

Improving the Robustness and Accuracy of the Marching Cubes Algorithm for Isosurfacing

Adriano Lopes and Ken Brodlie

Abstract—This paper proposes a modification of the Marching Cubes algorithm for isosurfacing, with the intent of improving the representation of the surface in the interior of each grid cell. Our objective is to create a representation which correctly models the topology of the trilinear interpolant within the cell and which is robust under perturbations of the data and threshold value. To achieve this, we identify a small number of key points in the cell interior that are critical to the surface definition. This allows us to efficiently represent the different topologies that can occur, including the possibility of “tunnels.” The representation is robust in the sense that the surface is visually continuous as the data and threshold change in value. Each interior point lies on the isosurface. Finally, a major feature of our new approach is the systematic method of triangulating the polygon in the cell interior.

Index Terms—Isosurface, marching cubes, robustness, accuracy, trilinear interpolation.

1 INTRODUCTION

THE accurate representation of an isosurface has proved to be one of the most interesting challenges in scientific visualization. Since the early work on the Marching Cubes algorithm by Lorensen and Cline [1], successive authors have sought to improve the topological correctness and accuracy of the surface representation. In this paper, we build on this work to propose an approach which gives a representation which is not only topologically correct and accurate, but is also robust in the sense of being continuous with respect to changes in the data or isosurface level.

We suppose we are given scalar data defined on a 3D rectilinear mesh and the required isosurface level. The Marching Cubes algorithm generates the set of isosurfaces for that value by inspecting each cell in turn and constructing the corresponding portion, or portions, of surface. Thus, the problem can be reduced to the construction of the isosurface within a cube and, without loss of generality, we may take this to be the unit cube with corner at origin and we may take the isosurface threshold value as zero.

Since the data is only given at the vertices of the cube, we must model the data in the interior, and we shall take as our model the trilinear interpolant, $F(x, y, z)$. F can be written in terms of the vertex values F_{ijk} , $i, j, k = 0, 1$ as:

$$\begin{aligned} F(x, y, z) = & F_{000}(1-x)(1-y)(1-z) + \\ & F_{001}(1-x)(1-y)z + F_{010}(1-x)y(1-z) + \\ & F_{011}(1-x)yz + F_{100}x(1-y)(1-z) + \\ & F_{101}x(1-y)z + F_{110}xy(1-z) + F_{111}xyz. \end{aligned} \quad (1)$$

An alternative representation of (1), which we shall use in this paper, is

$$\begin{aligned} F(x, y, z) = & a + ex + cy + bz + \\ & gxy + fxz + dyz + hxyz, \end{aligned} \quad (2)$$

with $a = F_{000}$, $b = F_{001} - F_{000}$, $c = F_{010} - F_{000}$, and so on. Other modeling functions can be used, but the trilinear interpolant is the most common. In this paper, we assume the modeling decision has been taken—to use a trilinear reconstruction—and we aim to represent this model accurately and efficiently.

The original Marching Cubes algorithm identifies 256 configurations for the cube, depending on whether each of the eight vertices is positive or negative. Any edge with endpoints of opposite sign is intersected by the isosurface and inverse linear interpolation yields an estimate of the intersection point. The set of intersection points can be triangulated to yield an approximation to the isosurface within the cube. In fact, Lorensen and Cline showed that, with rotational symmetry and complementarity (switching positive and negative vertices), the 256 configurations can be reduced to a set of 15 cases. A further case can be removed by reflectional symmetry, leaving a canonical set of 14 cases. These are shown in Fig. 1.

Since its original conception, the Marching Cubes algorithm has been the subject of much further research to improve its quality of surface representation and its performance on large data sets. It is the quality issue with which we are concerned in this paper: the quality of representing the trilinear interpolating surface. The key contributions on which we build have been by Nielson and Hamann [2], Natarajan [3], Chernyaev [4], and Cignoni et al. [5].

Nielson and Hamann [2] point out that there is an ambiguity in the face of a cube when all four edges of the face are intersected. The triangulation has to choose which pairs of intersections to connect. An inconsistent strategy can lead to “holes” when surfaces in adjacent cells use

- A. Lopes was with the Department of Mathematics of FST/CIS, University of Coimbra, Portugal. He is currently with the Department of Informatics, New University of Lisbon, Portugal. E-mail: alopes@di.fct.unl.pt.
- K. Brodlie is with the School of Computing, University of Leeds, United Kingdom. E-mail: kvb@comp.leeds.ac.uk.

Manuscript received 7 Jan. 2000; revised 19 Jan. 2001; accepted 28 Aug. 2001. For information on obtaining reprints of this article, please send e-mail to: tcvg@computer.org, and reference IEEECS Log Number 111190.

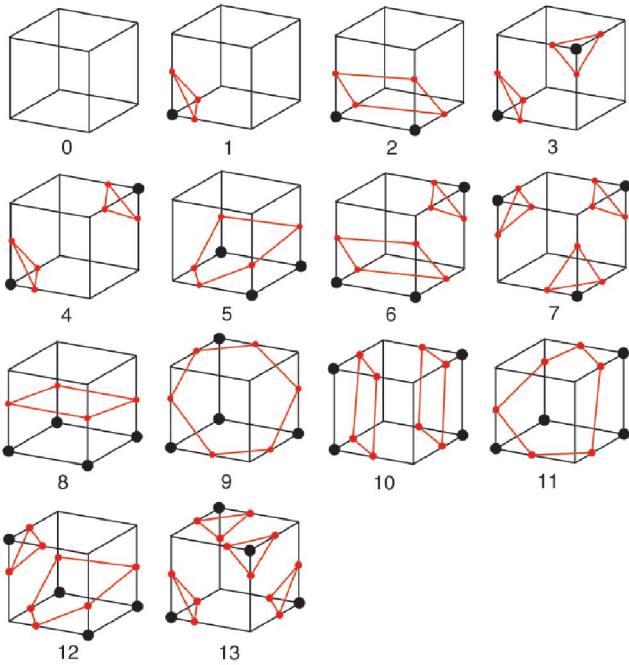


Fig. 1. Basic cases in the Marching Cubes algorithm and examples of how the points of intersection between the isosurface and the edges of the cube can be connected. Positive vertices are marked.

different connections on the common face (as pointed out at an earlier date by Dürst [6]). Nielson and Hamann propose a strategy based on saddle point value of the bilinear interpolant to dictate the edge connections on an ambiguous face: The connection is made so as to “separate” the saddle point from vertices of opposite sign. This gives a solution which is topologically correct in terms of the bilinear interpolant on the face. Of the 14 cases in Fig. 1, cases 3, 6, 7, 10, 12, and 13 all involve at least one ambiguous face.

Natarajan [3] and Chernyaev [4] independently recognize that there are additional ambiguities in the representation of the trilinear interpolant in the interior of the cube. For example, in case 4, shown in Fig. 2, there are two possible topologies: The surface may be in two separate pieces or there may be a single “tunnel” piece, topologically equivalent to a cylinder. Natarajan uses the value of the body saddle point (where all three first partial derivatives of $F(x, y, z)$ are zero) to discriminate the two cases. If the

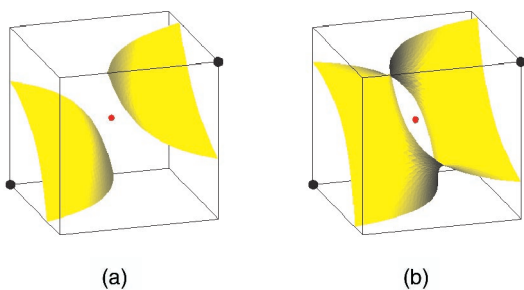


Fig. 2. Natarajan [3] uses the value of the body saddle point (red sphere in the interior) to discriminate the two possible topologies (a) and (b) in case 4.

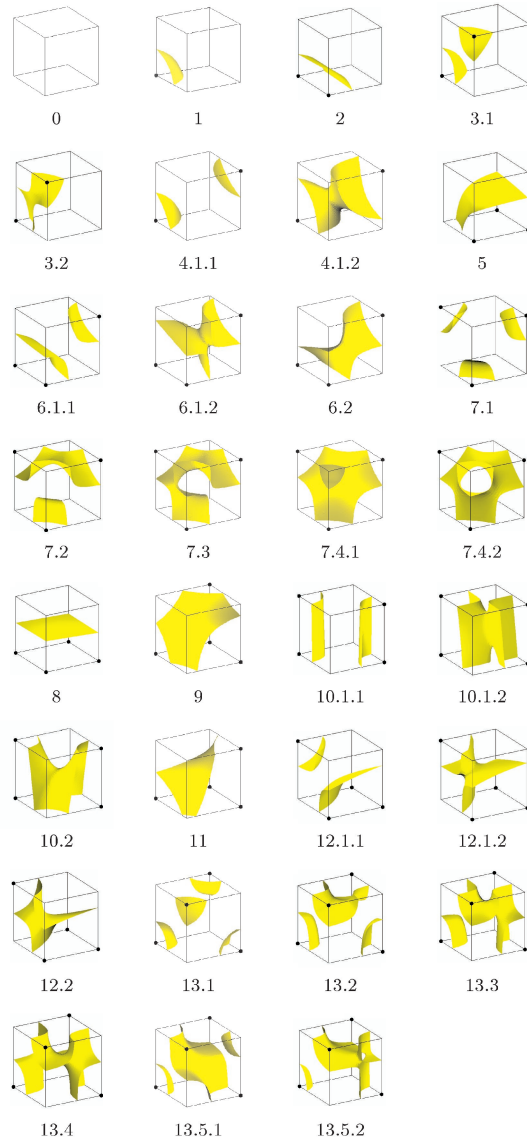


Fig. 3. The 31 cases in the Marching Cubes algorithm. As in Chernyaev’s notation [4], the first label number denotes the original Marching Cubes case number as in Fig. 1; the second indicates the resolution of a face ambiguity (six ambiguous faces in the group of case 13, three in case 7, two in cases 10 and 12, one in cases 3 and 6, none in the others); and the third the resolution of an interior ambiguity.

body saddle is negative (that is, opposite in sign to the two marked positive vertices), the isosurface has two pieces separated by the body saddle point. If the body saddle is positive, then there is a tunnel linking the two marked positive vertices.

Chernyaev uses a different strategy for discriminating, but the effect is the same. Chernyaev [4] gives a definitive classification of the cases that can occur, in total 33 different situations. Two cases can be removed: Case 14 of Chernyaev’s table is equivalent under reflection to case 11 and (as Chernyaev himself admits) case 12.3 is a mirror of case 12.2. Thus, we omit cases 14 and 12.3, leaving 31 distinct cases. These 31 cases are shown in Fig. 3, using the same notation as Chernyaev for labeling the cases. The first number denotes the original Marching Cubes case number as in Fig. 1, the second indicates the resolution of a face

ambiguity, and the third the resolution of an interior ambiguity. Case 13 is the most complex, with all six faces being ambiguous. It is also complex in the interior, where case 13.5 has two body saddle points within the cube. We shall see later in the paper that a tunnel may form in two different ways.

Matveyev [7] also discusses the interior ambiguity problem, resolving the ambiguous cases by considering the behavior of the trilinear function along the cell diagonals.

For practical purposes, the isosurface within a cell must be represented as a triangulated set of points which can be passed to a renderer. In the original algorithm of Lorensen and Cline, a simple strategy of triangulating the edge intersections is adopted, and this is sufficient for the 14 cases. However, if the ambiguities are properly treated, then the triangulation becomes more complex and interior points are required in addition to edge intersections.

In resolving the face ambiguities, Nielson and Hamann use interior points in subcases of cases 7, 10, and 13. For cases 7 and 13, inverse linear interpolation along a diagonal is used to estimate an interior point and, in case 10, a point on the line joining saddle points on opposite faces is used. The additional point will not lie exactly on the surface (unlike the edge intersections).

An important principle is mentioned by van Gelder and Wilhelms [8]. In order for the triangulated surface to be continuous (C_0), each polygon edge must belong to no more than two triangles. In particular, this forbids any triangles from lying in the face of a cell since these will have an edge belonging to two triangles from that cell and to at least one triangle from the adjoining cell.

Chernyaev [4] gives a detailed set of triangulations for each of the cases. However, some of these fail the van Gelder and Wilhelms test in that triangles lie on the faces of the cube (see, for example, case 6.1.2 in Fig. 8 of [4]). Hence, the resulting surface will not be continuous. When an interior point is required, Chernyaev chooses the average of the edge intersections—as with Nielson and Hamann, this point will not normally lie on the surface.

Natarajan [3] does not explicitly show how to triangulate the surface. Although he mentions the need for continuity as defined by van Gelder and Williams, the illustrations in the paper imply triangles in the faces (see Fig. 5—lower middle diagram—of [3]) and no remedial action to avoid discontinuity (such as adding interior points) is described.

Natarajan's work is used as a basis by Cignoni et al. [5]. They propose an "Extended Look Up Table (ELUT)" which aims to provide explicit triangulations for a large number of cases based on face saddle and body saddle values. When interior points are required, they use points on the cell's diagonal obtained by interpolation of the values at opposite corners of the cell. In particular, an interior point is introduced to avoid triangles lying on the cell faces. Note, however, that, with a simple inverse linear interpolation of cell vertex values, these interior points would not lie on the actual trilinear surface since the interpolant is cubic along the diagonals. The authors do not appear to consider case 13.4, where there is a single, complex surface piece, nor do they appear to fully handle the complexity of case 13.5,

where more than one body saddle point exists. (Specifically, we believe there are situations missing from Table 10 of [5]. Case 13.4 as illustrated in our Fig. 3 and Fig. 14f with data values in Table 2 does not appear. It has opposite-sign saddle values on opposing faces, $+ - + - +$ in the order used in [5] and, in the particular case of this data set, an imaginary body saddle point. Likewise, Table 10 only allows for one body saddle point.) Thus, we believe that, while the ELUT gives a very useful set of triangulations for many of the cases, it is incomplete. It is also rather lengthy, having some 798 different polygonizations.

Cignoni et al. do suggest an important refinement strategy that is useful when a highly accurate representation of the trilinear surface is required. Triangles are progressively split, with new interior points being added. As the authors point out, such a refinement process must start from a topologically correct first approximation. Likewise, for accuracy, the first approximation should use points lying on the surface—hence, any interior points generated in the initial triangulation should lie on the isosurface. Mesh refinement is particularly useful in virtual manufacturing, where Bailey [9] has independently suggested a similar refinement strategy.

Our conclusion is that we now have a good understanding of the interior topology of the trilinear interpolant within the cube, but we still need to understand better how to represent this as a triangulated set of points.

Moreover, there is an important consideration which has not been addressed at all to date. We would like the representation to be continuous with respect to changes in the data or changes in the threshold. This means that, as the threshold changes so that a different choice is made to resolve an ambiguity, so the representation should not display a visual discontinuity. Most existing representations, unfortunately, will not have this robustness property.

Our aim in this paper therefore is to propose a triangulation strategy according to a number of principles:

- The representation should give a topologically correct approximation to the isosurface within the cube. This should follow the definitive treatment of Chernyaev.
- The representation should be robust with respect to changes in the data or changes in the isosurface threshold. By this we mean that the transition between the different states in the ambiguous cases should be as smooth as possible.
- The representation should be efficient in terms of number of triangles used, subject to achieving the topological correctness and robustness properties.
- The representation should be accurate within the limits of the efficiency condition above. In particular, any interior points required should be chosen to lie on the surface. The triangulation should then be capable of acting as the basis of a highly accurate representation using a progressive refinement of the triangulation, as suggested by Cignoni et al. and by Bailey.
- The triangulation strategy should be explicit so as to act as a benchmark for implementation and to allow future improvement and extension.

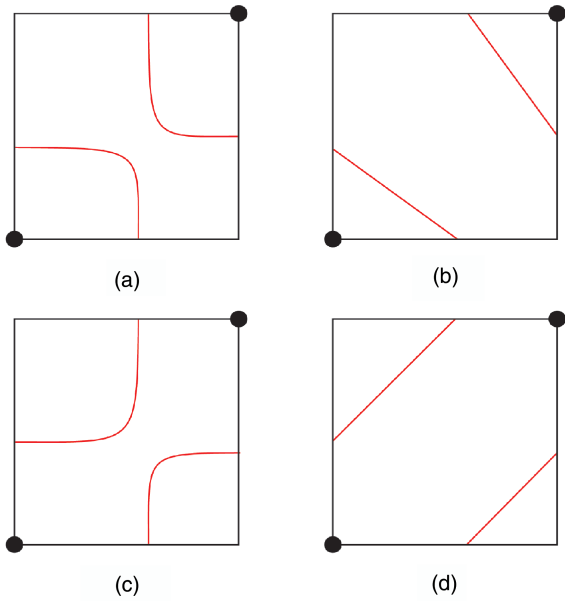


Fig. 4. Error in the approximation of the hyperbolae. In (a) and (b), the contours cut off the positive vertices, whereas, in (c) and (d), the negative vertices are separated. The difference in the visual appearance of the hyperbolae—(a) and (c)—is considerably magnified when each hyperbolic arc is represented by a straight line—(b) and (d), respectively.

2 REPRESENTATION ON THE FACES OF THE CUBE

The intersection of the isosurface with a face of the cube is a set of conic curves: hyperbolae which represent the zero level contour of the bilinear interpolant on that face. The standard approach is to approximate the hyperbolae with straight lines, using the asymptotic decider to resolve the ambiguous case as mentioned earlier. An unsatisfactory aspect of this approach is the discontinuous behavior of this approximation close to the critical case where the conic is a pair of straight lines.

We illustrate this in Fig. 4. In Fig. 4a and Fig. 4b, the contours cut off the positive vertices; in Fig. 4c and Fig. 4d, the negative vertices are separated—yet there has only been a small change to the contour threshold. Moreover, it is clear that there is a significant error in the approximation of the hyperbola.

This motivates the idea of adding one additional point to the polygonal representation of the contour on a face. We choose the point so that we gain continuity of visual appearance as we pass through the critical case and we select this additional point so as to minimize the error in the polygonal representation of the hyperbola.

The point we choose is the *shoulder point* of the hyperbolic arc. This is defined as the point on the hyperbolic arc where the tangent is parallel to the chord joining the endpoints of the arc—see Fig. 5, where \overline{PQ} is the chord and R the shoulder point. Crucially, in this context, we can show that the shoulder point lies on the line joining M , the midpoint of \overline{PQ} , and the saddle point S of the bilinear interpolant. Thus, in the ambiguous case, as the contour threshold approaches the critical value (the value at the saddle point), the hyperbolae tend to a pair of straight lines and the shoulder point R converges to S . The polygonal

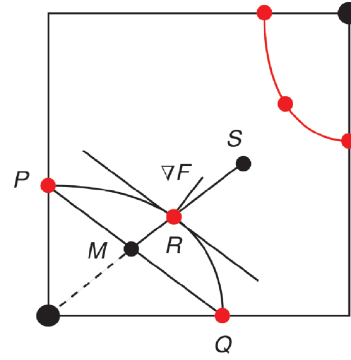


Fig. 5. Shoulder point R as an extra point on the hyperbolic arc. It lies on the line \overline{MS} , where M is the midpoint of \overline{PQ} and S the saddle point. The shoulder point R lies also on the line parallel to \overline{PQ} .

representation of the contour thus passes smoothly from one case to the other. This is illustrated in Fig. 6.

Moreover, the point R is optimal in the sense of providing a more accurate representation. Because the tangent at R is parallel to \overline{PQ} , R maximizes the area of the triangle PQR (for all possible points R on the hyperbola) and therefore minimizes the area between the hyperbola and its polygonal representation.

The formulae to compute R are given in Appendix A. A more complete discussion of this is given in Lopes’ PhD thesis [10] and a preliminary version (in context only of contouring) was presented by Lopes and Brodlie [11].

In terms of isosurfacing, the effect is as follows: The boundary polygon (or polygons) that is formed in other methods by a set of edge intersection points is now enhanced by the addition of these shoulder points—we call them *face shoulder points*—that provide greater accuracy and robustness.

3 REPRESENTATION IN THE INTERIOR OF THE CUBE

3.1 Understanding the Different Topologies

The work of earlier authors, notably Natarajan [3] and Chernyaev [4], has helped to clarify the behavior of the trilinear interpolant within the cube. For each surface piece, there are two topologically distinct types:

Type A: The surface piece is homeomorphic to a disc—as in case 1 for example;

Type B: The surface piece is homeomorphic to a cylinder—as in case 4.1.2 for example.

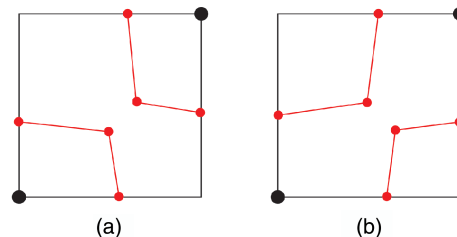


Fig. 6. Using shoulder points as additional points to represent the hyperbolae. In comparison to Fig. 4, the representation of the contour passes smoothly from (a) to (b).

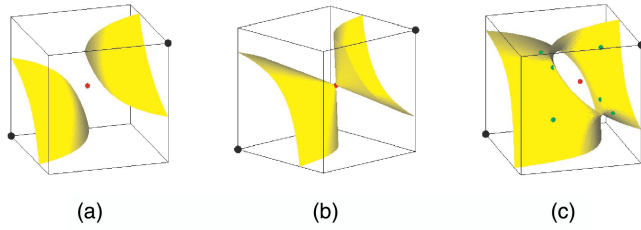


Fig. 7. Case 4: From (a), two surfaces homeomorphic to discs to (c) a surface homeomorphic to cylinder. In (c), the body saddle point is internal to a cuboid formed by six tangent points (term further explained in Section 3.4). Picture (b) shows the transition from (a) to (c), where the body saddle point and (three) tangent points all coincide; the cuboid degenerates into a point.

It is useful to subdivide type A further, according to whether the surface piece intersects a face (or faces) more than once. Thus, we have:

Type A.1: The surface piece intersects any face once only—as in case 1 for example;

Type A.2: The surface piece intersects a face (or faces) twice—as in case 6.2 for example.

Because robustness and visual continuity are a motivator for our work, we need to understand the transition process between the two topological types. It is instructive to examine case 4—see Fig. 7 where we may have either two surface pieces (of type A.1) or one surface piece (of type B). Following Natarajan, we can detect which case applies by examining the value at the body saddle point. If it is negative, the isosurface has two pieces separated by the body saddle; if it is positive, there is a single “tunnel” piece with the body saddle inside the tunnel. Imagine now a situation where there are two pieces and decrease the threshold from zero through the value of the body saddle—the two curved pieces will touch (at the body saddle point), merge, and open out into a tunnel. It will be important to preserve this topological “animation” in our isosurface representation.

In Table 1, we collect important information about each of Chernyaev’s cases. For each case, we show the topologies that occur (A.1, A.2, B). For type A.1, there will be a boundary polygon with a number of edge intersection points (say, M) and extended by intermediate face shoulder points—we indicate this by $P_{M/0}^E$, where E indicates the extension and 0 indicates there are no faces which are intersected twice. For type A.2, there will be a number of faces which are intersected twice—we indicate this by $P_{M/N}^E$, where N indicates the number of faces intersected twice. (If there is exactly one pair of opposite faces, this counts only once in the calculation for reasons which we shall explain later.) For type B, to reflect the fact that tunnels are formed by merging a pair of pieces, we use the notation $P_{M_\alpha/0}^E - P_{M_\beta/0}^E$. Note that, in each case, a tunnel is formed when two pieces of type A.1 merge.

Case 13.5 is complex and deserves more detailed discussion. This is given in the next section. We then look in turn at each topological type to understand how it may be represented in a robust, accurate, yet efficient way.

TABLE 1
Topologies and Polygons Involved
in Each Case Shown in Fig. 3

Case	Topologies	Polygons
0		
1	A.1	one $P_{3/0}^E$
2	A.1	one $P_{4/0}^E$
3.1	A.1	two $P_{3/0}^E$
3.2	A.2	one $P_{6/1}^E$
4.1.1	A.1	two $P_{3/0}^E$
4.1.2	B	one $P_{3/0}^E - P_{3/0}^E$
5	A.1	one $P_{5/0}^E$
6.1.1	A.1	one $P_{3/0}^E$, one $P_{4/0}^E$
6.1.2	B	one $P_{3/0}^E - P_{4/0}^E$
6.2	A.2	one $P_{7/1}^E$
7.1	A.1	three $P_{3/0}^E$
7.2	A.1, A.2	one $P_{3/0}^E$, one $P_{6/1}^E$
7.3	A.2	one $P_{9/3}^E$
7.4.1	A.1	one $P_{3/0}^E$, one $P_{6/0}^E$
7.4.2	B	one $P_{3/0}^E - P_{6/0}^E$
8	A.1	one $P_{4/0}^E$
9	A.1	one $P_{6/0}^E$
10.1.1	A.1	two $P_{4/0}^E$
10.1.2	B	one $P_{4/0}^E - P_{4/0}^E$
10.2	A.2	one $P_{8/1}^E$
11	A.1	one $P_{6/0}^E$
12.1.1	A.1	one $P_{3/0}^E$, one $P_{5/0}^E$
12.1.2	B	one $P_{3/0}^E - P_{5/0}^E$
12.2	A.2	one $P_{8/2}^E$
13.1	A.1	four $P_{3/0}^E$
13.2	A.1, A.2	two $P_{3/0}^E$, one $P_{6/1}^E$
13.3	A.1, A.2	one $P_{3/0}^E$, one $P_{9/3}^E$
13.4	A.2	one $P_{12/6}^E$
13.5.1	A.1	two $P_{3/0}^E$, one $P_{6/0}^E$
13.5.2	A.1, B	one $P_{3/0}^E$, one $P_{3/0}^E - P_{6/0}^E$

3.2 The Topology of Case 13.5

In most cases, the ambiguous interior behavior can be resolved by the “body saddle value” test of Natarajan described in the previous section. However, case 13.5 is harder and we need to extend the Natarajan arguments.

In this case, both body saddle points lie within the cube and, so, we have three cases which we illustrate in Fig. 8. In case 13.5.1, there are three surface pieces, one cutting off the back lower left, positive vertex, one cutting off the front top right, negative vertex, and the third lying across the middle of the cube. The body saddle points lie in the spaces between the surface pieces and are negative (in lower left gap) and positive (upper right gap).

Imagine now the threshold value decreasing so that (relatively speaking) both body saddle values are positive. This results in case 13.5.2 of Chernyaev, where the two lower pieces merge to form a tunnel whose interior is positive.

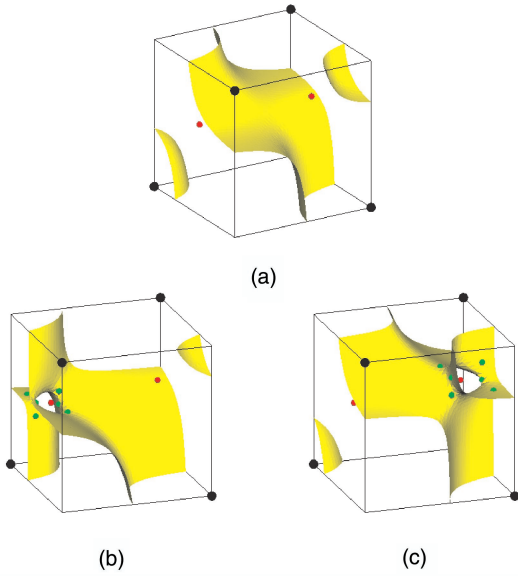


Fig. 8. Case 13.5, where both body saddle points lie within the cube. In (a), they lie in the spaces between the three surface pieces and are negative (in lower left gap) and positive (in upper right gap). In (b), the middle surface piece merges with the lower piece, whereas, in (c), it merges with the upper piece. In both (b) and (c), a body saddle point is internal to a cuboid formed by six tangent points (term further explained in Section 3.4).

Imagine, on the other hand, that the threshold value increases so that (again relatively speaking) both body saddle values are negative. This results in a complementary case, where the two upper pieces merge to form a tunnel whose interior is negative. (Following Chernyaev, we have not identified this as a separate canonical case since, by reversing signs, it can be transformed to case 13.5.2.)

Rather than decreasing or increasing the threshold, one can think of uniformly increasing or decreasing all vertex values to achieve the same effects.

This completes the understanding of the interior topology developed in the earlier papers of Natarajan, Chernyaev, and Cignoni et al., none of whom discuss the existence of two body saddle points.

3.3 Representing Type A.1

This is the simplest type, defined by an extended polygon with a certain number of intersection points and added face shoulder points. The approach to date has been to represent this type by triangulating the set of intersection points to form an approximation to the isosurface. We could continue this approach, with just the addition of face shoulder points to better define the boundary polygon.

However, the analysis of the previous section highlighted the fact that tunnels are formed when two surface pieces of type A.1 merge at the body saddle point. This motivates a search for an internal point to help improve the interior approximation of the isosurface—an internal point that would converge to the body saddle point in order to provide visual continuity at the point of merger as well as greater accuracy.

To do this, we build on the idea of the face shoulder as an optimal point on a face. Consider the cube as an infinite set of slices parallel to a pair of faces. Contour lines on these

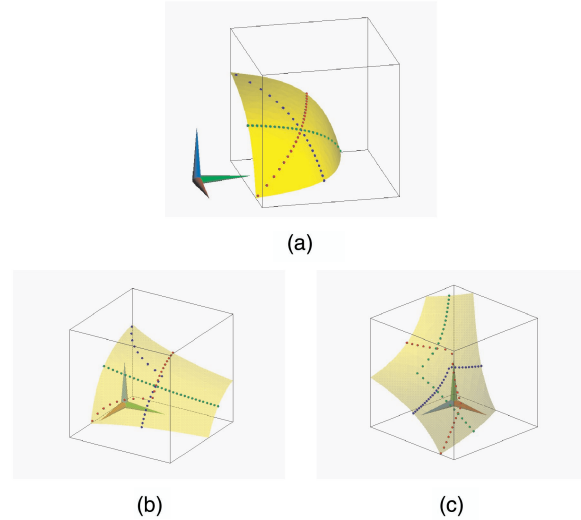


Fig. 9. Bishoulder point. (a) Shows a bishoulder point in case 1. The three colored curves on the isosurface are the loci of many shoulder points on different slices: red if considering slices of x constant, green for slices of y constant, and blue for z constant. These curves intersect each other just once and at the same point. However, (b) and (c) are two examples where more than one bishoulder point exists.

slices sweep out the isosurface and the shoulder points (we call them *slice shoulder points*) on these contour lines will sweep out a trajectory across the isosurface. This trajectory will comprise a set of points which optimally define the intersection of the isosurface with the set of parallel planes. By optimally, we mean points on the isosurface which are furthest from a surface created by sweeping out the chord joining the boundary intersections for each plane.

This can be done in any of the three coordinate directions, yielding a set of trajectories. Any point where the trajectories intersect will be simultaneously a slice shoulder point on (at least) two orthogonal slices through the cube. We term such a point a *bishoulder point* and we take this to be a significant interior point for definition of the isosurface. In Fig. 9, we illustrate the concept of a bishoulder point—for case 1, we show the loci of slice shoulder points on slices parallel to each pair of opposite faces; where they intersect is a bishoulder point.

The number of bishoulder points will vary according to the different cases. For boundary polygons $P_{3/0}^E$, there is a unique bishoulder, as shown in Fig. 9. We claim it is the optimal interior point (analogous to a shoulder point on a face) for the representation of the isosurface. We shall see in Section 5 that boundary polygons of type $P_{3/0}^E$ are the second most common in practice. However, in other cases with more complex boundary polygons, there may be more than one bishoulder—as also shown in Fig. 9. As we see later when we discuss triangulation, it is sufficient just to select one bishoulder point to aid the interior surface representation.

A bishoulder point is important for accuracy because of the optimality property just mentioned. However, it is also vital for robustness. In the limit where the threshold approaches the body saddle value, the contour lines on slices parallel to the cube faces degenerate to pairs of straight lines—three lines in all, intersecting in the body

saddle point. Thus, for this critical case of threshold equal to body saddle value, the trajectories of the slice shoulders will pass through the body saddle point. It follows that a bishoulder point will converge to the body saddle point. Thus, it is a critical point in providing robustness of representation.

In Appendix B, we discuss how an approximation to the bishoulder point can be efficiently computed.

3.4 Representing Type A.2

This is the type where we have a single surface piece, but where the piece intersects one or more faces of the cube twice. This type is not involved in any interior ambiguity (see Table 1). It is, however, highly involved in face ambiguities (see, for example, case 7.3).

This type proves a challenge in providing a good interior representation because the surface “loops back” inside the cube, to leave by the same face by which it entered. Just using boundary points is therefore insufficient to describe the nature of the surface and an interior point is needed. As explained earlier, different authors have suggested different strategies for choosing the interior point, but these are typically rather arbitrary and do not guarantee a point lying on the isosurface.

Our strategy is to identify the point at which the surface loops back toward the face by which it entered the cube. This will be the point where the surface is most distant from the face and it therefore can be regarded as an optimal point in the interior representation of the isosurface. This point will be where a slice parallel to the face is tangential to the surface. We will term such a point a *tangent point* on the isosurface.

A tangent point for slices orthogonal to the z -axis will be a point on the surface ($F(x, y, z) = 0$), where the partial derivatives with respect to x and y are zero ($F_x = F_y = 0$). We shall denote such a point as Tz . A similar definition follows for slices orthogonal to the other coordinate directions, denoting the corresponding tangent points Tx and Ty . Thus, we define:

$$Tz: F = F_x = F_y = 0, \quad (3)$$

$$Ty: F = F_x = F_z = 0, \quad (4)$$

$$Tx: F = F_y = F_z = 0. \quad (5)$$

In general, there are two tangent points for each direction, making six in all. Note, however, that these points may lie outside the cube or may be imaginary. The computation of the tangent points is given in Appendix C.

In Fig. 10, we show an instance of type A.2—it is case 3.2. The surface enters and leaves the horizontal face $z = 0$. The tangent point where the surface touches a slice parallel to the face $z = 0$ is marked, as are the contours on a set of parallel slices from $z = 0$ to $z = 1$. On the slice through the tangent point, the contours degenerate to a pair of straight lines; above and below, the contours have different topologies. It is clear that the tangent point is critical in defining the interior representation.

In cases where a surface “loops back” to each face of a unique pair of opposite faces—see case 10.2 in Fig. 3—only

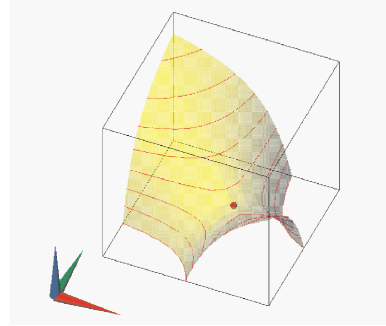


Fig. 10. Tangent point $Tz = (x_t, y_t, z_t)$ such that each of F , F_x , and F_y vanish. The figure also shows a range of (red) 2D contours parallel to the ambiguous face $z = 0$ and superimposed on the (yellow but transparent) isosurface. Tz lies on the horizontal slice $\Omega = z_t$ that behaves as a frontier between two different topologies, immediately above and below. On that slice the contours degenerate to a pair of crossing straight lines.

one tangent point is involved. This is the reason why, in our notation, we just count this pair of opposite faces once in calculating the value of N in $P_{M/N}^E$ (see Section 3.1). Thus, N denotes the number of tangent points involved.

3.5 Representing Type B

This is the case where the surface piece is homeomorphic to a cylinder. We detect the existence of a tunnel by using the body saddle test of Natarajan, as explained earlier. This is again a challenge for internal representation and no systematic approach has been proposed as far as we are aware. In our context, we shall require the representation to be visually continuous as we pass from two pieces of type A.1 to one piece of type B, as the threshold passes through the body saddle value. Fig. 7 illustrates the situation for case 4.

We are able to make further use of the tangent point concept introduced in the previous section. In Appendix C, we show that the computation of tangent points in each of the three coordinate directions reduces in each case to the solution of a quadratic equation. Hence, there are up to two tangent points as solutions in each of the three cases. In the case of a tunnel, we observe that all six points exist and, moreover, they form a cuboid which approximates the tunnel. The isosurface touches the planes defining the cuboid, at the set of six tangent points. This is shown in Fig. 7, where (c) shows the way the six tangent points delineate the shape of the tunnel, with the body saddle internal to the cuboid formed by the six points.

Indeed, because the tangent points are points where the slice contours degenerate to a pair of straight lines, we find that all edges of the cuboid lie on the surface. (A proof of the fact that the tangent points form a cuboid can be found in Lopes [10]; each pair of tangent points Tx , Ty , and Tz , lie at opposite corners of the cuboid.)

As can be seen in Appendix C, the three quadratic equations have the same discriminant—and, thus, when one equation has equal roots, so do the others; and, moreover, in that case, all tangent points are identical (the opposite corners of the cuboid converge to a single point)—and equal to the body saddle point. Thus, the tangent points converge to the body saddle point as the threshold

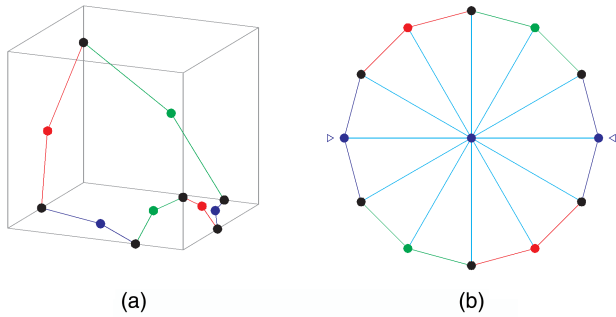


Fig. 11. Diagram showing triangulation of $P_{6/1}^E$. This is the example already shown in Fig. 10. In (a), we show the polygon on the faces of the cube. The triangulation diagram is in (b). The boundary comprises the faces: $z = 0$ (bottom), $y = 0$ (front), $x = 1$ (right), and so on. Each of x, y, z corresponds to one color: red, green, and blue, respectively. The intersection points are black spheres and the face shoulder points are in between. These are colored according to the faces upon which they lie. The tangent point in the interior inherits the color of the related face shoulder points. The triangulation is completed by cyan lines between each point in the boundary and the point in the interior. Finally, there are two small triangles indicating the face the polygon intersects twice and colored accordingly. In this and subsequent diagrams, the vertices of triangles are normally two neighboring boundary points and a point in the interior. Bishoulder points are colored magenta.

converges to the body saddle value. Therefore, the tangent points not only provide us with a means of representing the interior topology of the tunnel, they also allow us to define an interior representation that is robust and visually continuous.

The difficult case 13.5 (discussed earlier in Section 3.2 and illustrated in Fig. 8) is handled smoothly by our strategy. As the threshold passes through the lower of the two body saddle values, the lower tunnel converges to a point and two surface pieces are created. As the threshold increases through the higher of the two body saddle values, so the middle piece merges in a continuous fashion with the upper piece to form the upper tunnel.

Finally, we note that if the following conditions are satisfied: 1) The method to solve face ambiguities dictates at least two boundary polygons ($P_{M/0}^E$); 2) there are six tangent points within the cube; then we can conclude there is a tunnel.

For completeness, we give the computation of the body saddle points in Appendix D.

4 TRIANGULATION METHODOLOGY

We now discuss a strategy for creating a triangular mesh representing the isosurface, for each of the three surface types. For illustrative purposes, we use colored diagrams to depict triangulations. The caption for Fig. 11 explains in detail the notation and color scheme used in all the subsequent diagrams.

4.1 Triangulation for Type A.1

In this case, we have a boundary polygon with M points, augmented by intermediate face shoulder points. We saw earlier that bishoulder points are optimal interior points in representing the surface shape. In the case of $P_{3/0}^E$, there is only one bishoulder, but there may be several—see Fig. 9. It would be feasible, but expensive, to triangulate using all

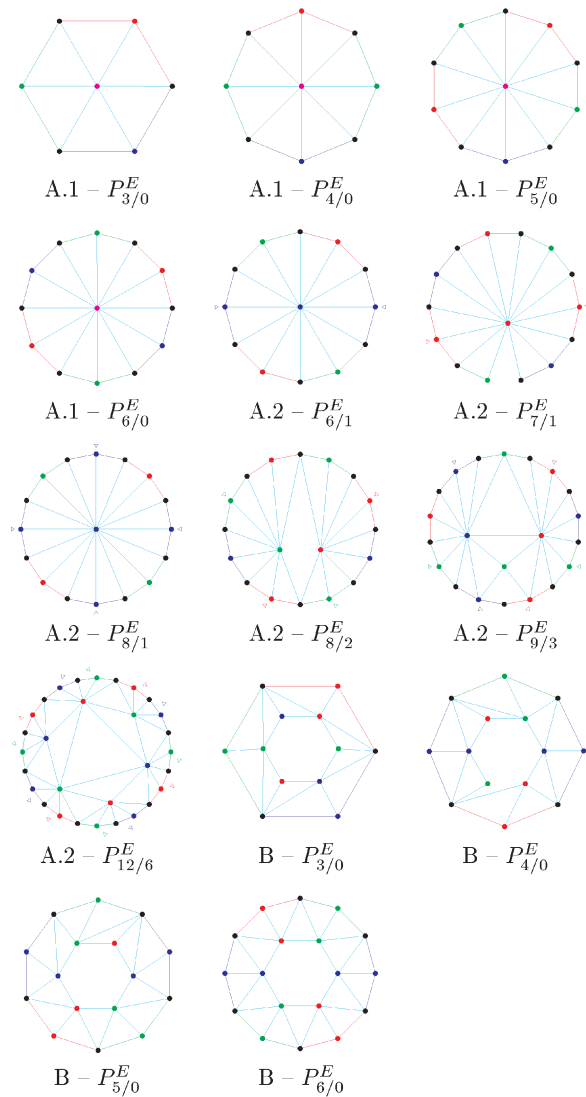


Fig. 12. Triangulation of polygons of type A.1, type A.2, and type B (polygons of type A.1 when involved in the formation of tunnels). We note that the interior area delimited by the tangent points in $P_{12/6}^E$ can be triangulated in a different order.

bishoulders, but, in practice, we have found it is sufficient to use just one, according to the following criterion. As can be seen from Fig. 9, a trajectory of slice shoulder points may run from face to edge or face to face. We prioritize trajectories which go from one face to the opposite face (and, hence, travel through all of the cube); then, trajectories from face to adjacent face; and, finally, trajectories from face to edge. We select a pair of crossing trajectories according to that rule, and use the corresponding bishoulder.

Each triangle involves the interior point and a pair of adjacent boundary points; the pattern is illustrated in Fig. 12 for all cases of type A.1.

4.2 Triangulation for Type A.2

Here, we have a boundary polygon with M points, augmented by intermediate face shoulder points, plus N interior tangent points. The strategy is similar to type A.1 and is also illustrated in Fig. 12.

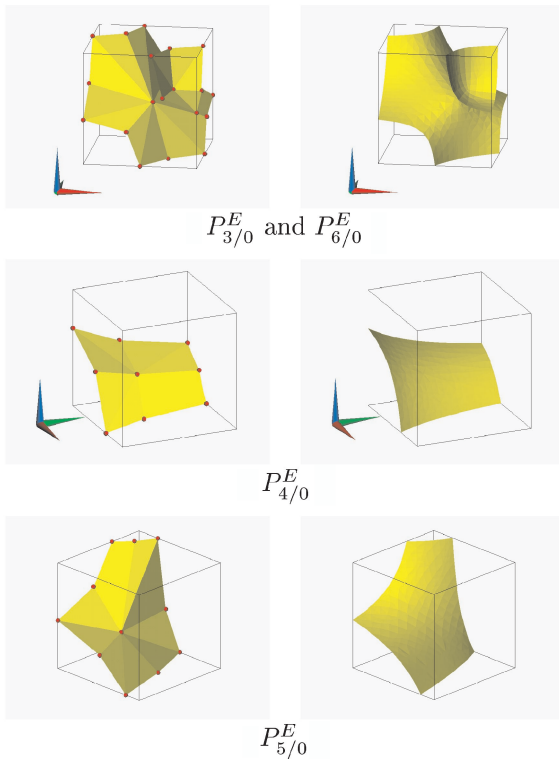


Fig. 13. Examples of triangulations of type A.1.

4.3 Triangulation for Case B

This is the case of a tunnel. We have seen that the six tangent points form a cuboid. In Appendix C, we show that it is possible to draw a polyline joining the six tangent points, forming a six-sided polygon. Topologically, we use this as an interior polygon, within the polygon formed by the boundary points (intersection points plus added face shoulder points). As mentioned earlier, the edges of this interior polygon (not just the vertices) lie on the surface and this makes it especially useful in the interior surface representation.

We illustrate this in Fig. 12. Look first at the triangulation of $P_{3/0}^E$ with the interior bishoulder point (first row). Imagine the transition to a tunnel situation, where the bishoulder point converges to the body saddle point and then the tunnel appears. The triangulation of $P_{3/0}^E$ transforms to that shown in the fourth row, where the interior six-sided polygon defines a hole. The cases of $P_{M/0}^E$ for other values of M are also shown.

In a tunnel situation, for each $P_{M\alpha/0}^E$, there will be a second $P_{M\beta/0}^E$ polygon and that will abut perfectly with its partner to form the tunnel. Both triangulations can be regarded as independent.

5 EXAMPLES

5.1 Examples within a Cell

Several representations of isosurfaces within a cell are shown in Figs. 13, 14, and 15. The order is as follows: First, our representation where the vertices of triangles (red spheres) are the points on the isosurface we use; then, on the right, the benchmark to compare results, which is the

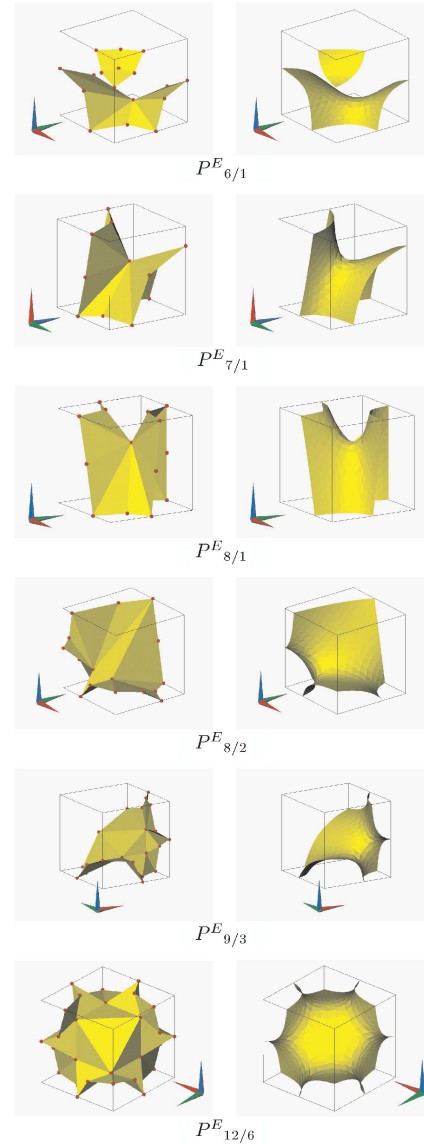


Fig. 14. Examples of triangulations of type A.2.

exact trilinear interpolant function. These images on the right are obtained when F is supersampled and then the isosurface from the new data set is extracted. The sampling factor used is 20, leading to a data set of dimension $[20 \times 20 \times 20]$, in the same interval $[0, 1]$. The data values are depicted in Table 2.

5.2 Practical Example

We analyze one example of using the method. Further examples are analyzed in Lopes [10], but the results are broadly the same. We look at a large data set of the brain, involving nearly four million cells. Fig. 16 shows an isosurface generated by the method, yielding the following results:

- The number of triangles generated is around 693,000.
- The polygons of type A.1 dominate—around 99 percent—with $P_{4/0}^E$ and $P_{3/0}^E$ the most common (50 percent and 28 percent, respectively).

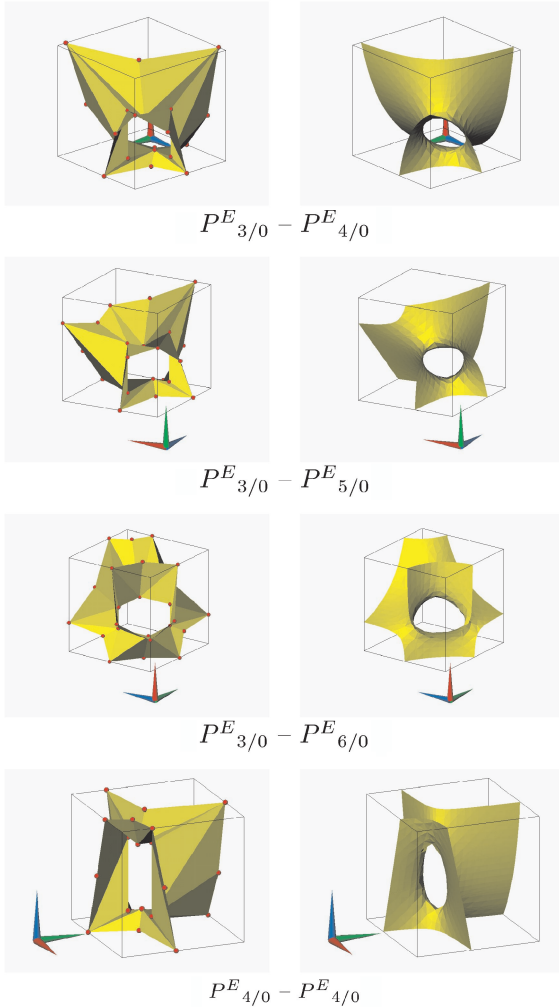


Fig. 15. Examples of triangulations of type B.

- There are few polygons of type A.2; around 0.7 percent.
- Tunnels are quite rare—12 in this data set. Thus, here, type B just amounts to 0.014 percent.

Thus, the difficult cases do not occur frequently, but they do occur. In some applications, it may be extremely important to handle them correctly.

5.3 Comparison with Other Methods

In this section, we highlight the distinguishing features of our method. First of all, to illustrate the improvement on a real data set, we zoom in on a section of the brain data. In Fig. 17, we show the surface generated by the method described in this paper and the corresponding surface generated by Marching Cubes with asymptotic decider. The exact trilinear isosurface has a staircase appearance which is preserved by our use of face shoulder and interior points. In contrast, a conventional Marching Cubes algorithm will tend to smooth out the edges in the surface.

Next, to illustrate accuracy and robustness, we show in Fig. 18 a sequence of pictures generated from the cell with data values given in the caption. With a threshold greater than 0.58, this data corresponds to case 3.1, and, for a threshold less than 0.58, it corresponds to 3.2. In each row,

TABLE 2
Data That Generates the Examples Shown in Section 5

	Figures						
	13 a	13 b	13 c	14 a	14 b	14 c	14 d
F_{000}	2.5	4.0	2.0	2.5	1.0	2.0	1.0
F_{001}	-3.0	-2.0	5.0	-2.0	-1.5	1.0	-1.0
F_{010}	3.0	5.0	3.0	-1.0	-1.5	-1.0	-1.5
F_{011}	2.5	-8.0	-6.0	-1.0	3.5	-1.0	-1.0
F_{100}	-3.0	-2.0	-2.0	-2.0	2.0	-1.0	-2.0
F_{101}	2.0	-1.0	-1.0	1.0	-1.0	-1.5	5.0
F_{110}	4.0	-8.0	-10.0	1.5	-1.0	1.0	1.0
F_{111}	-5.0	-1.0	-4.0	-1.5	-1.0	1.0	1.5
	14 e	14 f	15 a	15 b	15 c	15 d	
F_{000}	1.5	1.0	1.5	2.0	2.0	1.0	
F_{001}	-2.5	-1.0	-1.5	-1.0	-2.0	2.0	
F_{010}	-1.0	-1.0	-3.0	-1.0	-2.0	-1.0	
F_{011}	-1.0	2.0	2.0	-1.0	-2.0	-5.0	
F_{100}	-1.0	-2.5	10.0	-3.0	-4.0	-2.0	
F_{101}	1.0	1.0	-1.0	1.0	4.5	-1.0	
F_{110}	2.0	1.0	-1.0	1.0	4.5	2.5	
F_{111}	-1.0	-1.0	-1.0	5.0	-2.0	3.0	

we show (from left to right) the exact trilinear surface, the approximation generated by our method, the approximation generated by a Marching Cubes implementation with asymptotic decider, and the approximation generated by a Marching Cubes implementation without asymptotic decider. The three rows shown (from top to bottom) correspond to thresholds of 0.6, 0.5, and 0.4. In the top row, all methods generate the correct topology. However, as the threshold passes through the critical value, notice that the asymptotic decider variation generates the correct topology, but “flips” to a significantly different representation, while the original Marching Cubes algorithm does not “flip,” but generates the wrong topology. Notice also how our method through the use of face shoulder and bishoulder points—all lying on the surface—generates a more accurate representation of the shape of the exact surface than the other methods.

Generally, all methods which use the asymptotic decider for face topology, but use just one line segment in a face,

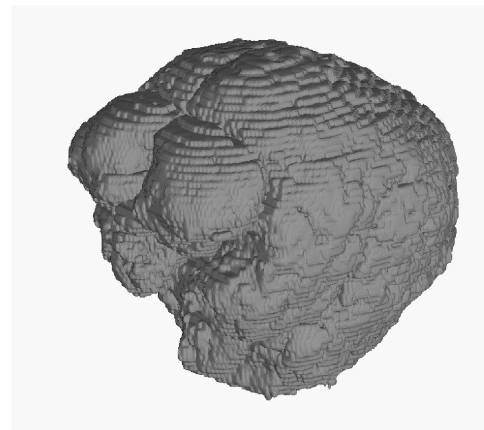


Fig. 16. Example from a data set of nearly four million cells. Our method generates around 693,000 triangles.

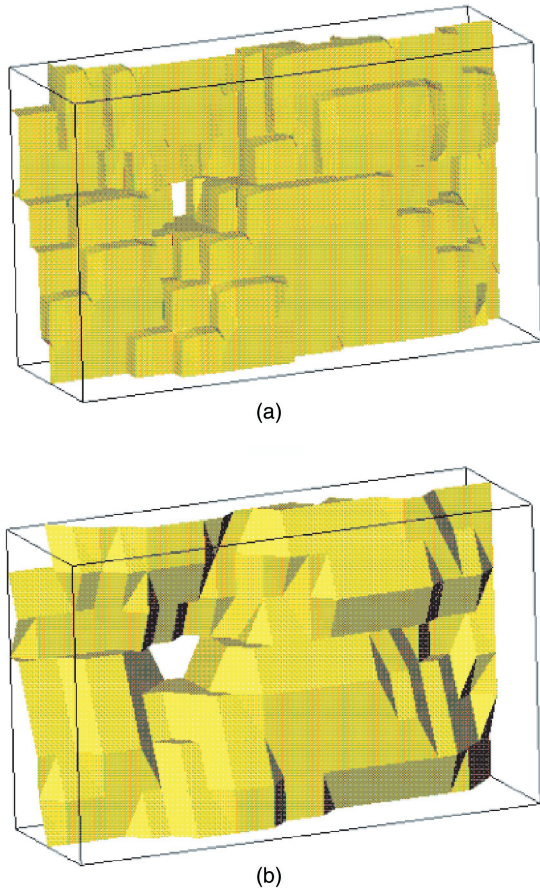


Fig. 17. Close-up of a section of the brain data set. (a) Method of this paper and (b) Marching Cubes with asymptotic decider.

will be discontinuous when the face topology changes and methods which generate correct interior topology, but restrict interior points to cell diagonals, will be discontinuous when the interior topology changes. The method described in this paper is continuous through face and interior topology changes.

There is a cost associated with the superior representation. Additional points need to be computed compared with a conventional Marching Cubes, but this computation is quite simple (see the Appendices). The significant extra cost is the increased number of triangles. For the full brain data set, our method generates approximately 693,000 triangles, against approximately 167,000 triangles for the conventional Marching Cubes.

6 CONCLUSIONS

We have proposed an approach to isosurfacing which meets the objectives set out in the introduction.

Topological Correctness. The method generates an approximation to the isosurface of the trilinear interpolant within the cube that is topologically correct.

Robustness. The representation of the isosurface is robust to small changes of data or small changes of the threshold. This means that there are no visual discontinuities when the threshold lies in the neighborhood of a saddle point value of the interpolant.

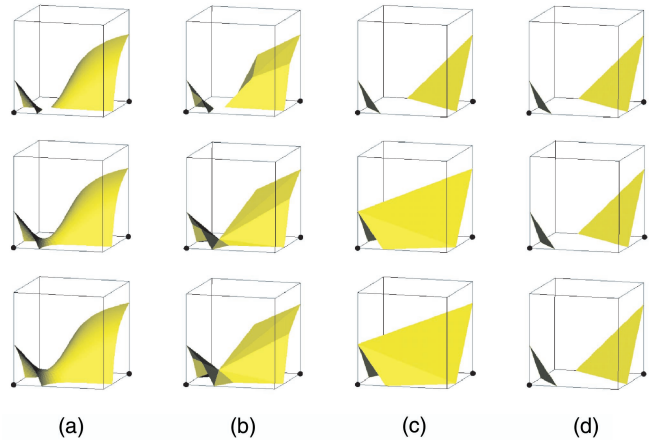


Fig. 18. Accuracy and robustness: moving from case 3.1 to 3.2. (a) Exact trilinear surface, (b) method of this paper, (c) Marching Cubes with asymptotic decider, and (d) Marching Cubes without asymptotic decider. The data set is $F_{000} = 1.5$, $F_{001} = -1.0$, $F_{010} = -1.0$, $F_{011} = -1.0$, $F_{100} = -2.0$, $F_{101} = -1.0$, $F_{110} = 5.0$, and $F_{111} = -1.0$. Top threshold = 0.6, then 0.5, then 0.4.

Efficiency. While achieving correctness and robustness, care has been taken to add as few additional points to the triangulation as possible (compared with the standard Marching Cubes algorithm).

Accuracy. The extra points which are added are chosen to lie on the isosurface of the trilinear interpolant and to be optimal in terms of accuracy of polygonal representation.

Triangulation Methodology. The methodology for triangulating the isosurface is explicit and open.

A key feature of our approach is that the classification is ultimately based on the type of boundary polygon $P_{M/N}^E$, rather than the case number. This is a considerable simplification: We just have 10 different types of boundary polygon, $P_{M/N}^E$: four with no faces intersected twice ($P_{3/0}^E$, $P_{4/0}^E$, $P_{5/0}^E$, and $P_{6/0}^E$), and six with some faces intersected twice ($P_{6/1}^E$, $P_{7/1}^E$, $P_{8/1}^E$, $P_{8/2}^E$, $P_{9/3}^E$, and $P_{12/6}^E$). The four cases with no faces intersected twice have two subcases: either with or without an interior hole (the subcases with a hole forming part of a tunnel). Thus, we have in total 14 canonical cases.

The decision process can be summarized as follows:

1. an initial classification based on relative vertex values allocates a cube to one of the original 14 Marching Cubes configurations;
2. resolving face ambiguities (according to value at face saddle point) and interior ambiguities (according to value at interior body saddle point or points in case 13.5) extends the classification to the topologically correct 31 Chernyaev configurations;
3. through Table 1, each of these cases reduces to some subset of the 14 canonical boundary polygons, each of which can be triangulated independently.

We therefore have a solution which is both simple and correct. We effectively reduce the 31 Chernyaev cases to 14 and we avoid the very large number of cases in the Cignoni et al. [5] ELUT. We claim it also yields better understanding of the topology: For example, Cignoni et al. generate four

subcases for case 6 (see Table 6 in their paper) yet topologically there are only three (see Fig. 3).

Our approach has been to assume that, for accuracy of modeling, it is sufficient to use trilinear interpolation, while, for accuracy of rendering, it is sufficient to have the correct topology, to be robust, and to show the basic shape of the surface. It is possible to seek higher accuracy for either modeling or rendering.

In the case of modeling, we can use a higher order interpolation method, such as tricubic, as described by Cohen-Or et al. [12], although this does lead to higher complexity in rendering if a polygonal representation of the isosurface is required.

High accuracy in rendering is required for applications such as virtual manufacturing, where the number of triangles generated is not an issue. The method proposed here is an excellent basis for such an application when enhanced by some triangle subdivision strategy such as suggested by Bailey [9] or Cignoni et al. [5]. Our method sets out the correct topology at minimum cost, with all points lying on the surface, as an initial approximation from which these refinement approaches can proceed. An alternative approach for higher accuracy is proposed by Hamann et al. [13], where triangular rational-quadratic Bezier patches are used to give a good approximation to the trilinear interpolant.

Finally, we observe that, in a typical isosurface extraction, most cells are empty. It is important therefore to incorporate some preprocessing step which sorts the cells according to the range of values they span. Techniques which can be used are described by Shen et al. [14] and Cignoni et al. [15]. The review paper by Brodlie and Wood [16] discusses performance issues in isosurfacing and gives a wider view of the whole field of volume visualization.

APPENDIX A

COMPUTATION OF FACE SHOULDER POINTS

In this Appendix, we show how the shoulder point of the hyperbola can be computed. This will allow us to compute the face shoulder point(s) for each face of the cube intersected by the isosurface.

Consider the bilinear interpolant on the face of a cube. Without loss of generality, we can focus on a unit square with lower left at origin. The interpolating function F can be written as:

$$F(x, y) = ax + by + cxy + d. \quad (6)$$

Consider Fig. 5. The conic $F(x, y) = 0$ intersects the edges of the square in points $P = (0, -d/b)$ and $Q = (-d/a, 0)$ and, so, M , the midpoint of \overline{PQ} , is the point $(-d/2a, -d/2b)$.

The first derivatives of F are

$$F_x = a + cy, \quad F_y = b + cx \quad (7)$$

and, so, the saddle point, S , is the point $S = (-b/c, -a/c)$.

The shoulder point R is the point where the tangent to the conic is parallel to the line \overline{PQ} as shown in Fig. 5. To compute R , first notice that the product of the two first

derivatives along the hyperbola is constant, as shown in Proposition 1.

Proposition 1 (Product $F_x F_y$ constant). *Given the hyperbola described by F of (6), the following condition holds for any point (x, y) lying on the hyperbola:*

$$F_x F_y = ab - cd. \quad (8)$$

Proof. Using (7), we obtain

$$\begin{aligned} F_x F_y &= ab + acx + bcy + c^2 xy \\ &= ab - cd, \quad \text{by (6).} \end{aligned}$$

□

The shoulder point has to lie on a line parallel to \overline{PQ} . At any point on the contour curve of $F = 0$, the gradient ∇F is always perpendicular to the curve. Therefore, the dot product $\nabla F \cdot \overrightarrow{PQ}$ is zero.

From the values of P and Q calculated earlier, we have:

$$\overrightarrow{PQ} = \left(-\frac{d}{a}, \frac{d}{b} \right) \propto (-b, a).$$

Thus, at the shoulder point R , we have:

$$-bF_x + aF_y = 0. \quad (9)$$

Then, solving (8) and (9), we have:

$$F_x = \pm \sqrt{ab - cd} \sqrt{\frac{a}{b}}, \quad F_y = \pm \sqrt{ab - cd} \sqrt{\frac{b}{a}}. \quad (10)$$

Then, from (7), we obtain the coordinates of R as:

$$x = \frac{1}{c}(F_y - b), \quad y = \frac{1}{c}(F_x - a).$$

We note that there are two solutions for R , say R and R' , (depending on the choice of sign in (10)). Both solutions R and R' give points which are collinear with M and S , with S as midpoint of the line $\overline{RR'}$. We choose R as the point between M and S .

APPENDIX B

COMPUTATION OF BISHOULDER POINTS

A bishoulder point can be computed as the intersection of three surfaces: the isosurface, described by the trilinear interpolant of (2), and two more surfaces that we now explain. Consider Fig. 19 depicting case 1 shown in Fig. 9a. A shoulder point R on a y -slice lies on the line joining the midpoint M and the saddle point S . Imagine now sweeping through the cube in the y -direction; this line \overline{MS} —which changes from slice to slice—will sweep out a surface. Similarly, with z -slices, sweeping the line from the middle point M^* to the saddle point S^* will sweep out another surface. The intersection of these two surfaces with the isosurface will give the bishoulder point. The equations for the three surfaces are easily established, but it does not appear possible to solve for their intersections analytically.

However, we can generate a good approximation to the solution of the nonlinear equations in the following way: We choose an initial y -slice (see Fig. 19), say Ψ_1 , which

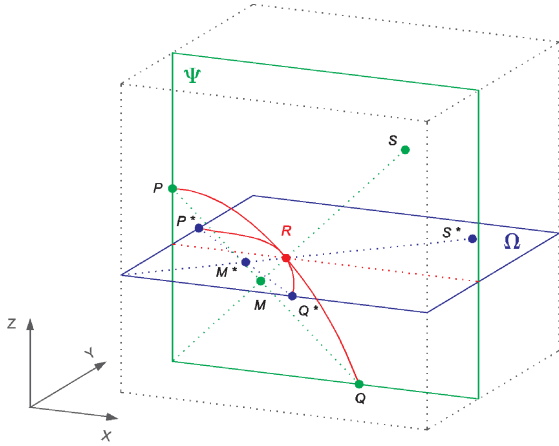


Fig. 19. Definition of a bishoulder point R . It is simultaneously a shoulder point for the contours drawn on each of the orthogonal y - and z -slices, respectively, Ψ and Ω . P and Q are intersection points, S saddle points, and M midpoints between P and Q .

passes through the face shoulder point in the $x = 0$ face. We compute the corresponding slice shoulder point R_1 . This is a first approximation to the bishoulder point, but note that, in the critical case where the threshold equals the body saddle value (transition to a tunnel), then R_1 is *exactly* the bishoulder point—it is in fact the body saddle point. Thus, a strategy of always using this as an approximation to the bishoulder will be both efficient (the cost is just the computation of the slice shoulder point) and robust. If a more accurate approximation is needed, we can iterate as follows: R_1 will lie on a z -slice, Ω_1 , on which we compute the corresponding shoulder point R_2 . If R_1 and R_2 lie within a tolerance, we are finished; otherwise, we continue, alternating between slices, until we achieve a satisfactory approximation.

APPENDIX C

COMPUTATION OF TANGENT POINTS

The first derivatives of the trilinear interpolant F of (2) are:

$$F_x = e + gy + fz + h y z, \quad (11)$$

$$F_y = c + gx + dz + h x z, \quad (12)$$

$$F_z = b + fx + dy + h x y, \quad (13)$$

and, so, with $F = F_x = 0$, $F = F_y = 0$, and $F = F_z = 0$, we obtain, respectively,

$$a + cy + bz + d y z = 0, \quad (14)$$

$$a + ex + bz + f x z = 0, \quad (15)$$

$$a + ex + cy + g x y = 0. \quad (16)$$

There are up to two solutions to consider for the tangent point Tz of (3). They are (by (11) set to zero and (15) or else by (12) set to zero and (14)):

$$x = -\frac{a + bz}{e + fz}, \quad y = -\frac{a + bz}{c + dz}, \quad z = \gamma, \quad (17)$$

where γ satisfies the second-order equation

$$(df - bh)\gamma^2 + (de + cf - bg - ah)\gamma + (ce - ag) = 0. \quad (18)$$

Similarly, for Ty of (4), we have (by (11) set to zero and (16) or else by (13) set to zero and (14)):

$$x = -\frac{a + cy}{e + gy}, \quad y = \beta, \quad z = -\frac{a + cy}{b + dy}, \quad (19)$$

with

$$(dg - ch)\beta^2 + (de + bg - cf - ah)\beta + (be - af) = 0. \quad (20)$$

And, finally, for Tx of (5), we obtain (by (12) set to zero and (16) or else by (13) set to zero and (15)):

$$x = \alpha, \quad y = -\frac{a + ex}{c + gx}, \quad z = -\frac{a + ex}{b + fx}, \quad (21)$$

with

$$(fg - eh)\alpha^2 + (cf + bg - de - ah)\alpha + (bc - ad) = 0. \quad (22)$$

It is possible to prove two results concerning the tangent points. First, if the quadratic equations have real and distinct solutions, the six tangent points form a cuboid, with the two solutions for each of Tx , Ty , and Tz at opposite corners. Second, all three quadratic equations—(18), (20), and (22)—have the same discriminant. Thus, if one equation has one double real root, so do the other two equations. (See Lopes [10] for proofs.) In this case, the opposite corners of the cuboid collapse to a single unique tangent point, which satisfies $F = F_x = F_y = F_z = 0$. Therefore, it is also a solution for a body saddle point. In conclusion, the tangent points converge to a body saddle point.

APPENDIX D

COMPUTATION OF BODY SADDLE POINTS

Setting to zero the first derivatives F_x , F_y , F_z of F of (2) shown in (11), (12), and (13), respectively, we obtain for the body saddle points

$$\begin{aligned} x &= -\frac{c + dz}{g + h z}, \\ y &= \frac{k_0 + k_1 z}{k_2}, \\ z &= -\frac{g}{h} \pm \frac{\sqrt{g^2 k_1^2 - h k_1^{1/2} (e k_2 + g k_0)}}{h}, \end{aligned}$$

with $k_0 = cf - bg$, $k_1 = df - bh$, $k_2 = dg - ch$ and where z satisfies

$$h k_1 z^2 + 2g k_1 z + (e k_2 + g k_0) = 0.$$

We note that the point $(-d/h, -f/h, -g/h)$, where all three second derivatives of the interpolant F vanish, lies

halfway between the two solutions of the body saddle points.

ACKNOWLEDGMENTS

The authors are very grateful to the referees whose insight has led to much better organization and explanation of the work. The figures were drawn using the IRIS Explorer visualization system. This new version of Marching Cubes will be available as an IRIS Explorer module from the module repository at the IRIS Explorer Centre of Excellence at the University of Leeds (<http://www.comp.leeds.ac.uk/iecoe>).

REFERENCES

- [1] W. Lorensen and H. Cline, "Marching Cubes: A High Resolution 3D Surface Reconstruction Algorithm," *Computer Graphics*, vol. 21, no. 4, pp. 163-169, 1987.
- [2] G. Nielson and B. Hamann, "The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes," *Proc. IEEE Visualization '92*, pp. 83-91, 1992.
- [3] B. Natarajan, "On Generating Topologically Consistent Isosurfaces from Uniform Samples," *The Visual Computer*, vol. 11, pp. 52-62, 1994.
- [4] E. Chernyaev, "Marching Cubes 33: Construction of Topologically Correct Isosurfaces," Technical Report CN/95-17, CERN, 1995, <http://wwwinfo.cern.ch/asdoc/psdir/mc.ps.gz>.
- [5] P. Cignoni, F. Ganovelli, C. Montani, and R. Scopigno, "Reconstruction of Topologically Correct and Adaptive Trilinear Surfaces," *Computers and Graphics*, vol. 24, no. 3, pp. 399-418, 2000.
- [6] M. Dürst, "Letters: Additional Reference to Marching Cubes," *Computer Graphics*, vol. 22, no. 2, pp. 72-73, 1988.
- [7] S. Matveyev, "Approximation of Isosurface in the Marching Cube: Ambiguity Problem," *Proc. IEEE Visualization '94*, pp. 288-292, 1994.
- [8] A. van Gelder and J. Wilhelms, "Topological Considerations in Isosurface Generation," *ACM Trans. Graphics*, vol. 13, no. 4, pp. 337-375, 1994.
- [9] M. Bailey, "Manufacturing Isovolumes," *Volume Graphics*, M. Chen, A. Kaufman, and R. Yagel, eds., pp. 79-93, Springer, 2000.
- [10] A. Lopes, "Accuracy in Scientific Visualization," PhD thesis, Univ. of Leeds, Leeds, U.K., 1999.
- [11] A. Lopes and K. Brodlie, "Accuracy in Contour Drawing," *Eurographics UK 98 Conf. Proc.*, 1998.
- [12] D. Cohen-Or, A. Kadosh, D. Levin, and R. Yagel, "Smooth Boundary Surfaces from 3D Datasets," *Volume Graphics*, M. Chen, A.E. Kaufman, and R. Yagel, eds., pp. 71-78, Springer, 2000.
- [13] B. Hamann, I. Trotts, and G. Farin, "On Approximating Contours of the Piecewise Trilinear Interpolant Using Triangular Rational-Quadratic Bezier Patches," *IEEE Trans. Visualization and Computer Graphics*, vol. 3, no. 3, pp. 215-227, 1997.
- [14] H.-W. Shen, C. Hansen, Y. Livnat, and C. Johnson, "Isosurfacing in Span Space with Utmost Efficiency," *Proc. IEEE Visualization '96*, pp. 287-294, 1996.
- [15] P. Cignoni, P. Marino, C. Montani, E. Puppo, and R. Scopigno, "Speeding Up Isosurface Extraction Using Interval Trees," *IEEE Trans. Visualization and Computer Graphics*, vol. 3, no. 2, pp. 158-170, Apr.-June 1997.
- [16] K. Brodlie and J. Wood, "Recent Advances in Volume Visualization," *Computer Graphics Forum*, vol. 20, no. 2, pp. 125-148, 2001.



Adriano Lopes graduated with the BSc degree in electrical engineering and informatics in 1986 and the MSc degree in computer science in 1993, both from the University of Coimbra, Portugal. In 1999, he received the PhD degree in scientific visualization from the University of Leeds, United Kingdom. He was a lecturer on the Faculty of Science and Technology at the University of Coimbra, where he was also a member of the Centre for Informatics and Systems (CISUC). He is currently a lecturer with the Department of Informatics of the Faculty of Science and Technology, New University of Lisbon, Portugal. His research interests are in scientific visualization and computer graphics in general.



Ken Brodlie received the BSc degree in mathematics from the University of Edinburgh in 1969 and the MSc and PhD degrees in numerical analysis from the University of Dundee in 1970 and 1973. He joined the School of Computing at the University of Leeds in 1987 and was promoted to a chair of Visualization in 2001. His teaching and research interests are in scientific visualization (with special interest in collaborative and web-based visualization and accuracy of algorithms) and web-based virtual environments (with special interest in surgical simulation). He is a fellow of Eurographics and a member of the SIGGRAPH Education Committee. At the University of Leeds, he is Director of the IRIS Explorer Centre of Excellence.

► For more information on this or any computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.