Topology Control in Multiresolution Isosurface Extraction

Renato Pajarola, Thomas Gerstner

UCI-ICS Technical Report No. 05-01
Department of Computer Science
University of California, Irvine

January 2005

Topology Control in Multiresolution Isosurface Extraction

Renato Pajarola and Thomas Gerstner

[1] Computer Science Department
University of California Irvine
e-mail: pajarola@acm.org
[2] Institute for Numerical Simulation
University of Bonn

Abstract. Multiresolution modeling is a very important technique for interactive visualization of very large data sets. Multiresolution volume data structures allow the extraction of adaptively triangulated isosurfaces in large volume data sets at interactive frame rates, providing the user the ability to visualize the volume data in real-time. The isosurface is displayed at a coarse resolution for areas of low variance in the isovalue or for fast interaction, and at finer resolution for areas of high variability or detail inspections. Ideally, a coarse approximation of the isosurface has the same topological structure as in the highest resolution. Important topological properties such as the genus, or structure and number of connected components of an isosurface are often neglected in existing multiresolution isosurface extraction and rendering algorithms. This can result in sudden, and uncontrolled topological changes of the rendered isosurface whenever the global or local level-of-detail of the extracted isosurface changes. The scope of this paper is to propose an efficient technique which provides control over topological changes, in particular preservation of topolgical properties, in an interactive multiresolution isosurface extraction and rendering framework. We present two methods for topology preserving multiresolution isosurface extraction: the first is optimal in run-time and space cost, and the second is optimal with respect to the size of the extracted isosurface. Furthermore, we also present extensions of our method for controlled topology simplification.

Key words tetrahedral multiresolution modeling – isosurface extraction – level-of-detail – surface topology – critical points

1 Introduction

Isosurface extraction is a very common and useful tool for interactive visualization of volume data. In the last years, the resolution of volumetric data sets has been increasing dramatically. This applies to typical measurement data arising i.e. in medical imaging as well as to output of large-scale numerical simulations i.e. in scientific computing.

The importance of multiresolution methods becomes apparent when such large data sets have to be visualized interactively in applications such as computer aided surgery or scientific visualization. Without multiple *levels-of-detail* (LODs), the number of triangles which make up an isosurface can easily exceed the capacity that can be rendered at interactive frame rates. Furthermore, the isosurface extraction phase producing this large number of triangles may simply be too slow in order to be able to change the isovalue interactively.

Both, the isosurface extraction speed and rendering performance can significantly be increased by multiresolution methods. These methods allow the user to control the accuracy of the extracted isosurfaces by reducing the mesh complexity (or vice versa). Therefore, the user is able to explore and interactively display large volume data using simplified approximations of the isosurface to view the overall structure of the isosurface, or to adapt the mesh complexity to available rendering capacity. Moreover, the user can use more detailed resolutions when zooming into the data set for detail inspections, or to render single shot high-detail views. The user can also adjust the isovalue interactively at lower LODs.

Different LODs, and adaptive triangulation of the isosurface aid the user in navigating and exploring the data set interactively. However, the user should also be prevented from missing interesting parts of the isosurface, and the overall structure of the data set should correctly be retained. The topological genus (also number of handles), and the connected components – even the local spatial connectivity – of an isosurface are certainly important properties, and the user should have control over changes in these properties that are only due to simplification of the isosurface.

A multiresolution isosurface extraction algorithm is called *topology preserving* if all simplified isosurface approximations have the same topological genus, connected components, and local connectivity as the highest resolution isosurface. It is called *topology simplifying* if the genus, or the global or local connectivity change for different LODs. In the latter case, however, the user should be able to control the extent and type of possible topological changes.

Based on our preliminary results in [8], in this paper we include a more detailed treatment of topology and homeomorphism, and we present improved methods which allow preservation, and controlled simplification of topology of iso-

surfaces at different LODs in a real-time multiresolution isosurface extraction and rendering framework. Our techniques are based on hierarchical tetrahedral volume meshes instead of the triangle surface meshes of particular isosurfaces. Although the focus will here be on regular grid volume data, the approach can be generalized to curvilinear or unstructured grids generated by bisection in a straightforward manner.

The remainder of this paper is organized as follows. In Section 2 we review previous related work. Section 3 describes the hierarchical multiresolution mesh framework based on tetrahedral bisection (triangle bisection in 2D). In Section 4 we present our technique for topology preservation and include the mathematical background and references to the theory on topological spaces that forms the theoretical foundation of our approach. The controlled topology simplification method is described in Section 5. Section 6 concludes the paper.

2 Related Work

Very often, an isosurface represented as a triangle mesh is extracted in a preprocessing step. The extraction can be very time-consuming when standard marching algorithms [18, 24] are used. Therefore, a variety of methods have been designed to speed up the extraction step [14, 17, 29], or to limit the extraction to the visible triangles [10, 16]. General topological problems in isosurface generation have been addressed in [4, 21, 27, 31].

One way to turn an isosurface into a multiresolution representation is to apply a triangle mesh simplification algorithm to the extracted isosurface triangle mesh. For a detailed overview of the large variety of available triangle mesh simplification methods see the recent surveys of [13, 25]. In this context, controlled topology simplification of isosurface meshes has been considered in [1, 12, 7]. The triangle mesh simplification approach has several disadvantages though. It requires initial extraction of the isosurface at the finest resolution, which makes interactive adjustments of the isovalue impossible, unless isosurfaces for all possible isovalues are extracted which in turn is highly memory-inefficient. Moreover, the construction of a multiresolution triangle mesh hierarchy is generally slow since it has to deal with non-gridded, irregular triangle mesh input.

On the other hand, it is possible to extract isosurfaces at multiple resolutions directly from the volume data. Thereby, a multiresolution hierarchy is not inferred on the triangulated isosurface itself, but on the underlying 3D volume data set. A coarser isosurface is simply defined as the isosurface of a less detailed approximation of the volume data. Isosurface extraction is then usually performed in a recursive top-down traversal of the multiresolution volume data structure, resulting in a hierarchical spatial subdivision of the volume data containing the isosurface, and an adaptive triangulation of that isosurface. Starting with an extremely coarse initial approximation, details are added in areas where an error indicator shows a large intolerable local error with respect to the data on the finest resolution. If the approximation error drops below a user-defined threshold, the algorithm stops the recursive refinement locally, and extracts the isosurface at the current LOD. The isosurface may not even have to be stored as a triangle mesh, but is directly rendered during the recursive traversal of the multiresolution hierarchy.

If the data domain is refined adaptively (i.e. not uniformly) it can happen that the extracted isosurface contains holes (cracks) due to non-matching tetrahedral subdivisions at transition zones between different LODs. The analogous problem exists in hierarchical triangle subdivisions of height-field data [15, 6, 23]. Different solutions have been devised for this problem, including remeshing [5, 11], point insertion [26], filling, adaptive projection [22], and saturation of the error indicator [23, 9, 10]. In the following, we introduce a simple and highly efficient way to incorporate topology control into adaptive multiresolution isosurface extraction based on the error saturation technique.

A preliminary short report on the basic idea of the techniques presented in this paper has previously been published in [8]. In this extended paper, we provide a more thorough treatment of the mathematical concepts underlying our approach. We introduce the multiresolution framework based on isoline extraction to better understand its extension to multiresolution isosurface extraction. Furthermore, we review the mathematical theory of topological spaces and homeomorphism in the planar domain and show how it extends into 3D. Additionally, we present improved critical interval saturation and topology preserving isosurface extraction, new metrics for controlled topology simplification, and additional experiments of the improved algorithms.

3 Multiresolution Framework

In this section we review preliminary techniques for the construction of multiresolution isosurfaces based on tetrahedral meshes. We first introduce the analogous triangle mesh based 2D equivalent of this technique that provides easy to understand examples in the planar domain of the same topology concepts as used in 3D.

3.1 Triangle Bisection

The two-dimensional analog to the gridded volume data is a grid-digital height-field as it is familiar from terrain rendering. The multiresolution triangle mesh is based on a quadtree-like hierarchy, and recursive longest edge bisection of triangles, also called restricted quadtree triangulation (see also [28, 15, 6, 23]). Figure 1 shows an example of recursive longest edge bisection starting with a rectangle that is initially composed of two triangles. The triangles are recursively refined by splitting their longest edge at its midpoint, and connecting the midpoint to the opposite vertex of the triangles incident upon the bisected edge. Note that given a uniform square grid of input points, and starting with the square bounding box triangulated by a diagonal, all subsequent midpoints of recursive longest edge bisections fall onto points of the input grid.

Adaptively refining the triangulation, however, can cause *cracks* or so called hanging nodes in the triangulation as shown in Figure 2 a) when triangle A is refined, but triangle B is not. This is possible for a given approximation error threshold ε if the *error indicator* η indicates that triangle B provides a sufficient approximation for the given error tolerance ($\eta(B) < \varepsilon$),

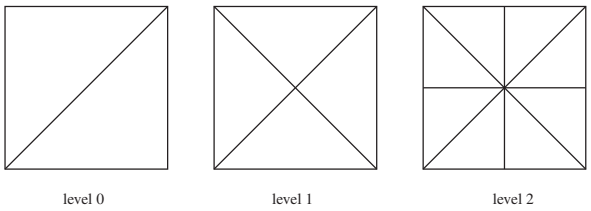

Fig. 1. Recursive refinement of triangles by longest-edge bisection.

but triangle A does not ($\eta(A) > \varepsilon$). To avoid cracks both triangles sharing a bisection edge must be refined, and must be on the same level of the refinement hierarchy as shown in [23]. This can be achieved by *saturating the error indicator* of triangles appropriately (see also Figure 2). The approximation error η of a triangle T is measured at the refinement vertex $x_{\text{ref}}(T)$, the midpoint of the longest edge $e_{\text{ref}}(T)$, and thus $\eta(T) = \eta(x_{\text{ref}}(T))$. Therefore, as shown in Figure 2 b) all triangles $\{T_l, T_r\}$ sharing the same refinement edge e_{ref} have the same error indicator $\eta(T_l) = \eta(T_r)$ since $x_{\text{ref}}(T_l) = x_{\text{ref}}(T_r)$.

The error indicator is saturated if and only if for each refinement edge, and its corresponding refinement vertex the following inequality holds for $\forall T \in \{T_l, T_r\}$:

$$\eta(T) \geq \max\{\eta(C_l(T)), \eta(C_r(T))\}. \tag{1}$$

If this condition is initially not satisfied for a particular error indicator, saturation can be achieved by error propagation in a bottom-up traversal of the triangle hierarchy. A minimally saturated error indicator $\bar{\eta}$ can be achieved by setting the error indicator on the refinement vertices to

$$\bar{\eta}(x_{\text{ref}}) = \max_{T \in \{T_l, T_r\}} \{\eta(x_{\text{ref}}), \bar{\eta}(C_l(T)), \bar{\eta}(C_r(T))\}, \tag{2}$$

and $\bar{\eta}(T) = \bar{\eta}(x_{\text{ref}}(T))$ correspondingly.

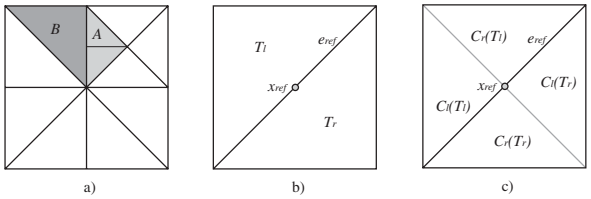

Fig. 2. a) Cracks occur if triangles sharing an edge are not equally subdivided at that edge. b) All triangles $\{T_l, T_r\}$ sharing the same longest edge have the same error indicator. c) The error indicator of a refinement vertex x_{ref} is saturated from all children of triangles $\{T_l, T_r\}$ sharing x_{ref}.

The extraction of an isoline for a particular isovalue τ is performed in a top-down traversal of the multiresolution triangle hierarchy. Given a geometric approximation error tolerance ε as stopping criterion, triangles are recursively subdivided until $\eta(T) < \varepsilon$, or all height values at the vertices of T are either smaller or bigger than τ. Finally, the isoline is computed locally in each triangle that contains the isovalue.

3.2 Tetrahedral Bisection

The multiresolution framework for 3D volume data is similar to the triangle mesh hierarchy described above, and is based on recursive bisection of tetrahedra. This refinement scheme is well known from numerical methods for partial differential equations [19], and it has recently be applied to multiresolution isosurface extraction in volume visualization [9, 30]. Octree approaches use a trilinear interpolation for modeling the volume data, and a bilinear interpolation for isosurface extraction – generating piecewise linear, triangular surface elements. In contrast, the data models of the volumetric grid, and the triangulated isosurface coincide in the tetrahedral volume mesh framework. Therefore, a lot of problems arising from different data models are eliminated. This also includes problems related to the topology of isosurfaces.

Let us consider a nested hierarchy of tetrahedral grids $\mathcal{T}^l$ with levels $0 \leq l \leq l_{\max}$. The tetrahedra are refined by *recursive bisection*: As shown in Figure 3, a tetrahedron T is split into two child tetrahedra $C_l(T)$ and $C_r(T)$ at the midpoint $x_{\text{ref}}(T)$ of the longest edge $e_{\text{ref}}(T)$ by the face spanned by $x_{\text{ref}}(T)$ and the two vertices of T opposite to $e_{\text{ref}}(T)$. Through recursive application of this subdivision, a binary tree hierarchy is inferred on the tetrahedra.

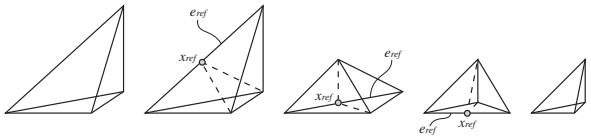

Fig. 3. Decomposition of a tetrahedron by recursive bisection.

We assume that the input volume data is arranged in a uniform grid with n^3 nodes, and $n = 2^k + 1$. The initial tetrahedral mesh $\mathcal{T}^0$ consists of the six tetrahedra whose vertices are adjacent corners of the bounding cube of the volume, and which all share the same diagonal of that cube as shown in Figure 6a. Thus, all refinement vertices $x_{\text{ref}}(T)$ fall onto grid points of the original data set.

Note that the tetrahedral mesh hierarchy does not need to be constructed explicitly. The binary tree of tetrahedra can be traversed implicitly, and all required information can be computed on-the-fly using a recursive depth-first traversal procedure as shown below. It is initially called for the six tetrahedra on the coarsest mesh $\mathcal{T}^0$ with their longest edge given by (x_1, x_2), and opposite vertices x_3 and x_4:

```
recursive_descent(Coord x1, x2, x3, x4, Int l)
{
  xref = (x1+x2) / 2;
  if (l < lmax) {
    if ((l mod 3) == 0) {
      recursive_descent(x1, x3, x4, xref, l+1);
      recursive_descent(x2, x4, x3, xref, l+1);
    } else {
      recursive_descent(x1, x3, x4, xref, l+1);
      recursive_descent(x2, x3, x4, xref, l+1);
    }
  }
}
```

Using linear interpolation inside each tetrahedron, a piecewise linear function on $\mathcal{T}^l$ is obtained that is uniquely defined by the data values $f(x_i)$ at the corresponding nodes of each tetrahedron.

3.3 Isosurface Extraction

The isosurface extraction algorithm for an isovalue τ traverses the tetrahedral hierarchy in a depth-first order. It recursively subdivides a tetrahedron $T = (x_1, x_2, x_3, x_4)$ if the data value interval of at least two nodes x_i and x_j of T includes the isovalue, i.e. $f(x_i) \leq \tau < f(x_j)$ (for $f(x_i) < f(x_j)$). The isosurface can then be extracted locally for each tetrahedron it intersects using the look-up table of the marching tetrahedra algorithm [24].

Given a geometric approximation error threshold ε, an adaptive multiresolution isosurface extraction algorithm additionally checks the error indicator $\eta(T)$ of a tetrahedron T before splitting T recursively. A tetrahedron T intersecting the isovalue τ is recursively subdivided only if $\eta(T) > \varepsilon$. Similar to the two-dimensional analog described in Section 3.1, adaptive subdivision can lead to hanging nodes in the tetrahedral mesh, and thus cracks in the extracted isosurface. If due to $\eta(T') < \varepsilon$ the recursive subdivision stops on a specific tetrahedron T', but since $\eta(T) > \varepsilon$ refines another tetrahedron T that shares its refinement edge $e_{\text{ref}}(T)$ with T', an inconsistency occurs at the hanging node $x_{\text{ref}}(T)$. Therefore, it is necessary to ensure that whenever a tetrahedron T is refined, all tetrahedra sharing the refinement edge $e_{\text{ref}}(T)$ are refined as well.

As shown in [9,10] cracks can be avoided by saturation of the error indicator η. First, let us define the error indicator $\eta(T)$ of a tetrahedron T on the refinement vertex $x_{\text{ref}}(T)$, the midpoint of its longest edge $e_{\text{ref}}(T)$, and thus $\eta(T) = \eta(x_{\text{ref}}(T))$. Therefore, all tetrahedra $\mathcal{T}(e_{\text{ref}}) = \{T \in \mathcal{T}^l | e_{\text{ref}}(T) = e_{\text{ref}}\}$ sharing the same refinement edge e_{ref} have the same error indicator.

Similar to Equation 1, the error indicator on the tetrahedral mesh fulfills the saturation condition if for all tetrahedra the following inequality holds.

$$\forall T \in \mathcal{T}(e_{\text{ref}}) : \eta(T) \geq \max\{\eta(C_l(T)), \eta(C_r(T))\} \qquad (3)$$

If the error indicator fulfills the saturation condition of Equation 3 for all $T \in \mathcal{T}^l$ with $l < l_{\text{max}}$, no hanging nodes can occur for all possible values of ε (see also [22,30]). If an error indicator η does not fulfill this condition it can be saturated in a preprocessing step. In a level-wise bottom up traversal of the hierarchy it is possible to construct a minimally saturated error indicator $\bar{\eta}$ by setting the error on the refinement vertices to

$$\bar{\eta}(x_{\text{ref}}) = \max_{T \in \mathcal{T}(e_{\text{ref}})} \{\eta(x_{\text{ref}}), \bar{\eta}(C_l(T)), \bar{\eta}(C_r(T))\}. \qquad (4)$$

Given a saturated error indicator η, the isosurface extraction algorithm for a given isovalue τ, and geometric approximation error threshold ε performs a depth-first traversal of the binary tetrahedra hierarchy. The recursive traversal is stopped if the current tetrahedron T is not a candidate for an intersection with the isosurface or if $\eta(T) < \varepsilon$, and in the latter case the isosurface is extracted locally for T. Therefore, the complexity of the multiresolution isosurface extraction algorithm is linear in the number of extracted triangles for a particular LOD, and independent of the input size.

4 Topology Preservation

The shapes of different LODs of an extracted isosurface for a particular isovalue τ strongly depend on the used error indicator η. A wide range of possible error indicators have been explored, and can be used to efficiently control tetrahedral grid refinement for visualization purposes [10]. However, the geometric error indicators in general do not allow to control or prevent changes in the topological structure of the isosurface. That is even the case if a conservative error measurement (i.e. based on the L_∞-norm) is used.

In this section we will first develop the notation of critical points which are extraordinary points in the data set that define the topological structure of an isosurface. And second, we derive a technique how these critical points can correctly be identified with respect to our tetrahedral multiresolution framework. Then we will show how the minimum set of critical points required to guarantee correct topology for arbitrary LOD ε can efficiently be selected for any isovalue τ. Finally, we will show experiments, and performance results of our topology preserving multiresolution isosurface extraction method.

4.1 Topology and Homeomorphism

In this section we want to provide the mathematical foundation and related work found in the theory on topological spaces, homeomorphism, and homotopy type that supports our approach on topology preservation (see also [20,3]). Let us first define the term *homeomorphism* that defines topological equivalence classes according to the theory of geometric topology that we want to follow here.

Definition 1. *(Homeomorphism) Two topological spaces X and Y are said to be homeomorphic or topologically equivalent if there exists a continuous bijection $f : X \to Y$ with a continuous inverse function.*

Homotopy type is also an equivalence relation between spaces, and it relates to the homeomorphism type of topological spaces through the following two lemmas found in homotopy theory.

Lemma 1. *Two spaces X and Y are homotopy equivalent if and only if X and Y are homeomorphic.*

Lemma 2. *If two spaces X and Y are not of the same homotopy type then they cannot be homeomorphic.*

The topological spaces of interest to us are (iso-) lines and surfaces. We will study the basics here for isolines, but the same principles hold analogously for isosurfaces, and thus define the topological space of a line.

Definition 2. *A topological line is a space S such that every $x \in S$ has a neighborhood homeomorphic to either $\mathbf{R}$ or $\mathbf{R}_{\geq 0}$.*

For example a circle and a square in 2D are topologically homeomorphic lines, however, a closed line of the form of an eight is not homeomorphic to a circle.

An isoline L_τ of a continuous function $f : \mathbf{R}^2 \rightarrow \mathbf{R}$ defining a compact manifold surface $M = \{p(x, y, f(x, y)) \in \mathbf{R}^3\}$ is the set of points $L_\tau = f^{-1}(\tau) = \{p \in M : f(p) = \tau\}$. Thus $L_\tau = f^{-1}(\tau)$ is a smooth sub-manifold of M. Note that we do not consider the boundary of the domain over which the function f is defined here, and we assume without loss of generality that the manifold M is compact (has no holes). See Figure 4 for an example of such an isoline.

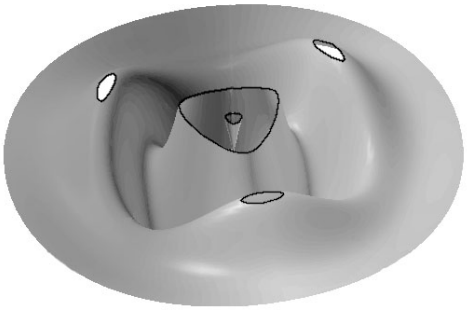

Fig. 4. Isoline L_τ for $\tau = 0.5$ of function $f = e^{-r^2}(\sin 2\pi r - r\cos 3\theta)$.

We now define the so called *critical points* and *critical values* that are used to detect changes in topology, or better in homotopy type, between different isolines.

Definition 3. *(Critical point) A point $p \in M$ is called a critical point of the function f if $\frac{\partial f}{\partial x} = 0$ and $\frac{\partial f}{\partial y} = 0$ for p. The value $\tau_p = f(p)$ is called a critical value of f.*

The following two lemmas derived from homotopy theory [20] make it clear that the topology of isolines only changes at critical points. We assume that critical points are isolated and non-degenerate.

Lemma 3. *Let $a < b$ and suppose that the set $f^{-1}[a, b] = \{p \in M : a \leq f(p) \leq b\}$ is compact, and contains no critical points of f. Then the two spaces $f^{-1}(a)$ and $f^{-1}(b)$ are of the same homotopy type.*

Lemma 4. *Let p be a critical point with value $\tau_p = f(p)$, and suppose that $f^{-1}[\tau_p - \varepsilon, \tau_p + \varepsilon]$ is compact for some small $\varepsilon > 0$. Then, the two spaces $f^{-1}(\tau_p - \varepsilon)$ and $f^{-1}(\tau_p + \varepsilon)$ are in different homotopy type equivalence classes.*

Due to the relation between homotopy type and homeomorphism expressed in Lemmas 1 and 2 and the effect of critical points on the homotopy type of isolines as described above one can conclude that:

Corollary 1. *Two isolines L_{τ_1} and L_{τ_2} with $\tau_1 < \tau_2$ are homeomorphic if there exists no critical point with critical value $\tau_p \in [\tau_1, \tau_2]$, but can be of different topological type if there exists such a critical point.*

Therefore, the preservation of critical points is a sufficient measure to control topological equivalence. Critical points cannot be omitted without careful consideration of the affected isolines and isovalues. In particular, all critical points that affect a particular isoline must be retained in a multiresolution environment. We will discuss the omission of non-critical and specific critical points in a discrete multiresolution setting in the following sections.

These fundamental properties of critical points with respect to the topology of isolines in 2D of functions $f : \mathbf{R}^2 \rightarrow \mathbf{R}$ also hold for isosurfaces $S_\tau = f^{-1}(\tau) = \{p \in \mathbf{R}^3 : f(p) = \tau\}$ of functions $f : \mathbf{R}^3 \rightarrow \mathbf{R}$. Thus the conclusion that the topology of an isosurface is defined by the critical points extends analogously into 3D.

4.2 Discrete Critical Points

The definition of critical points of continuous functions as outlined above, translates in a simple way to piecewise linear functions defined over a discrete regular grid. In 2D a critical point can form a local *minimum*, *maximum* or *saddle point*, and in 3D you have several additional variations of saddle points. For non-degenerate isolated critical points the relative difference in isovalues of the immediate grid neighbors can be examined to detect local minima, maxima or saddle points (i.e. around a local minimum all neighbors have larger values). At critical points, the isosurface locally changes its genus or number of components (connectivity) depending on if this data point is used in the piecewise linear interpolation of the isovalues or not. Since our data model is piecewise linear, critical points can only arise at vertices of the tetrahedral mesh. Note that this would not be true for octahedral meshes and trilinear interpolation.

Figure 5 shows a 2D example of a critical point at the refinement vertex x_{ref} of two triangles sharing the same refinement edge. Sample isovalues are given for the corners of the triangles and for the refinement vertex. Multiple isolines are drawn in light grey color according to the piecewise linear interpolation between vertices. With respect to the coarse triangulation on the left side, one can easily see the local maximum introduced by the refinement vertex on the finer triangulation on the right side. The critical point x_{ref} influences the local topology of a set of isolines.

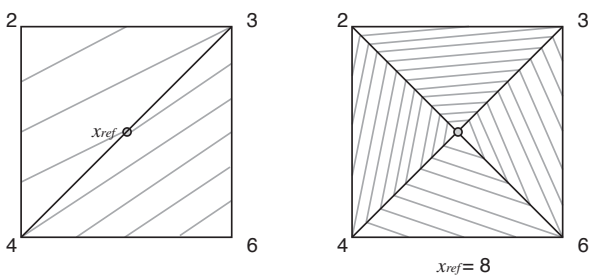

Fig. 5. Critical point in multiresolution isoline extraction. The corners have isovalues 2, 3, 4, and 6, and the refinement vertex x_{ref} has an isovalue of 8. Isolines are drawn in light grey at 0.5 intervals. Inclusion of x_{ref} equals to a local maximum at this critical point.

Similarly, a critical refinement vertex x_{ref} in a tetrahedral mesh defines the local topology of a set of isosurfaces with isovalues $\tau = f(x_{\text{ref}}) \pm \tau_\varepsilon$ (with τ_ε denoting the range of affected isosurfaces).

The basic idea to incorporate topology preservation into our multiresolution isosurface extraction method is the following:

> If an adaptive tetrahedral mesh contains all critical points, then any approximate isosurface has the same topology as the corresponding isosurface on the finest resolution.

Critical points, however, have to be defined hierarchically based on the tetrahedral bisection hierarchy since points which are non-critical with respect to the finest resolution can become critical with respect to a coarser resolution [2]. This is not because the removal of a non-critical point introduces any new singularities in the mesh not previously present. However, a single singularity such as a minimum, maximum or saddle point can have an effect on multiple levels of resolutions.

Therefore, a refinement vertex x_{ref} is called a *hierarchical critical point* if it is a critical point with respect to the refinement edge e_{ref} and the vertices of the incident tetrahedra in the current resolution. Thus x_{ref} is hierarchical critical if the topology of an isosurface restricted to all tetrahedra sharing the refinement edge e_{ref} changes when the tetrahedra are refined at x_{ref}. For topology preservation we can then conclude that:

Corollary 2. *If an adaptive tetrahedral mesh contains all hierarchical critical points, topology preservation has been achieved.*

4.3 Lookup-Tables

4.3.1 Critical Point Identification

Although hierarchical critical points can be found in a preprocessing step, their identification should nevertheless be done as efficiently as possible. Extraction of multiple isosurfaces for isovalues $\tau = f(x_{\text{ref}}) \pm \tau_\varepsilon$ with and without taking x_{ref} into account, and comparing their topology would be a tedious and too time-consuming task. Fortunately, since the topology of the isosurface only depends on the relative ordering of the data values at the vertices of the tetrahedra incident on the refinement edge, critical points can be identified quite efficiently based on look-up tables. In [30], however, an overly simplified and conservative table has been proposed which generates many unnecessarily refined tetrahedra. Our experiments have shown dramatically that a careful, and not overly conservative selection of critical points is crucial to achieve efficient isosurface simplification despite topology preservation.

To identify a critical point p of the function $f : \mathbf{R}^2 \rightarrow \mathbf{R}$ in 2D let us evaluate f on a circle C centered at p with sufficiently small radius ε. There exists a natural segmentation of C into k compact segments $C_1 \ldots C_k$, with each segment C_i having values $f(C_i)$ smaller or greater than $f(p)$. If this binary segmentation results in only one segment $C_1 = C$ then all points surrounding p have smaller (or greater) values than $f(p)$, and thus p is a local maximum (or minimum) of the function f. If the binary segmentation results in exactly $k = 2$ segments then the point p is not a critical point of f, and if $k > 2$ then we have a (critical) saddle point. Analogously in 3D we evaluate the function $f : \mathbf{R}^3 \rightarrow \mathbf{R}$ on an ε-sphere around p, and its binary segmentation leads to the same definition of a critical point as in 2D.

The examination of an ε-neighborhood around a point as outlined above can be translated to defining a hierarchical critical refinement vertex x_{ref} by examining the piecewise linear (triangulated) boundary surface of all tetrahedra incident on the refinement edge e_{ref}. First, let us define the *surrounding polyhedron* C_{poly} of a refinement edge e_{ref} as the boundary of all adjacent tetrahedra sharing e_{ref}. In our case of regular gridded data sets and recursive tetrahedral bisection, three types of polyhedra have to be considered as shown in Figure 6: a triangulated cube, an octahedron, and a diamond. The cube applies to tetrahedra of levels $(l \bmod 3) = 0$, the octahedron to levels $(l \bmod 3) = 1$, and the diamond to levels $(l \bmod 3) = 2$.

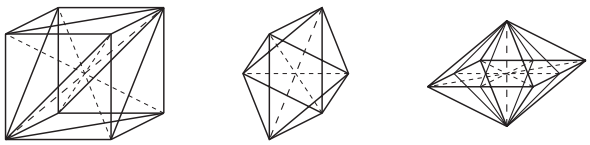

Fig. 6. The three types of surrounding polyhedra which arise during the refinement. Refinement edges are bold-dashed, spoke edges are light-dashed.

The m vertices of C_{poly} are marked with a $+$ sign if the data value at the vertex is larger than the value at the refinement vertex, and with a $-$ sign if it is smaller. The look-up table for each type of surrounding polyhedron consists of the 2^m possible cases, and contains a bit indicating if the refinement vertex is critical or not. Simplified look-up tables can be constructed by the identification of all symmetry classes for the $+/-$ bit-patterns of the different types of polyhedra and checking for criticality in each of these classes only. Figure 7 shows the resulting 25 classes for the cube. Not shown is case 1 where all vertices have the same sign which would lead to a local minimum or maximum which is always critical. The main symmetry classes for the cube are identical to the marching cubes look-up table [18], but special care has to be taken to include a few more subclasses. These subclasses occur because the endpoints of the refinement edge (from the front-bottom-left corner to the back-top-right corner of the cube) have different connectivity than the other vertices of C_{poly}. The images in Figure 7 show isosurfaces after refinement, inclusion of x_{ref}, for an isovalue of $\tau = f(x_{\text{ref}})$.

In general for a refinement edge e_{ref}, its refinement vertex x_{ref} is critical if two or more components of the isosurface with isovalue of exactly $\tau = f(x_{\text{ref}})$ have a non-manifold point connection at x_{ref}. For the cube in Figure 7 this is the case for the symmetry classes 4b, 5a, 5b, 7b, 8b, 12b, and 13. In all these cases, changing the isovalue slightly in positive or negative direction would lead to different connectivities of the isosurface components, and thus to local changes in the isosurface topology.

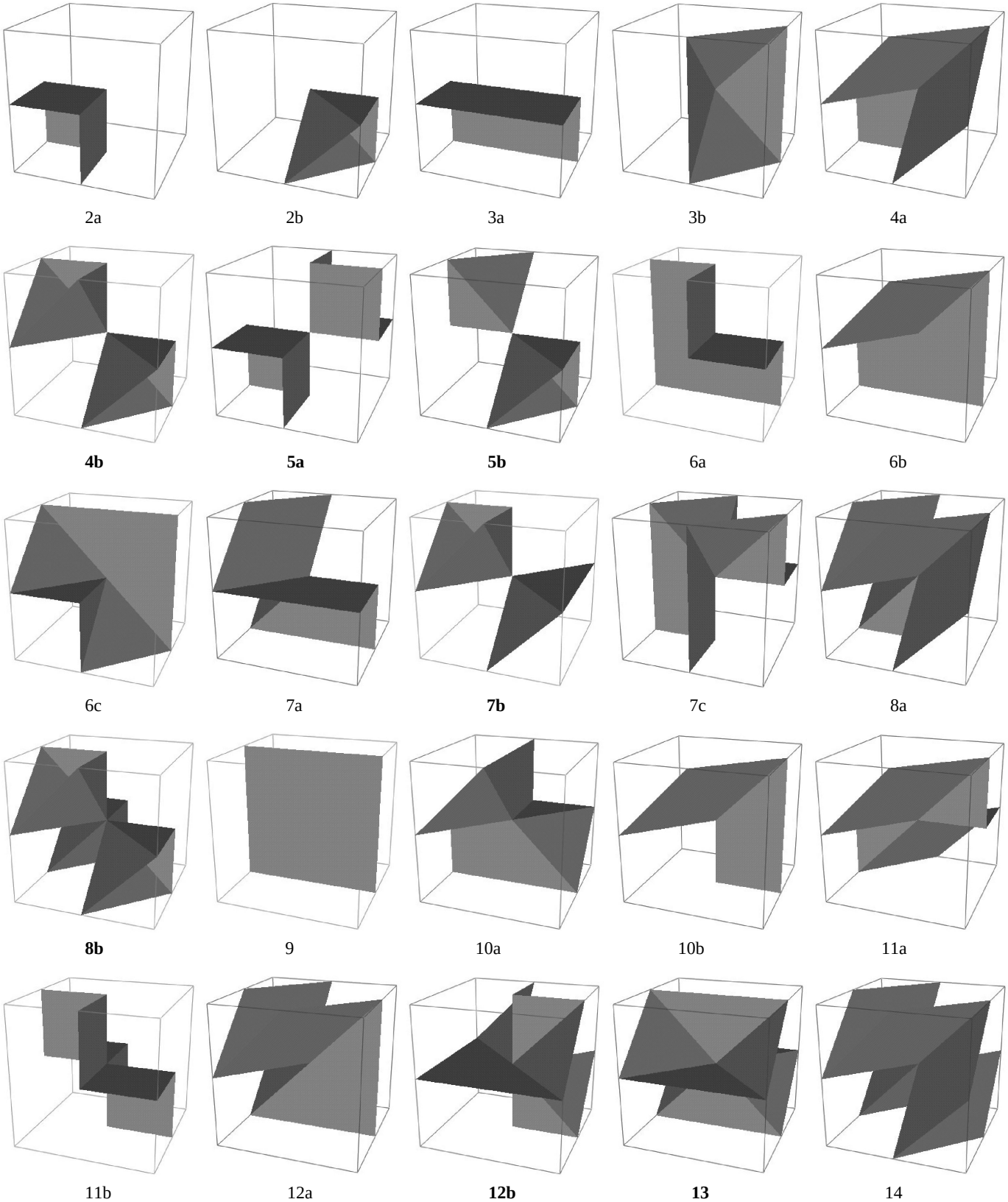

Fig. 7. Topological classes for the cube. Cases with critical refinement vertices are labeled in bold, all other situations are non- critical.

In the following section we show how the $\pm$-labeling of the vertices of C_{poly} can efficiently be exploited to compute the binary segmentation of the piecewise linear ε-neighborhood around the refinement vertex x_{ref}.

4.3.2 Automatic Construction of Look-up Tables

Manual construction of look-up tables based on the binary segmentation of the polyhedron surface is of course potentially erroneous since sub-cases may easily be forgotten or wrongly classified. However, there is an efficient and exact way to construct look-up tables for an automatic identification of critical points automatically based on the $\pm$-labeling of the surrounding polyhedron's vertices.

As described in the previous section, hierarchical critical refinement vertices can be identified by the binary segmentation of the piecewise linear ε-neighborhood, C_{poly}, around the refinement vertex. Local minima and maxima can easily be identified by the fact that all vertices of C_{poly} have the same $\pm$ label, thus the segmentation of C_{poly} results in one single segment. Other hierarchical critical points correspond to various saddle point configurations where the isosurface of isovalue $\tau = f(x_{\text{ref}})$ has multiple components that touch each other at the non-manifold point x_{ref} (see also cases 4b, 5a, 5b, 7b, 8b, 12b, and 13 in Figure 7). Such a hierarchical critical refinement vertex x_{ref} is characterized by a binary segmentation of C_{poly} that results in $k > 2$ different segments.

To compute the binary segmentation of C_{poly} let us first define the *edge graph* G_{edge} which consists of all vertices and edges of the triangulated surface of the surrounding polyhedron C_{poly} (triangulated cube, octahedron, or diamond) of a refinement edge. We furthermore label each node in G_{edge} with a $+$ or $-$ sign with respect to its isovalue being greater or smaller than the value $f(x_{\text{ref}})$ at the refinement vertex under consideration. Based on the labeling of G_{edge} we can determine if x_{ref} is critical or not. Basically we can determine from this graph into how many segments $C_1 \dots C_k$ the ε-neighborhood C_{poly} is partitioned. This can be done by computing the number of connected components of the graph G_{edge} which consist solely of either $+$ or $-$ labeled nodes. Therefore, the automatic identification of a hierarchical critical refinement vertex x_{ref} consists of the following four steps:

1. construct the edge graph G_{edge} of the surrounding polyhedron C_{poly} of x_{ref},
2. delete all edges from G_{edge} between a $+$ and a $-$ node,
3. count the remaining connected components of G_{edge},
4. if the number k of components is 1 or greater than 2, then the refinement vertex is critical.

If the number of components of the edge graph is $k = 1$ after Step 3 then the extracted isosurface $S_{\tau=f(x_{\text{ref}})}$ does not intersect with the the surrounding polyhedron, and the considered point x_{ref} represents a local minimum (if all labels are $+$) or maximum (if all labels are $-$), thus it is critical. If the number of components of G_{edge} is exactly $k = 2$ after removal of all $\pm$-edges from G_{edge}, then the isosurface $S_{\tau=f(x_{\text{ref}})}$ consists only of one connected component going through x_{ref}, and thus x_{ref} is not critical. In all other cases, number of connected edge graph components being greater than 2, multiple isosurface components of $S_{\tau=f(x_{\text{ref}})}$ meet exactly at x_{ref}, and thus x_{ref} is critical (it is a saddle point).

As an example, let us consider the case shown for the symmetry class 13 in Figure 7, and its edge labeling as shown in detail in Figure 8. The vertices of the cube are numbered front to back, bottom to top, left to right. This way, the signs of the vertices are given by $--+++--$.

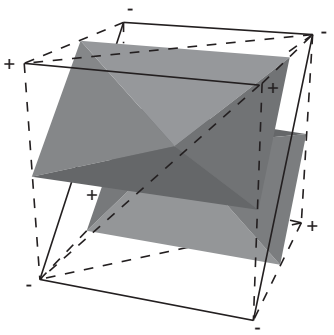

Fig. 8. Edge graph of the cube type of surrounding polyhedron with example $\pm$-labels indicated at the corners, and dashed edges between corners which have different labes.

For this example cube the steps of the automatic identification of critical points are the following:

1. The complete edge graph G_{edge} for the cube is given by the following adjacency matrix:

	$--+++--$
$-$	$\times$
$-$	$\times\times$
$+$	$\times\ \ \times$
$+$	$\times\times\times\times$
$+$	$\times\ \ \ \ \times$
$+$	$\times\times\ \ \ \ \times\times$
$-$	$\times\ \ \times\ \ \times\ \ \times$
$-$	$\ \ \times\times\times\times\times\times\times$

2. All $\pm$-edges shown as dashed edges in Figure 8 are removed from the adjacency table, leaving only:

	$--+++--$
$-$	$\times$
$-$	$\times\times$
$+$	$\ \ \ \times$
$+$	$\ \ \ \times\times$
$+$	$\ \ \ \ \ \times$
$+$	$\ \ \ \ \ \times\times$
$-$	$\times\ \ \ \ \ \ \times$
$-$	$\ \times\ \ \ \ \ \times\times$

3. To count the number of connected components of the remaining edge graph we compute the transitive hull of this graph (reachable nodes) expressed by the following reachability matrix:

	$--+++--$
$-$	$\times$
$-$	$\times\times$
$+$	$\ \ \ \times$
$+$	$\ \ \ \times\times$
$+$	$\ \ \ \ \ \times$
$+$	$\ \ \ \ \ \times\times$
$-$	$\times\times\ \ \ \ \ \times$
$-$	$\times\times\ \ \ \ \ \times\times$

4. The graph consists of 3 connected components: one consisting of all four $-$ nodes, and 2 components each consisting of two $+$ nodes. Since the number of components is 3 the point x_{ref} on the diagonal of the cube (front-bottom-left to back-top-right corners) is critical.

The total running time of the automatic look-up table generation is in the order of $O(2^m \cdot m^2)$ for each type of polyhedron with m nodes in the edge graph. Obviously this computation only has to be performed once as a preprocessing step for tetrahedral bisection on a regular grid. Furthermore, these look-up tables could also be stored in files for the three types of surrounding polyhedra. For recursive tetrahedral bisection refinement, critical points occur in 68 out of 256 cases for the cube, in 8 out of 64 cases for the octahedron, and in 400 out of 1024 cases for the diamond.

Critical points located on boundary of the data set do not need any special treatment. The existing lookup-tables can be used if missing vertices (respectively their signs) are mirrored across the boundary. This way, boundary critical points will also be classified correctly.

Note that this automatic construction of look-up tables is not restricted to regular grids, but can be applied to any irregular tetrahedral mesh generated by edge bisection as well.

4.4 Critical Intervals

A straightforward way to incorporate topology preservation into the multiresolution isosurface extraction algorithm is to set error indicator values at hierarchical critical points to infinity as proposed in [30]. However, for a particular isosurface of isovalue τ only a subset of all critical points is really relevant. Thus including all critical points in the isosurface extraction, neglecting the current isovalue τ, results in extensive refinements of tetrahedra that are not actually required to preserve the topology of the isosurface. In a multiresolution isosurface extraction method this solution is not tolerable due to the goal of minimizing the number of triangles to represent the isosurface.

Each hierarchical critical point is only critical for a limited range of isovalues, and not critical at all for all other isovalues. Therefore, every critical point will be assigned a *critical interval*. The critical interval is defined as the range of isovalues for which the isosurface changes topology when the tetrahedra are refined at the critical point. With this information, the extraction algorithm can selectively refine only at those critical points whose critical interval contains the current isovalue τ.

The range of isovalues for which a critical point x_{ref} is indeed critical is defined by the isovalues of the endpoints of its refinement edge e_{ref}. With respect to the 2D example in Figure 5 one can easily see that x_{ref} is only critical for isovalues $4 \leq \tau \leq 8$ (only within this range the isoline changes from having just one intersection with e_{ref} to having two intersections after refining at x_{ref}).

Given the endpoints x_1 and x_2 of e_{ref}, and their isovalues $f(x_1)$ and $f(x_2)$, the critical interval $I(x_{\text{ref}})$ of the refinement vertex only depends on the $\pm$-labels of x_1 and x_2. If both have $+$ labels, then the critical interval is given by

$$I(x_{\text{ref}}) = [f(x_{\text{ref}}), \min\{f(x_1), f(x_2)\}]\,. \tag{5}$$

If both endpoints x_1 and x_2 have $-$ labels, their isovalues $f(x_1)$ and $f(x_2)$ are both smaller than $f(x_{\text{ref}})$, then the critical interval is given by

$$I(x_{\text{ref}}) = [\max\{f(x_1), f(x_2)\}, f(x_{\text{ref}})]\,. \tag{6}$$

Note that refinement edges whose endpoints have different signs never lead to critical points.

4.5 Saturation of Critical Intervals

The critical intervals of refinement vertices are basically used as additional error indicators in the multiresolution isosurface extraction algorithm. As outlined in Section 3.3, tetrahedra are recursively refined until the geometric error indicator $\eta(T)$ is smaller than the tolerated approximation error ε. To preserve topology, tetrahedra are additionally recursively subdivided if the isovalue τ of the extracted isosurface is within the critical interval of the refinement vertex, $\tau \in I(x_{\text{ref}})$.

Considering the critical interval $I(x_{\text{ref}})$ as an additional error indicator as outlined above also requires correct saturation to prevent hanging nodes in the tetrahedral refinement, and cracks in the isosurface as explained in Section 3.3.

4.5.1 Minimal Saturation

In order to avoid missing critical points during the tree traversal, critical intervals $I(x_{\text{ref}})$ have to be saturated similarly to the geometric error indicator values $\eta(x_{\text{ref}})$. Each tetrahedron T is assigned a critical interval $I(T)$ which is defined as the critical interval of its refinement vertex $I(x_{\text{ref}}(T))$. The critical intervals as defined by Equations 5 and 6 generally do not satisfy the saturation condition as given for the geometric error indicator in Equation 3. However, in a level-wise bottom-up traversal of the tetrahedral bisection hierarchy it is possible to generate a *single minimally saturated* critical interval $\bar{I}(x_{\text{ref}}) = [\bar{I}_{lo}, \bar{I}_{hi}]$ by interval merging (taking the upper and lower bounds of the intervals of the current tetrahedron and its two children). Similarly to Equation 4 the critical intervals can be saturated using the following saturation:

$$\bar{I}_{hi} = \max_{T \in \mathcal{T}(e_{\text{ref}})} \{I_{hi}(x_{\text{ref}}), \bar{I}_{hi}(C_l(T)), \bar{I}_{hi}(C_r(T))\} \tag{7}$$

$$\bar{I}_{lo} = \min_{T \in \mathcal{T}(e_{\text{ref}})} \{I_{lo}(x_{\text{ref}}), \bar{I}_{lo}(C_l(T)), \bar{I}_{lo}(C_r(T))\} \tag{8}$$

Saturation of the critical interval according to Equations 7 and 8 has the advantage that in addition to the geometric error indicator $\bar{\eta}$ only two more values $\bar{I}_{lo}$ and $\bar{I}_{hi}$ have to be stored for each refinement vertex.

Using this saturation method, the multiresolution isosurface extraction algorithm subdivides a tetrahedron T recursively if $\bar{\eta}(x_{\text{ref}}(T)) > \varepsilon$, or if $\bar{I}_{lo}(x_{\text{ref}}(T)) \leq \tau \leq \bar{I}_{hi}(x_{\text{ref}}(T))$. We will refer to this method as the *minimally saturated* isosurface extraction.

4.5.2 Optimal Saturation

The minimally saturated critical intervals $\bar{I}(x_{\text{ref}})$ of vertices prevent the isosurface extraction algorithm from missing any critical points that would be required to preserve the topology for any arbitrary LOD of the isosurface. However, this method of critical interval saturation does not guarantee that only the minimal necessary set of critical points is selected.

The example in Figure 9 shows this analogous in the 2D situation. In this example, the two refinement vertices on the bottom and right edges of the square are both critical for two disjunct intervals, $\bar{I}_a = [1,3]$ and $\bar{I}_b = [4,7]$ respectively, and the other grid points around the square are all non-critical. The refinement vertex x_{ref} itself is not a hierarchical critical point either since its isovalue $f(x_{\text{ref}}) = 4$ is in between the isovalues of the endpoints of the refinement edge e_{ref}. Nevertheless, minimal saturation according to Equations 7 and 8 will result in a critical interval $\bar{I}(x_{\text{ref}}) = [1,7]$ at that refinement vertex. It is obvious that for some isovalues such as $\tau = 4$ there are no critical points in Figure 9. However, x_{ref} will be selected by the isosurface extraction algorithm unnecessarily since $\tau \in \bar{I}(x_{\text{ref}})$, even in the case where its saturated error indicator $\bar{\eta}(x_{\text{ref}})$ is smaller than the tolerated error threshold ε.

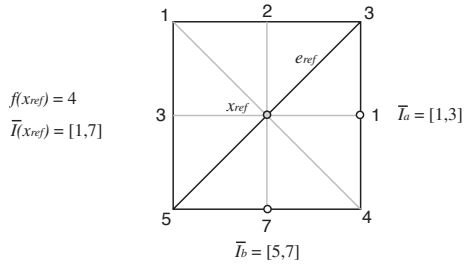

Fig. 9. Minimal saturation $\bar{I}(x_{\text{ref}}) = [\min\{I_a, I_b\}, \max\{\bar{I}_a, \bar{I}_b\}]$ from merging disjunct intervals $\bar{I}_a$ and $\bar{I}_b$ results in overly conservative critical point selection.

The minimal saturation of critical intervals is overly conservative since it covers multiple disjunct intervals by one critical interval, thus also including the non-critical isovalue ranges between disjunct intervals. In our second approach to saturation of critical intervals, we merge two intervals I_a and I_b into one only if they are not disjunct $I_a \cap I_b \neq \emptyset$, and keep an ordered list of disjunct intervals otherwise. Therefore, saturation of critical intervals actually generates ordered lists of disjunct intervals $\mathcal{I} = \{I_1, I_2, ...I_m\}$. In a level-wise bottom-up traversal of the tetrahedral hierarchy the *optimally saturated* critical interval lists can be computed by

$$\mathcal{I}(x_{\text{ref}}) = \{I(x_{\text{ref}})\} \bigoplus_{T \in \mathcal{T}(e_{\text{ref}})} \Big(\mathcal{I}(C_l(T)) \oplus \mathcal{I}(C_r(T)) \Big) . \tag{9}$$

The operation $\oplus$ in Equation 9 denotes the merging of two ordered lists of intervals into one ordered interval list in which all overlapping intervals are joined to reduce the number m of intervals in a list. Using this approach, each vertex not only stores its geometric error indicator $\bar{\eta}$ but also a list of critical intervals $\mathcal{I}$ of size m.

The adaptive isosurface extraction algorithm subdivides a tetrahedron T recursively if $\bar{\eta}(x_{\text{ref}}(T)) > \varepsilon$, or if

$$\exists I \in \mathcal{I}(x_{\text{ref}}(T)) \text{ with } \tau \in I. \tag{10}$$

The test in Equation 10 if the isovalue τ intersects any interval segment in the ordered list $\mathcal{I}$ of critical intervals can efficiently be implemented using binary search on the ordered interval list $\mathcal{I}$. We will refer to this method as the *optimally saturated* isosurface extraction.

4.6 Experimental Results

4.6.1 Mesh Complexity

In order to show the power of the proposed algorithms to extract triangulated isosurfaces of minimal complexity at arbitrary LODs while preserving the topology of the original full resolution isosurface let us consider the zero set of the algebraic function

$$f(x,y,z) = \sqrt{x^2+y^2} - (\frac{1}{2}x + \frac{1}{2}y - z + 0.01)^2 \tag{11}$$

defined on $[-1,1]^3$, sampled on a uniform grid of size 65^3. This function has a strong singularity near the origin as can be seen in Figure 10. In Table 1 we compare the size of triangle meshes for the isosurface $f(x,y,z) = 0$ extracted at different LODs with various geometric error threshold values ε (L_∞-norm). The basic multiresolution isosurface extraction algorithm [30,9] extracts triangle meshes of low complexity, however, the important isosurface structure of the singularity at the origin gets lost for certain error thresholds. In contrast, our proposed multiresolution isosurface extraction techniques preserve the original isosurface topology even at low triangle counts for arbitrary LODs. Compared to the non-topology preserving approach, the *minimally saturated* isosurface extraction method achieves competitive mesh simplification rates up to a geometric error threshold of $\varepsilon = 1/4$. The *optimally saturated* method performs even much better, and produces isosurfaces of comparable mesh complexity up to very large geometric error thresholds such as $\varepsilon = 1$.

As a second example we will consider the geometrically much more complicated buckyball data set (courtesy of AVS). Figure 11 shows isosurfaces for several LODs with different geometric error threshold values, and for an isovalue where the data set exhibits a very complex topological structure. Note that our proposed topology preserving isosurface extraction methods perform nearly as well as the non-topology preserving approach for isosurfaces with low topological complexity. The values of ε and the corresponding triangle counts are listed in Table 2. Without topology preservation, the isosurface does not maintain the correct surface connectivity over different LODs (third image from the left in top row), or even completely falls apart (fourth image in top row). With our topology preserving approaches, however, the original topology is exactly retained. The *minimally saturated* approach performs well for small geometric error thresholds up to $\varepsilon = 0.01$, however, cannot reduce the triangle mesh complexity significantly anymore for larger thresholds. The *optimally saturated* approach achieves better results, and is able to reduce

ε	without topology preservation	minimally saturated topology preservation	optimally saturated topology preservation
0	59290	59290	59290
1/64	25456	25464	25456
1/16	7124	7528	7124
1/4	1936	3503	2028
1	374	3095	808

Table 1. Triangle counts for the algebraic function of Equation 11 for varying geometric error threshold values ε.

ε	without topology preservation	minimally saturated topology preservation	optimally saturated topology preservation
0.0	395798	395798	395798
0.01	153539	163818	156042
0.05	29594	110101	55838
0.1	12552	109981	52268

Table 2. Triangle counts for the buckyball data set for varying geometric error threshold values ε.

the triangle count for geometric error thresholds as high as $\varepsilon = 0.05$.

In both *minimally* and *optimally saturated* approaches, however, the amount of geometric simplification is limited due to the high complexity of the topology that the isosurface exhibits nearly everywhere in space for the buckyball data set. After a certain LOD, increasing the geometric error threshold ε will not result in lower triangle counts anymore. In fact, the lower bound of the triangle mesh complexity will be reached for a tetrahedral mesh that consists of all (saturated) hierarchically critical points, thus no more simplification is possible without changing the isosurface's topology. Due to the improved handling of saturated critical intervals, the *optimally saturated* approach exhibits this limiting effect significantly later, for much higher geometric error thresholds than the *minimally saturated* approach.

4.6.2 Performance

In this algorithm, the extraction speed per triangle is almost as fast as the rendering speed for a triangle on modern graphics workstations. In [9] a combined extraction and drawing rate of up to 300.000 triangles per second for a single processor and up to 800.000 triangles per second for a multiprocessor machine have been reported. This speed can also be measured for our topology-preserving version since only one additional check (namely if the isovalue is contained in the current critical interval) is necessary for each tetrahedron during the tree traversal. Typical mesh simplification algorithms have lower simplification rates of at least 2-3 orders of magnitude [25].

Let us also take a look at the memory overhead. In addition to the data the error indicator values, the minmax bounds, and the critical intervals have to be stored for each refinement vertex up to the second finest resolution. In some cases, the minmax bounds can be derived from the error indicator and therefore don't need to be stored [10]. However, with conservative quantization all additional information can be stored with roughly the same memory requirements as the data itself (see [9]).

The preprocessing step for the computation of this information requires a single level-wise bottom-up traversal of the whole data set which takes at most a few seconds for the considered examples. If the data set is stored in traversal order, the necessary computations can even be done during the data reading phase.

5 Controlled Topology Simplification

As we have seen in the experiments of Section 4, topology preservation will usually infer a limit on the minimum number of triangles for a particular isosurface mesh. Depending on the application, it might be desirable to further simplify the resulting triangle meshes at the expense of controlled topological changes of the isosurface.

While the original – non-topology preserving – multiresolution algorithm simplifies topology naturally, it does so in an uncontrolled fashion. In principle, it would be possible to employ controlled topology simplifying triangle mesh reduction schemes such as filtering and resampling [12], or α-hulls [7] in a postprocessing step of the isosurface extraction. However, these algorithms work in a bottom-up fashion on the extracted triangle mesh at the highest resolution and would therefore not meet our real-time performance constraints.

5.1 Choices for Simplification

The topology preserving algorithm described in the previous section can be transformed into a controlled topology simplifying one. To this end, let us define a *simplification weight* $w(x_{\text{ref}})$ on all hierarchical critical points which indicates the importance of that particular point. The topology simplifying algorithm will then be able to discard those critical points which are not important, i.e. where $w(x_{\text{ref}}) < \delta$ with δ being a user-defined threshold to bound topological changes.

This way, the user has complete control over topology simplification by the specification of a suitable weight function w on the hierarchical critical points together with a threshold δ. The actual values of the weight w at the hierarchical critical points can be precomputed and saturated together with the error indicator and the critical intervals.

An appropriate choice for the weight w strongly depends on the type of application and structure of the data set. Let us consider a few possibilities for the simplification weight:

- An obvious choice for w would be the size of the critical interval. This requires no extra computational effort in the presented framework. By a multiplication with the maximum gradient of all adjacent tetrahedra, one has an indicator for the amount of topological changes in the isosurface structure introduced by refinement of the tetrahedra surrounding the critical point.
- Another possibility would be the size of the corresponding local isosurface, i.e. the number of tetrahedra the family of isosurfaces within the critical interval would intersect with. This provides a control mechanism over the local geometric complexity.
- One could also choose bounds for the size of critical features, i.e. distance between parts of the isosurface that are

separated by the critical point. This for example makes sense in the case of local minima or maxima (decision if small local minima/maxima features are shown or not), or for saddle points (i.e. may isosurface components be merged if sufficiently close together). This allows control of the geometric separation between surface components, i.e. disconnected components could be merged and simplified if the distance between them is small enough.

5.2 Example

An extensive comparison of simplification techniques for different applications would certainly exceed the scope of this paper. Therefore, we will here just consider a simple but characteristic example. Let us look at a typical bio-medical data set, the famous CT scan of a lobster (courtesy of AVS). In Figure 12 we see that original isosurface (1089004 triangles) is very noisy. The topology preserving algorithm can reduce the number of triangles to 609893, but the noise in the data set will of course remain. Let us therefore define the weight function dependent on the size of the critical interval,

$$w(x_{\text{ref}}) := |I(x_{\text{ref}})|.$$

Topology simplification will then reduce these artifacts nicely and maintain the overall topological structure of the animal. The simplified isosurfaces provide visually better representations of the data set with only 367271 triangles for $\delta = 0.2$ and 262357 triangles for $\delta = 0.7$, respectively.

6 Concluding Remarks

We have shown how topology preservation and controlled topology simplification can be achieved in multiresolution isosurface extraction on hierarchical tetrahedral meshes generated by recursive bisection. Thereby, we did not focus on methods working on the isosurface triangle mesh but on the underlying 3D volume data set itself.

While we mainly considered tetrahedral meshes based on regular refinement in this paper, our topology considerations apply to more irregular tetrahedral meshes generated by bisection as well. Refinement methods for tetrahedral mesh generation can thereby consider the most critical points for insertion, whereas decimation methods will consider non-critical or less-critical points for removal. With appropriate modifications the methodology could be extended to handle other hierarchical tetrahedral meshes as well although this would probably require an identification of more cases and look-up tables.

Finally, the proposed methods may not only be important by themselves, but can also be used as a fast preprocessor to other algorithms which rely on a topologically consistent input mesh.

References

1. C. Andujar, D. Ayala, P. Brunet, R. Joan-Arinyo, and J. Sole. Automatic Generation of Multiresolution Boundary Representations. *Computer Graphics Forum*, 15(3):87–96, 1996.
2. C. Bajaj and D. Schikore. Topology Preserving Data Simplification with Error Bounds. *Computers & Graphics*, 22(1):3–12, 1998.
3. E. Bloch. *A First Course in Geometric Topology and Differential Geometry*. Birkhäuser, Boston, Massachusetts, 1997.
4. P. Cignoni, F. Ganovelli, C. Montani, and R. Scopigno. Reconstruction of Topologically Correct and Adaptive Trilinear Isosurfaces. *Computers & Graphics*, 24(3):399–418, 2000.
5. P. Cignoni, C. Montani, and R. Scopigno. MagicSphere: An Insight Tool for 3D Data Visualization. *Computer Graphics Forum*, 13(3):317–328, 1994.
6. M. Duchaineau, M. Wolinsky, D. E. Sigeti, M. C. Miller, C. Aldrich, and M. B. Mineev-Weinstein. Roaming terrain: Real-time optimally adapting meshes. In *Proceedings Visualization 97*, pages 81–88, Los Alamitos, California, 1997. IEEE, Computer Society Press.
7. J. El-Sana and A. Varshney. Controlled Simplification of Genus for Polygonal Models. In *Proceedings IEEE Visualization '97*, pages 403–412. IEEE Press, 1997.
8. T. Gerstner and R. Pajarola. Topology preserving and controlled topology simplifying multiresolution isosurface extraction. In *Proceedings Visualization 2000*, pages 259–266, Los Alamitos, California, 2000. IEEE, Computer Society Press.
9. T. Gerstner and M. Rumpf. Multiresolutional Parallel Isosurface Extraction based on Tetrahedral Bisection. In M. Chen, A. Kaufman, and R. Yagel, editors, *Volume Graphics*, pages 267–278. Springer, 2000.
10. T. Gerstner, M. Rumpf, and U. Weikard. Error Indicators for Multilevel Visualization and Computing on Nested Grids. *Computers & Graphics*, 24(3):363–373, 2000.
11. R. Grosso, C. Lürig, and T. Ertl. The Multilevel Finite Element Method for Adaptive Mesh Optimization and Visualization of Volume Data. In *Proceedings IEEE Visualization '97*, pages 387–394. IEEE Computer Society Press, 1997.
12. T. He, L. Hong, A. Varshney, and S. Wang. Controlled Topology Simplification. *IEEE Transactions on Visualization and Computer Graphics*, 2(2):171–184, 1996.
13. P. Heckbert and M. Garland. Survey of Surface Approximation Algorithms. In *SIGGRAPH '97 Course Notes*, 1997.
14. T. Itoh and Y. Yamaguchi. Isosurface Generation using Extrema Graphs. In *Proceedings IEEE Visualization '94*, pages 77–83. IEEE Computer Society Press, 1994.
15. P. Lindstrom, D. Koller, W. Ribarsky, L. F. Hodges, N. Faust, and G. A. Turner. Real-time, continuous level of detail rendering of height fields. In *Proceedings SIGGRAPH 96*, pages 109–118. ACM SIGGRAPH, 1996.
16. Y. Livnat and C. Hansen. View Dependent Isosurface Extraction. In *Proceedings IEEE Visualization '98*, pages 175–180. IEEE Computer Society Press, 1998.
17. Y. Livnat, H. Shen, and C. Johnson. A Near Optimal Isosurface Extraction Algorithm using the Span Space. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):73–83, 1996.
18. W. Lorensen and H. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics*, 21(4):163–169, 1987.
19. J. Maubach. Local Bisection Refinement for n-simplicial Grids generated by Reflection. *SIAM J. Sci. Comp.*, 16:210–227, 1995.
20. J. Milnor. *Morse Theory*. Princeton University Press, Princeton, New Jersey, 1963.
21. B. Natarajan. On Generating Topologically Consistent Isosurfaces from Uniform Samples. *The Visual Computer*, 11(1):52–62, 1994.

22. M. Ohlberger and M. Rumpf. Adaptive Projection Methods in Multiresolutional Scientific Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 4(4):74–94, 1998.
23. R. Pajarola. Large scale terrain visualization using the restricted quadtree triangulation. In *Proceedings Visualization 98*, pages 19–26,515, Los Alamitos, California, 1998. IEEE, Computer Society Press.
24. B. Payne and A. Toga. Surface Mapping Brain Function on 3D Models. *IEEE Computer Graphics and Applications*, 10(5):33–41, 1990.
25. E. Puppo and R. Scopigno. Simplification, LOD and Multiresolution Principles and Applications. In *Eurographics '97 Tutorial Notes*, 1997.
26. R. Shekhar, E. Fayyad, R. Yagel, and J. Cornhill. Octree–Based Decimation of Marching Cubes Surfaces. In *Proceedings IEEE Visualization '96*, pages 335–244. IEEE Computer Society Press, 1996.
27. A. Van Gelder and J. Wilhelms. Topological Considerations in Isosurface Generation. *ACM Transactions on Graphics*, 13(4):337–375, 1994.
28. B. Von Herzen and A. H. Barr. Accurate triangulations of deformed, intersecting surfaces. In *Proceedings SIGGRAPH 87*, number 4 in ACM Journal Computer Graphics, pages 103–110. ACM SIGGRAPH, 1987.
29. J. Wilhelms and A. Van Gelder. Octrees for Faster Isosurface Generation. *ACM Transactions on Graphics*, 11(3):201–227, 1992.
30. Y. Zhou, B. Chen, and A. Kaufman. Multiresolution Tetrahedral Framework for Visualizing Volume Data. In *Proceedings IEEE Visualization '97*, pages 135–142. IEEE Computer Society Press, 1997.
31. Y. Zhou, W. Chen, and Z. Tang. An Elaborate Ambiguity Detection Method for Constructing Isosurfaces within Tetrahedral Meshes. *Computers & Graphics*, 19(3):355–364, 1995.

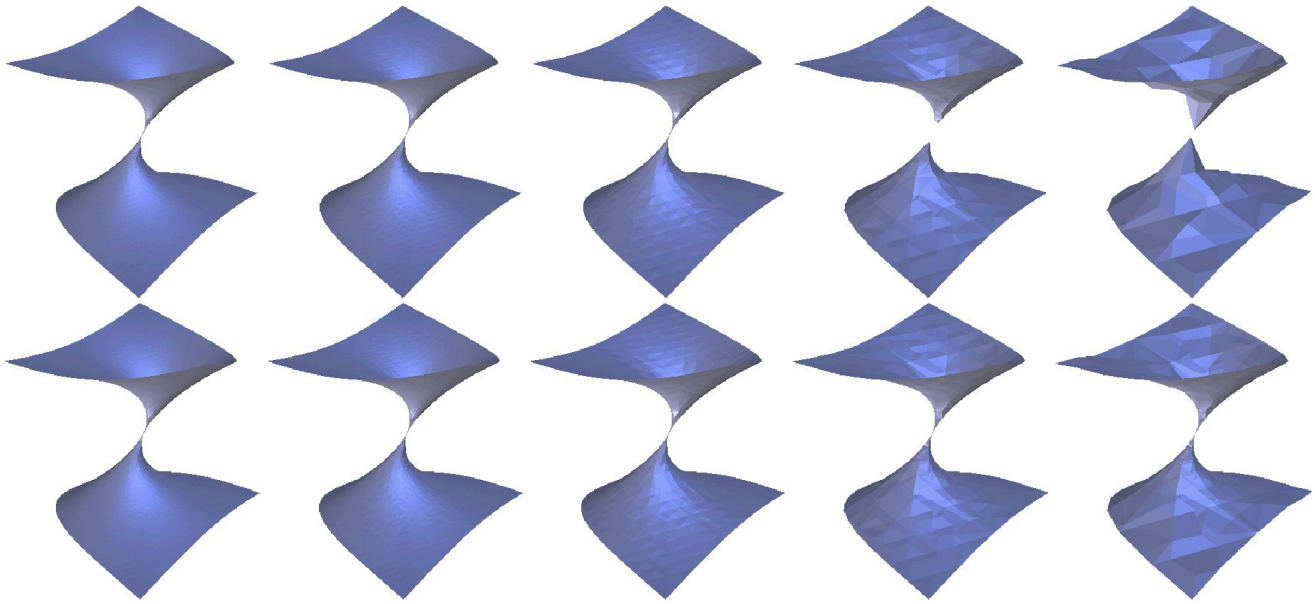

Fig. 10. Isosurfaces for the algebraic function of Equation 11 without topology preservation (upper row), with the minimally saturated (middle row), and with the optimally saturated (lower row) topology preserving approach for varying error tolerances ε of 0, $1/64$, $1/16$, $1/4$, and 1 (from left to right).

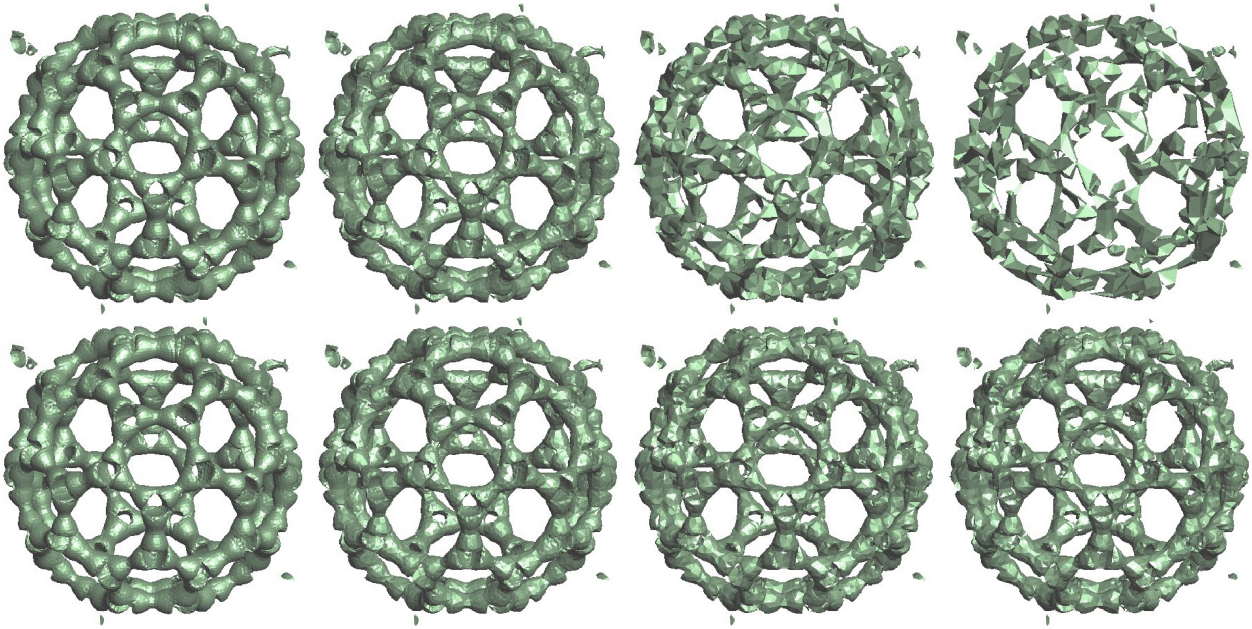

Fig. 11. Isosurfaces of the buckyball data set without topology preservation (upper row), with the minimally saturated (middle row), and with the optimally saturated (lower row) topology preserving approach for varying error tolerances ε of 0.0, 0.01, 0.05, and 0.1 (from left to right).

Fig. 12. Isosurfaces of the lobster data set: original, topology preserving and controlled topology simplified versions (from left to right).