbors.

As the weight value for each link, Takahashi et al. [13] used the value D that represents the difference in the scalar field between the end critical points of the link. However, this definition of weight values is sometimes unsuccessful in extracting important topological transitions of isosurfaces because the sizes of the corresponding isosurfaces are not taken into account. Instead of this, we define a new weight value that is given by

$$V \times D, \quad (2)$$

where V represents the volume swept by the isosurface

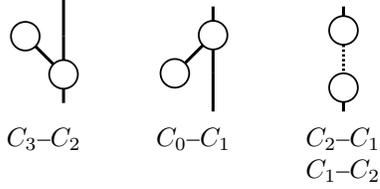


Figure 3. Candidate patterns to be removed from the volume skeleton tree in the simplification process.

component that corresponds to the target link. Note that the value V is equivalent to the size of the interval volume [4] bounded by the two isosurface components containing the end critical points. In our framework, the value V can be calculated easily because each link of the VST has a list of voxels assigned to it. It is clear that the volume of each voxel can be calculated as a quarter of the total volume of its incident tetrahedra because a tetrahedron is shared by four corner voxels. Now the swept volume V of the VST link is obtained by summing up the volumes of the voxels that belong to the link.

This formulation is fully justified because the new weight value $V \times D$ actually represents the size of the 4D subspace swept by the corresponding isosurface, which is contained in the entire 4D space spanned by the (x, y, z) -coordinates and scalar field. Our experiments show that this formulation of the weight value allows us to obtain the simplified version of the VST that adequately reflects the global behavior of isosurface transitions with respect to the scalar field.

It is noted that while simplifying the VST, our algorithm transfers the list of voxels from the removed link to one of its incident link that still remains in the VST. This makes precise extraction of the global volume skeleton because we properly take over the interval volume removed in the simplification process.

3 Adaptive Tetrahedralization

This section describes a method for *adaptive tetrahedralization* for linear interpolation of 3D scalar fields, which serves as an earlier stage for extracting topological volume skeletons. As described previously, the number of primitive tetrahedra becomes enormous if we subdivide each volume cell uniformly. Thus, our method reduces the number of tetrahedra by adjusting their sizes according to the local features of the 3D scalar fields. In addition, this enables robust extraction of topological volume skeletons even when the input volume contains high-frequency noise of small amplitude.

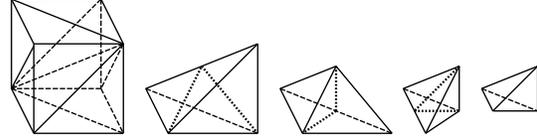


Figure 4. Tetrahedral decomposition rule in the bisection method.

Conventional adaptive tetrahedralization methods are roughly classified into two groups: *top-down approaches* that adaptively subdivide a large tetrahedron into small ones as the need arises, and *bottom-up approaches* that simplify tetrahedral subdivision by merging small tetrahedra into a large one. The top-down approaches include the *bisection method* [8] that subdivides a tetrahedron into two tetrahedra of equal size, and the *red-green method* [2] that introduces tetrahedra of two different shapes called “red” and “green.” Gerstner et al. [5] and Grosso et al. [6] proposed techniques for efficient isosurface extraction using the above two top-down methods, respectively. Moreover, Holliday et al. presented a method for generating smooth interpolation using the Coons volume [7], even when the tetrahedralization contains T-vertices, i.e., the inconsistency between the faces of adjacent tetrahedra. On the other hand, as the bottom-up methods, Zhou et al. [16] developed a method for merging tetrahedra having specific connectivity generated by the bisection rule, and Stadt et al. [11] devised a method for simplifying tetrahedra by contracting edges. In our implementation, we use the bisection method that is the simplest top-down approach introduced by Maubach [8], in order to assign fine tetrahedra adaptively to significant volume features.

3.1 The Bisection Method

Suppose that the input volume is decomposed into $N \times N \times N$ initial volume cells, each of which contains $(2^n + 1) \times (2^n + 1) \times (2^n + 1)$ voxels. This implies that the resolution of the input volume is $(2^n N + 1) \times (2^n N + 1) \times (2^n N + 1)$, while we often set $N = 1$. The bisection method first partitions each initial cell into six tetrahedra as shown on the left of Figure 4. Then each tetrahedron is further bisected if it cannot sufficiently approximate the 3D scalar field inside it according to some appropriate error criterion. Figure 4 shows how a tetrahedron is bisected in this method from left to right, where the longest edge is bisected each time.

This bisection rule requires that an edge we want to bisect must be the longest edge in every incident tetrahedron. If any of the incident tetrahedra does not satisfy this condition, we turn our attention to the longest edge of that inci-

dent tetrahedron. We then check again if this edge is also the longest of all the other incident tetrahedra. If all the incident tetrahedra share the edge as the longest one this time, we bisect the edge together with its incident tetrahedra and then go back to the previous edge. Otherwise, we further find incident tetrahedra that violate the condition and check their longest edges. In this way, the process of adaptive tetrahedralization terminates when all the tetrahedra satisfy the given error criterion.

3.2 Criteria for the Adaptive Tetrahedralization

To extract the topological volume skeleton correctly, we have to carefully formulate the criteria for estimating approximation errors of tetrahedra. For this purpose, this study introduces a topological error criterion as well as a conventional geometric one, so that the present method can capture the significant volume skeleton in early stages of the adaptive tetrahedralization.

3.2.1 Geometric Error Criterion

As the geometric error criterion, this method employs the root mean square error (RMSE) that is the most commonly used. The RMSE is estimated for each tetrahedron as follows. Suppose that a tetrahedron has a list of interior voxels that have the scalar field values $\{p_i\}$ ($i = 1, 2, \dots, m$). On the other hand, we can calculate the corresponding approximate scalar field values $\{q_i\}$ ($i = 1, 2, \dots, m$) by linearly interpolating the four corner voxels in the tetrahedron using the barycentric combination. Now the RMSE of this tetrahedron can be written as

$$\sqrt{\frac{\sum_{i=1}^m (p_i - q_i)^2}{m}}. \quad (3)$$

In this case, the method bisects tetrahedra where their RMSE errors exceed the given geometric error threshold, which helps us control the final adaptive tetrahedralization. In our implementation, this is possible because each tetrahedron possesses a list of interior voxels at the initial tetrahedralization stage so that the method refers to the list for estimating the corresponding RMSE. Furthermore, the interior voxels of each tetrahedron are accurately distributed to new tetrahedra when the original tetrahedron is bisected. This implementation conveniently prevents us from recalculating the list of interior voxels when the adaptive tetrahedralization is performed.

3.2.2 Topological Error Criterion

The geometric error criterion based on the RMSE allows us to assign smaller tetrahedra to 3D scalar fields of steep gradient that appear around the object boundary, and thus it

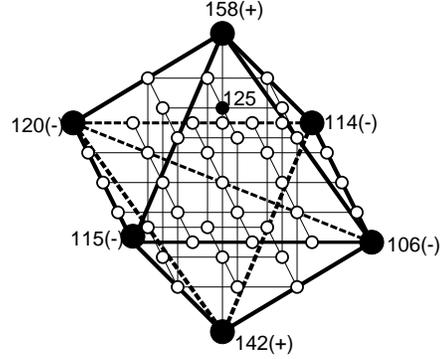


Figure 5. Topological error criteria: the interior voxel having the scalar field value 125 appears to be a critical point if it is adapted to the bisection point of the edge.

generally provides a sound interpolation that reflects the underlying volume features with the smaller number of tetrahedra. However, the criterion often requires more tetrahedra to capture the topological features that are indispensable for the simplification of volume skeletons because the previous geometric criterion only considers local geometric features. This motivates us to introduce a new topological criterion that brings global volume features more effectively to our tetrahedralization scheme.

The topological error criterion here has close relationships with that introduced by Gerstner et al. [5]. In fact, they check if critical points are contained in a set of tetrahedra that share the longest edge as shown in Figure 5. However, their goal differs from ours in that they try to control the adaptive tetrahedralization to keep the topological type (genus) of the target isosurface, while our goal is to extract the global evolution of the isosurfaces over the entire volume. In addition, their method first performs the bisection-based adaptive tetrahedralization using some geometric criterion as a preprocessing step, and constructs the hierarchies of critical points by traversing the refined tetrahedralization in a bottom-up manner. Conversely, our method introduces a topological criterion that aims at a top-down tetrahedralization approach while following the edge-based error estimation of Gerstner et al. For this purpose, we evaluate the importance of topological features associated with critical points as the topological errors, although Gerstner et al. only examine the existence of the critical points.

Our topological error criterion is formulated to simulate the weight values assigned to the links of the VST (See Section 2.6). For calculating the topological errors, we first find an edge and its associated tetrahedra as shown in Figure 5, where the edge is the longest in each tetrahedron and thus

can be bisected immediately. In this figure, the big black disks represent the voxels that constitute the corner vertices of the incident tetrahedra, and small disks represent the interior voxels of the tetrahedra. Actually, our topological error criterion tests if any of the interior voxels can serve as a critical point in the space defined by the incident tetrahedra.

Whether an interior voxel becomes a critical point or not can be determined by evaluating the difference in the scalar field from the corner voxels (i.e., black disks) in Figure 5 [5, 13]. In practice, we assign a sign “+” to the corner voxel if it has a larger scalar field value than the interior voxel, and a sign “-” if it has a smaller scalar field value. We then consider the boundary edges of the neighboring tetrahedra (except the interior edge to be split), and eliminate the edges if their endpoints has different signs. Finally, we count the number of connected components for each sign, and conclude that the interior voxel is critical if either of the two numbers differs from 1. Figure 5 shows a case where the voxel having the scalar field value 125 (represented by small black disk) becomes critical if it is employed as the interior voxel to be examined. In this case, the connected components of the corner vertices are {158} and {142} for the sign “+,” and {115, 106, 114, 120} for the sign “-.”

It is also necessary to measure the topological error if any of the interior voxels becomes a critical point, which indeed provides us with an effective criterion for the top-down approach to the adaptive tetrahedralization. In order to estimate the topological error, we first collect scalar field values of the corner voxels (black disks) and the interior voxel (white disk) we have just employed, and find the difference between the maximum and minimum scalar field values among them. The topological error is obtained by multiplying this difference and the volume of the space defined by the neighboring tetrahedra together, as shown in Figure 5. It follows from Equation (2) that the definition of these topological errors just approximates weight values assigned to the VST links that are affected by the space defined by the neighboring tetrahedra. Note that, if the topological error depends on the selection of interior voxels, the largest topological error is used to represent the final error. Moreover, this error criterion effectively avoids degenerate critical points because the degenerate critical points have little difference in the scalar field in most cases. In this way, this formulation enables the top-down approach to the adaptive tetrahedralization while tracking all the important topological features. In fact, as shown in Figure 6, the tetrahedralization with the topological error criterion reflects topological volume skeleton more accurately than that with the previous geometric criterion.

3.2.3 Hybrid Error Criteria

Although the tetrahedralization generated using the above topological error criterion undoubtedly represents significant features of the topological volume skeleton, it still contains unexpectedly discontinuities of the 3D scalar field unfortunately. In practical cases, the adaptive tetrahedralization needs to produce a smoother interpolation while preserving the significant topological features especially when it is used to visualize the underlying inner structures in the volume. For this purpose, our tetrahedralization scheme uses the hybrid error criteria that incorporate both the geometric and topological error criteria. In our implementation, the method first uses the topological error criterion to track the significant global structures of the input volume, and then applies the geometric error criterion to generate a smooth interpolation of the 3D scalar field.

4 Experimental Results

This section presents several experimental results to demonstrate the applicability of our method. Our prototype system has been implemented on a Linux-based PC system (CPU: Pentium IV 2.4GHz, RAM: 1GB).

Suppose a 3D scalar field represented by the following function:

$$\begin{aligned}
 f(x, y, z) = & 4c^2((x - R)^2 + (z - R)^2) \\
 & - ((x - R)^2 + y^2 + (z - R)^2 + c^2 - d^2)^2 \\
 & + 4c^2((x + R)^2 + (z + R)^2) \\
 & - ((x + R)^2 + y^2 + (z + R)^2 + c^2 - d^2)^2, \quad (4)
 \end{aligned}$$

where $c = 0.6$, $d = 0.5$, and $R = 0.2$. By taking samples of this function, a regular volume dataset of resolution $65 \times 65 \times 65$ was generated for our experiments. Figures 6(a), (b), and (c) show tetrahedralizations of this dataset and the corresponding volume skeletons where an ordinary uniform tetrahedralization scheme, an adaptive tetrahedralization scheme with the geometric criterion, and an adaptive tetrahedralization scheme with the topological criterion are applied, respectively. As shown in Figure 6(a), the ordinary uniform tetrahedralization produces a large number of minor critical points due to the discrete sampling and quantization whereas it can generate a smooth interpolation of the scalar field. Figure 6(b) exhibits a result obtained using an adaptive subdivision based on the geometric error criterion. Although the geometric criterion is helpful in generating a rather smooth interpolation, it offers incorrect topological features if the number of tetrahedra is limited as shown in the figure. To the contrary, as shown in Figure 6(c), the tetrahedralization and skeletonization algorithm with the topological criterion can extract a correct volume skeleton with a small number of tetrahedra. However,

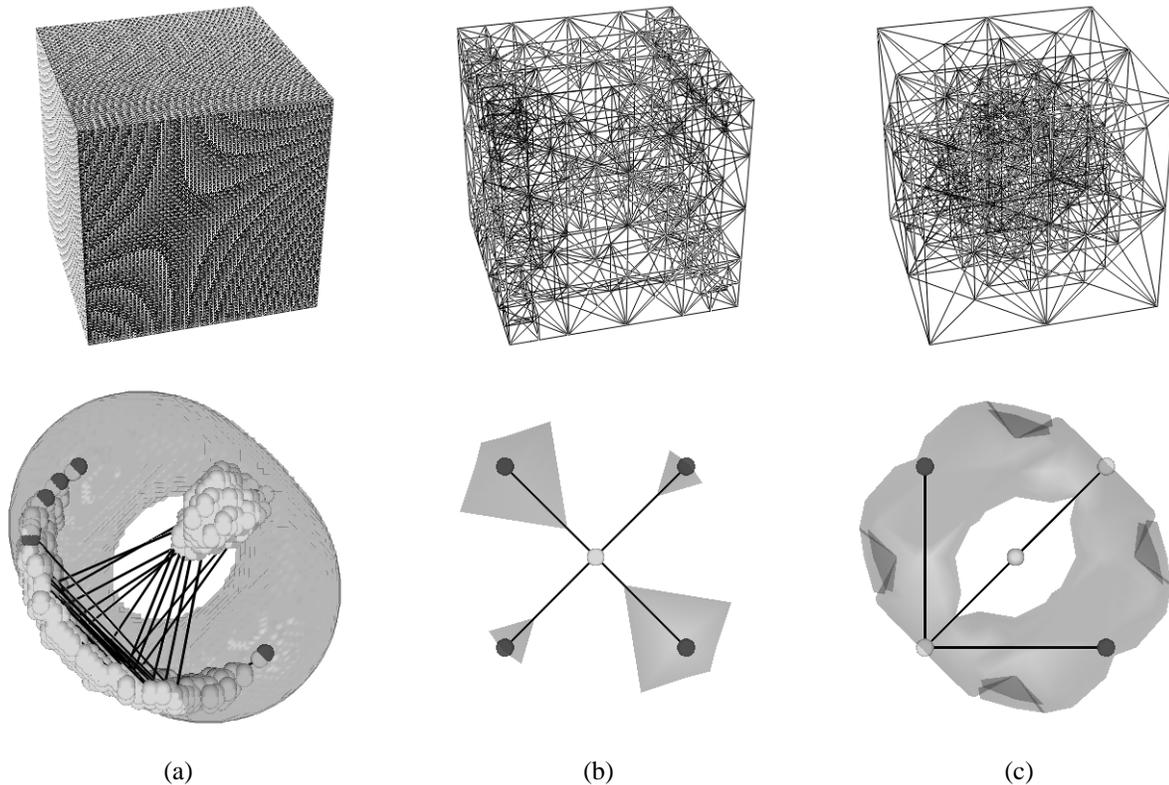


Figure 6. Tetrahedralizations and topological volume skeletons of a $65 \times 65 \times 65$ dataset generated from the volume function of Equation (4): (a) An ordinary uniform tetrahedralization is used (No. of tetrahedra: 1,572,864). (b) An adaptive tetrahedralization based on the geometric error criterion is used (No. of tetrahedra: 960). (c) An adaptive tetrahedralization based on the topological error criterion is used (No. of tetrahedra: 904).

it can only produce a poor interpolation of the 3D scalar field. It took 16 minutes to extract the topological volume skeleton for the case in Figure 6(a) while only 5 seconds (4 seconds for tetrahedralization and 1 second for skeletonization) for both Figures 6(b) and (c).

Figure 7 shows the nucleon dataset of resolution $41 \times 41 \times 41$ [9] where the two-body distribution probability of a nucleon in the atomic nucleus ^{16}O is simulated. Figures 7(a), (b), and (c) show an adaptive subdivision with approximately 10,000 tetrahedra, an adaptive subdivision with approximately 30,000 tetrahedra, and an ordinary uniform subdivision with 384,000 tetrahedra, respectively. These results are accompanied by the initial and simplified VSTs together with the corresponding final visualization results where the topological features are accentuated [13, 14]. Here, the hybrid error criteria are used to generate the adaptive subdivision. The tetrahedralization with approximately 10,000 tetrahedra in Figure 7(a) yields only a roughly ap-

proximated volume skeleton and thus the final visualization result is not satisfactorily refined. Nonetheless, the tetrahedralization only with approximately 30,000 tetrahedra allows us to generate excellent results without worrying about minor degenerate critical points as shown in Figure 7(b). Surprisingly, this visualization result can be matched to that in Figure 7(c), which is obtained using 384,000 tetrahedra. Note that when generating the visualization results on the right, we use the scalar field values assigned to the original regular volume datasets. Furthermore, we assign a new attribute value to each voxel so that we can take advantage of multi-dimensional transfer functions [14] to emphasize inner structures in the volume. Thanks to the adaptive tetrahedralization, our method only calculates the new attribute values for the voxels involved in the adaptive tetrahedralization, and then interpolates the values for other voxels using the barycentric coordinates. This actually accelerates the rendering process if the computational complexity for

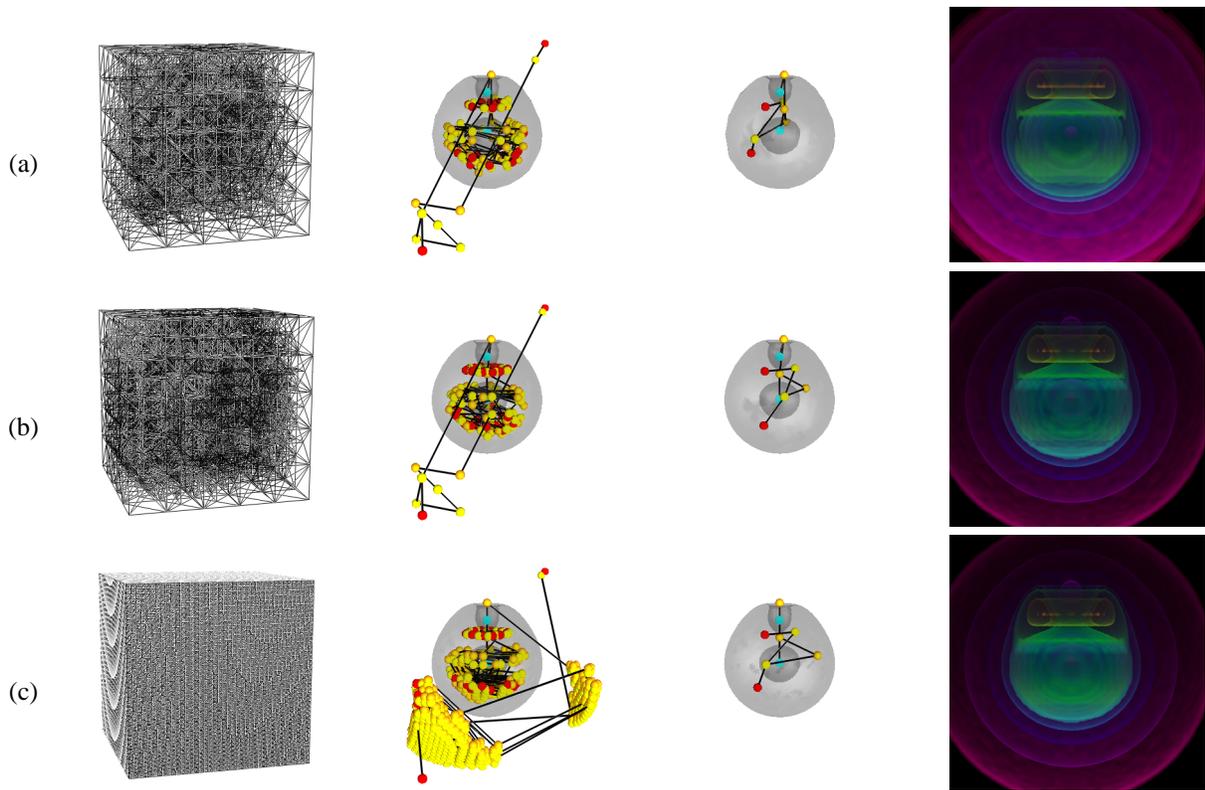


Figure 7. Effects of adaptive tetrahedralization for visualizing the nucleon dataset of resolution $41 \times 41 \times 41$: Tetrahedralizations, initial volume skeleton trees, simplified volume skeleton trees, and visualization results with topological features accentuated when (a) an adaptive subdivision with approximately 10,000 tetrahedra is used, (b) an adaptive subdivision with approximately 30,000 tetrahedra is used, and (c) an ordinary uniform subdivision with 384,000 tetrahedra is used.

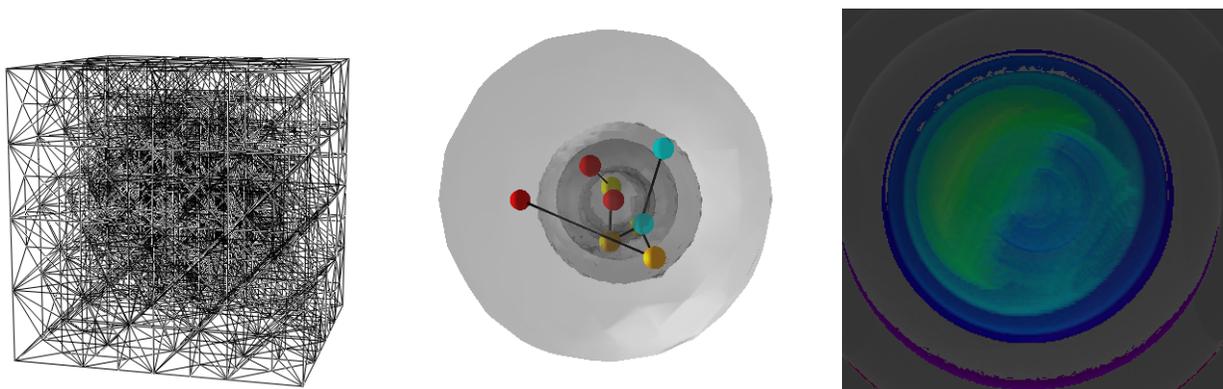


Figure 8. Visualization of the antiproton-hydrogen atom collision volume dataset with resolution $129 \times 129 \times 129$: (a) An adaptive subdivision where the number of tetrahedra is approximately 40,000, (b) the corresponding volume skeleton tree after the simplification, and (c) the final visualization result.

calculating the new attribute values is high. Our prototype system extracted the initial topological volume skeletons in 7 seconds (6 seconds for tetrahedralization and 1 second for skeletonization), 25 seconds (21 seconds for tetrahedralization and 4 seconds for skeletonization), and 96 seconds for Figures 7(a), (b), and (c), respectively. This implies that the overhead for the adaptive tetrahedralization is small enough to reduce the computation time of the entire process.

Figure 8 presents the visualization results of the volume dataset that is obtained by simulating the antiproton-hydrogen collision at intermediate collision energy below 50keV [12]. Since the resolution of this dataset is $129 \times 129 \times 129$, the ordinary uniform tetrahedralization scheme runs out of memory space on our computational environment. However, the present adaptive tetrahedralization scheme with the hybrid error criteria offers an interpolation as shown in Figure 8(a), which allows us to effectively extract the global volume skeleton from this dataset in 60 seconds (53 seconds for tetrahedralization and 7 seconds for skeletonization). Actually, the method successfully identifies the four-fold nested inclusion relationships of isosurfaces as shown in Figure 8(b), and thus emphasizes it in the final visualization image as shown in Figure 8(c).

5 Conclusion

This paper has presented an accelerated method for extracting topological volume skeletons using the adaptive tetrahedralization. The adaptive tetrahedralization scheme enables robust extraction of the volume skeletons by eliminating minor degenerate critical points arising from small-amplitude noise and zero-gradient scalar fields inside objects. In order to locate the significant volume features effectively, the topological error criterion as well as the geometric one was introduced to the adaptive subdivision scheme. Experimental results demonstrate that the present method considerably accelerates the topological volume skeletonization by offering an interpolation that reflects the significant topological features of the original 3D scalar fields.

Our future research topics include the automatic control of error thresholds for the adaptive tetrahedralization. Our experiments prove that the present method can successfully extract correct volume skeletons if it uses more than a certain number of tetrahedra to approximate the input 3D scalar field. Nevertheless, it may be possible to reduce the number of tetrahedra according to the property of the input dataset by relaxing the accuracy of the interpolation. We plan to formulate such a control by taking advantage of the frequency-based analysis of the input volumes.

Acknowledgements We wish to acknowledge the support of the Office of Naval Research (N00014-02-1-0287), the

National Science Foundation (NSF IIS-9980166 & ACI-0083609), DARPA (MDA972-00-1-0027), and the Japan Society of the Promotion of Science under Grants-in-Aid for Young Scientists (B) No. 14780189 and No. 15700081.

References

- [1] C. L. Bajaj, V. Pascucci, and D. R. Schikore. The contour spectrum. In *Proceedings of IEEE Visualization '97*, pages 167–173, 1997.
- [2] J. Bey. Tetrahedral grid refinement. *Computing*, 55(4):355–378, 1995.
- [3] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75–94, 2003.
- [4] I. Fujishiro, Y. Maeda, H. Sato, and Y. Takeshima. Volumetric data exploration using interval volume. *IEEE Transactions on Visualization and Computer Graphics*, 2(2):144–155, 1996.
- [5] T. Gerstner and R. Pajarola. Topology preserving and controlled topology simplifying multiresolution isosurface extraction. In *Proceedings of IEEE Visualization 2000*, pages 259–266, 2000.
- [6] R. Grosso, C. Lürig, and T. Ertl. The multilevel finite element method for adaptive mesh optimization and visualization of volume data. In *Proceedings of IEEE Visualization '97*, pages 387–394, 1997.
- [7] D. J. Holliday and G. M. Nielson. Progressive volume models for rectilinear data using tetrahedral coons volumes. In *Data Visualization (Proceedings of VisSym'00: Joint Eurographics - IEEE TCVG Symposium on Visualization)*, pages 83–92, 2000.
- [8] J. M. Maubach. Local bisection refinement for n -simplicial grids generated by reflection. *SIAM Journal of Scientific Computing*, 16(1):210–227, 1995.
- [9] M. Meißner. Web Page [<http://www.volvis.org/>].
- [10] V. Pascucci and K. Cole-McLaughlin. Efficient computation of the topology of level sets. In *Proceedings of IEEE Visualization 2002*, pages 187–194. IEEE Computer Society Press, 2002.
- [11] O. G. Staadt and M. H. Gross. Progressive tetrahedralizations. In *Proceedings of IEEE Visualization '98*, pages 397–403, 1998.
- [12] R. Suzuki, H. Sato, and M. Kimura. Antiproton-hydrogen atom collision at intermediate energy. *IEEE Computing in Science and Engineering*, 4(6):24–33, 2002.
- [13] S. Takahashi, Y. Takeshima, and I. Fujishiro. Topological volume skeletonization and its application to transfer function design. *Graphical Models*, 66(1), 2004. (to appear).
- [14] S. Takahashi, Y. Takeshima, I. Fujishiro, and G. M. Nielson. Emphasizing isosurface embeddings in direct volume rendering. submitted.
- [15] M. van Kreveld, R. van Oostrum, C. Bajaj, V. Pascucci, and D. Schikore. Contour trees and small seed sets for isosurface traversal. In *13th ACM Symposium on Computational Geometry*, pages 212–220, 1997.
- [16] Y. Zhou, B. Chen, and A. Kaufman. Multiresolution tetrahedral framework for visualizing regular volume data. In *Proceedings of IEEE Visualization '97*, pages 135–142, 1997.