DOI: 10.1111/cgf.12975

Eurographics Symposium on Geometry Processing 2016
Maks Ovsjanikov and Daniele Panozzo
(Guest Editors)

Volume 35 (*2016*), *Number 5*

Construction of Topologically Correct and Manifold Isosurfaces

Roberto Grosso

Friedrich–Alexander–Universität Erlangen-Nürnberg, Germany

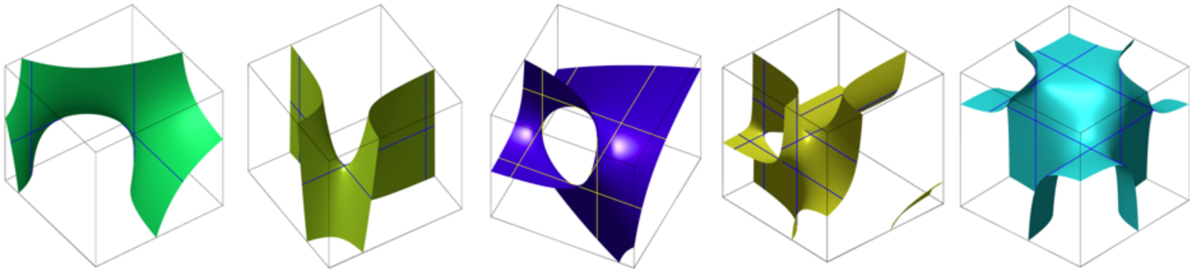


Figure 1: Some exemplary level sets of a trilinear interpolant within a unit cell

Abstract

We present a simple method to describe the geometry and topologically classify the intersection of level sets of trilinear interpolants with a reference unit cell. The solutions of three quadratic equations are used to correctly triangulate the level set within the cell satisfying the conditions imposed by the asymptotic decider. This way the triangulation of isosurfaces across cells borders is manifold and topologically correct. The algorithm presented is intuitive and easy to implement.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Geometric algorithms, languages, and systems—

1. Introduction

The marching cubes (MC) algorithm [LC87] is certainly one of the most popular technique for isosurface extraction. Its popularity relies on its simplicity, robustness and efficiency. The MC algorithm reduces the problem of determining the isosurface of a scalar field to the computation of the intersection of this surface with the gird cells. Each cell is classified according to the values stored at the grid nodes. For each class, a fixed triangulation scheme is coded in a lookup table. Using symmetries, the 256 configurations are classified into 15 different classes. Figure 2 shows the representatives of the classes. Because no other information than the data values at the grid nodes is used, MC cannot guarantee either that the extracted triangle mesh is homeomorphic to the level set of the underlying interpolant, nor that the generated triangle mesh is manifold. Soon after the appearance of the MC algorithm new methods were proposed which deal with these problems such as the well known Chernyaev's Marching Cubes 33 (MC33) [Che95].

An algorithm to compute level sets must address the following two problems: *manifoldness* and *topologically correctness*. Whereas manifoldness can be solved by a carefully triangulation of the cell satisfying the conditions imposed by the asymptotic decider [NH91], guaranteeing topological correctness is a much more difficult problem. After the Chernyaev's MC33 appeared much effort has been spent in the development of new lookup tables and deciders to tackle topological correctness, [CGMS00,Nie03,CEPS13, LB03].

In this work we propose an alternative strategy. We use a mixed or hybrid classification. Cells which correspond to non-ambiguous cases are triangulated efficiently as usual based on the standard MC classification table. For ambiguous cases we introduce a set of three quadratic equations. The solution of these equations is used to classify the topology and describe the geometry of the level set within the cell. Figure 1 shows the intersection of level sets of a trilinear function with the unit cube. Furthermore, we derive a set of few rules to triangulate the level set in a topologically correct manner satisfying the conditions imposed by the asymptotic decider, thus the triangulation is topologically correct within cells and consistent across cell borders satisfying the manifold property.

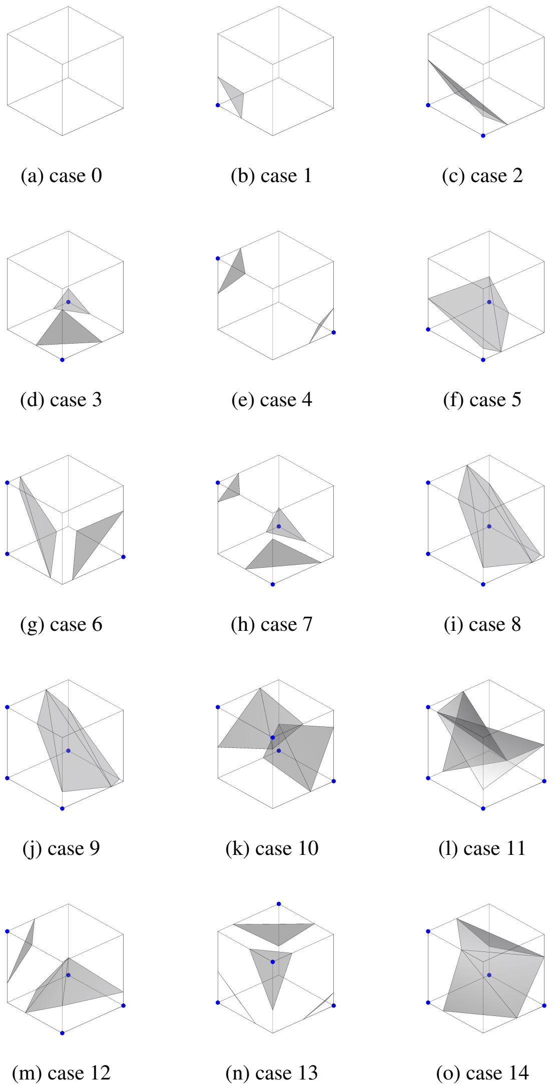


Figure 2: Basic MC cases including case 14 as in original classification. Positive vertices are marked.

The major contributions of this work can be summarized as follows:

1. Easy to evaluate criteria to classify the topology of the intersection of the level set with a cell based on the solution of three quadratic equations
2. A set of rules to triangulate ambiguous cases into a topologically correct manner guaranteeing a manifold triangle mesh
3. A triangulation scheme which introduces few additional interior vertices to keep complexity low. Additional interior vertices are saddle points or combination of saddle points of the level set within the unit cell and lie on the surface to improve triangulation accuracy.

In the following we give a short review of the literature with particular emphasis on the problem of topological correctness. In Section 3 we present the overall algorithms to construct isosurfaces. In Section 4 we develop methods to classify the level set topology within the unit cube and in Section 5 we describe a set of simple rules to triangulate grid cells in a topologically correct manner preserving manifoldness across cells border. Finally, in Section 6 we remark the differences with respect to other methods and in Section 7 we present and discuss the results.

2. Related Work

The literature dealing with isosurface extraction techniques is very extensive. In this section we mainly focus on works concerned with topological and consistency issues.

Topological Considerations: Soon after the publication of the MC algorithm, researcher started working on techniques to compute topologically correct isosurfaces. Dürst [Dür88] was the first to note that some cases in the standard MC have not a unique triangulation. The asymptotic decider of Nielson and Hamann [NH91] provides an elegant solution to cope with face ambiguity. Natarajan [Nat94] was the first to deal with the problem of the interior ambiguities and proposed to extend the MC cases by adding four *subcases*. The disambiguation was based on the analysis of face and interior critical points. Nevertheless, his method misses some interior critical points generating surfaces which are topological incorrect. Chernyaev [Che95] introduced his MC33 by extending the standard MC lookup table adding subcases into a total of 33 different triangulation schemes. He proposed an interior discriminant to resolve among the subcases. Since then much effort was put in the development of lookup tables, efficient implementations or new tests to resolve ambiguities, e.g. Matveyev [Mat99], Cignoni et al. [CGMS00], Lewiner et al. [LLVT03], or Montani et al. [MSS94], see Newman and Yi [NY06] for a survey on this topic. Nielson [Nie03] discussed how the analytic properties of the trilinear interplant and provided a set of triangulation schemes for all possible configurations. All these methods are based on lookup tables with cases and subcases, thus many decision stages have to be passed before the right triangulation scheme can be selected. e.g. according to MC33 for case 13 there are 6 subcases.

Manifoldness: Etiene et al. [ENS*12] and Custodio et al. [CEPS13] showed that manifoldness might be lost, if the topological correct triangulation of a cell introduces non-manifold edges at cell boundary. Nelson [Nie03] presented a set of lookup tables with triangulation schemes including interior vertices generating topologically correct and consistent surfaces. Lopes and Brodlie [LB03] proposed a technique to construct topologically correct isosurfaces by computing additional vertices at the boundaries and the interior of the cell to improve geometrical accuracy. Renbo et al. [RWY05] presented a method which does not need lookup tables and is able to correctly resolve internal ambiguities. These methods resemble one of the first methods to analyze scalar trivariate functions introduced by Pasko et al. [PPP88]. All these strategies are mainly focused on a classification of all possible configurations of a trilinear functions, usually by computing critical points. In this work we go a different way. Based on a set of quadratic equations we show how to first classify the topology of the intersection of the level set with the unit cube, and then we derive set of simple rules to generate a topologically correct and manifold triangulation. The resulting algorithm is more clear and easy to implement. Furthermore, additional interior vertices obtained from saddle points of the level set are introduced to increase accuracy and guarantee consistency.

Other Strategies: Dual Contouring (DC) [JLSW02, SW04] is a robust and crack-free technique for isosurface extraction, which works on the dual grid and follows a different approach than other Marching Cubes algorithms. Schaefer et al. [SJW07] addressed the problem of non-manifold surfaces and simplification; Varadhan et al. [VKKM03] was concerned with the reconstruction of thin features, and Zhang et al. [ZHK04] used DC to simplify isosurfaces while preserving topology. Kazhdan et al. [KKDH07] presented an algorithm to extract watertight isosurfaces from an octree directly from the node data and independently of the octree topology. Nevertheless, these methods are not guaranteed to preserve the topology of the isosurface extracted from a trilinear interpolant. If some knowledge on the surface being reconstructed is given, algorithms can be developed that reproduce the desired topology [BK02].

3. An algorithm for isosurface extraction

Let $\mathcal{M}$ be a hexahedral mesh and assume that data values are only known at the vertices of the cells. Let $T : \mathcal{M} \to \mathbb{R}$ be the piecewise trilinear interpolant defined on $\mathcal{M}$. Given an isovalue $i0$, the isosurface of interest is defined as:

$$S_{i0} = \left\{ (x,y,z) \in \mathbb{R}^3 : T(x,y,z) = i0 \right\},$$

The restriction of T to a cell has the form

$$\begin{aligned} F(u,v,w) = & \\ & f_0(1-u)(1-v)(1-w) + f_1 u(1-v)(1-w) \\ & + f_2(1-u)v(1-w) + f_3 uv(1-w) \\ & + f_4(1-u)(1-v)w + f_5 u(1-v)w \\ & + f_6(1-u)vw + f_7 uvw \qquad (1) \end{aligned}$$

where $(u,v,w) \in [0,1]^3$ are local coordinates and the f_i are the data values at the cell vertices, i.e. cells are mapped to a *reference unit cube*. Without loss of generality we restrict the analysis in the following to the computation of level sets within the unit cube $[0,1]^3$ and use local coordinates (u,v,w) to distinguish from the usual (x,y,z) coordinates in the 3D Euclidean space. By solving the equation $F(u,v,w) = i0$ for the trilinear function (1) the level set can be given in parametric form:

$$w(u,v) = \frac{auv + bu + cv + d}{euv + fu + gv + h}, \quad (u,v) \in [0,1]$$

with similar expressions for $u(v,w)$ or $v(u,w)$. The constant coefficients a,b,c,d,e,f,g,h depend on the f_i and on $i0$. Note that the level set of a trilinear function is not a compact closed surface. The intersection of the level set with the boundary of the unit cube is a set of up to 4 closed contours consisting of hyperbolic arcs on the six faces of the cell. The closed contours will be approximated by *closed polygons*. In the following we call these closed spatial polygons simply *contours*. The case of four contours occurs when each branch of the level set intersects a face at most once for all six faces. A vertex v_i of $\mathcal{M}$ is said to be positive with respect to the isovalue $i0$, if $T(v_i) > i0$, otherwise it is negative.

Definition 1. *A cell c_i of $\mathcal{M}$ is defined to be* unambiguous *if any pair of vertices with the same sign can be connected by a path along the edges of the cell, otherwise the cell is called* ambiguous *[GW94, ESN*09]. A* face ambiguity *occurs when face vertices have alternating signs.*

A cell might be ambiguous, even if all faces are unambiguous, e.g. the level set for case 4 can have two different topologies, it has an *interior ambiguity*, see Figure 2e and 4a.

We introduce some few more notation. The vertices of the reference cell (the unit cube) are

$$\begin{aligned} &v_0 = (0,0,0), \quad v_1 = (1,0,0), \quad v_2 = (0,1,0), \quad v_3 = (1,1,0) \\ &v_4 = (0,0,1), \quad v_5 = (1,0,1), \quad v_6 = (0,1,1), \quad v_7 = (1,1,1) \end{aligned}$$

The faces of the reference cell are named by the plane containing that face as follows:

face	plane	opposite face	plane
face 1	$w=0$	*face 2*	$w=1$
face 3	$v=0$	*face 4*	$v=1$
face 5	$u=0$	*face 6*	$u=1$

For example, the face defined by the vertices $\{v_0,v_1,v_4,v_5\}$ is called face 3, $v=0$. There are three pairs of opposite faces, these are *pair 1* with faces 1 and 2, i.e. $w=0$ and $w=1$, *pair 2* with faces 3 and 4, $v=0$ and $v=1$, and *pair 3* with faces 5 and 6, $u=0$ and $u=1$. The restriction of the trilinear interpolant to a face f is the bilinear interpolant for that face:

$$\begin{aligned} G_f(u,v) = & h_0(1-u)(1-v) + h_1 u(1-v) \\ & + h_2(1-u)v + h_3 uv \end{aligned}$$

where $\{h_0,h_1,h_2,h_3\}$ represent the data values at the vertices of the face, e.g. for face 6 these are $\{f_0,f_2,f_4,f_6\}$. The non-empty intersection of the level set with a face of the unit cube consists of one or two hyperbolic arcs. The *normal* form for the bilinear interpolant at the face is:

$$G(u,v) = \alpha + \eta(u-u_0)(v-v_0)$$

$G(u,v) = i0$ defines a *rectangular* hyperbola and the lines $u = u_0$ and $v = v_0$ are the asymptotes.

The non-empty level sets of the trilinear interpolant within the unit cube will be approximated by triangle meshes. The intersection of these meshes with the unit cube are closed polygons, the *contours* mentioned above, and the vertices of these contours are the intersection of the level set with the edges of the unit cell. In order to guarantee manifoldness across cells borders the contours must be computed using the asymptotic decider [NH91] at the ambiguous faces. Contours can have 3, 4, 5, 6, 7, 8, 9 or 12 vertices. For certain configurations the intersection of the level set with the unit cube is homeomorphic to a cylinder. We call this case a *tunnel* [LB03, Nie03], Figure 4 shows tunnels for the MC cases 4 and 13. For all other cases the surface segment is homeomorphic to a disk. Following the naming convention given in Figure 2, the ambiguous cases are *case 4*, *case 6*, *case 7*, *case 10*, *case 12*, and *case 13*. Case 4 can only have an interior ambiguity.

We propose a hybrid algorithm to compute topologically correct and manifold triangle meshes from scalar data. The main loop runs over all cells of $\mathcal{M}$. If the cell is unambiguous, it is triangulated with the standard MC algorithm. Otherwise, the intersection of the isosurface S_{i0} with the cell is computed at the cell edges. From these intersection points a set of up to four contours is generated using the asymptotic decider to preserve consistency across cells borders. The topology and the geometry of the level set within the cell is determined by solving a set of three quadratic equations, Section 4. If two contours build a tunnel, the tunnel is triangulated. The remaining contours have the topology of a disk. Based on the solution of the three quadratic equations an appropriate triangulation rule for the contours is applied. Algorithm 1 illustrates the global structure of the method.

```
Data: hexahedral mesh
Result: topologically correct and manifold triangle mesh
for each cell do
    classify cell ;              /* as in standard MC */
    if unambiguous case then
        triangulate cell with standard MC algorithm
    else
        /* based on asymptotic decider      */
        compute closed contour polygons;
        /* classify topology                */
        set and solve the three quadratic equations;
        if two contours build a tunnel then
            triangulate tunnel;
        end
        triangulate remaining closed contours;
    end
end
```

Algorithm 1: Topologically correct and manifold Marching Cubes

In the next two sections we present methods and rules to classify and triangulate contours including the case of a tunnel.

4. Level sets of trilinear functions

The intersection of the level set with the boundary of the unit cube is a set of up to 4 closed curves consisting of hyperbolic arcs. In the case of an interior ambiguity, the surface might be homeomorphic to a cylinder and we have the case of a tunnel. In this section we propose a new method to classify the topology and describe the geometry of the surface within the cell.

4.1. Topology of the level set

Let's consider first the pair of opposite faces 1 and 2. Let $G_{w=0}(u,v) = F(u,v,0)$ be the restriction of the trilinear interpolant to face 1, and $G_{w=1}(u,v) = F(u,v,1)$ the restriction to face 2. Given the isovalue $i0$, the equation for the rectangular hyperbola at face 1 is given by

$$\begin{aligned} i0 &= G_{w=0}(u,v) \\ &= (1-v)g_0(u) + vg_1(u) \end{aligned} \tag{2}$$

where

$$\begin{aligned} g_0(u) &= f_0(1-u) + f_1 u \\ g_1(u) &= f_2(1-u) + f_3 u \end{aligned}$$

and for the rectangular hyperbola at face 2 by

$$\begin{aligned} i0 &= G_{w=1}(u,v) \\ &= (1-v)\tilde{g}_0(u) + v\tilde{g}_1(u) \end{aligned} \tag{3}$$

with

$$\begin{aligned} \tilde{g}_0(u) &= f_4(1-u) + f_5 u \\ \tilde{g}_1(u) &= f_6(1-u) + f_7 u \end{aligned}$$

The hyperbolic arcs defined by equation (2) and (3) will intersect, if the following quadratic equation has a real solution for u:

$$au^2 + bu + c = 0 \tag{4}$$

with

$$\begin{aligned} a &= (f_5 - f_4)(f_0 + f_3 - f_1 - f_2) \\ &\quad - (f_1 - f_0)(f_4 + f_7 - f_5 - f_6) \\ b &= (i0 - f_0)(f_4 + f_7 - f_5 - f_6) - (f_1 - f_0)(f_6 - f_4) \\ &\quad - (i0 - f_4)(f_0 + f_3 - f_1 - f_2) + (f_5 - f_4)(f_2 - f_0) \\ c &= (i0 - f_0)(f_6 - f_4) - (i0 - f_4)(f_2 - f_0) \end{aligned}$$

If the discriminant $d = b^2 - 4ac$ is positive, the hyperbolic arcs will intersect at two different points given by:

$$u_{1,2} = \frac{b \pm \sqrt{b^2 - 4ac}}{2a}, \qquad v_{1,2} = \frac{i0 - g_o(u_{1,2})}{g_1(u_{1,2}) - g_0(u_{1,2})}$$

We associate to each solution a pair of 3D points at the planes containing the pair of opposite faces:

$$\begin{aligned} \mathbf{p}_{1,1} &= (u_1, v_1, 0), & \mathbf{p}_{1,2} &= (u_2, v_2, 0) \\ \mathbf{p}_{2,1} &= (u_1, v_1, 1), & \mathbf{p}_{2,2} &= (u_2, v_2, 1) \end{aligned}$$

The first index indicates the face and the second enumerate the solution. These points are within the faces if $(u,v) \in [0,1]^2$. At these points the trilinear and the bilinear interpolants are equal to the isovalue:

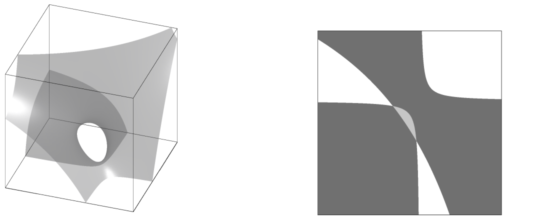
$$\begin{aligned} i0 &= F(u_{1,2}, v_{1,2}, 0) = F(u_{1,2}, v_{1,2}, 1) \\ &= G_{w=0}(u_{1,2}, v_{1,2}) = G_{w=1}(u_{1,2}, v_{1,2}) \end{aligned}$$

The interpolant F is linear along lines parallel to the coordinate axes, hence it must be constant along straight lines connecting solutions at opposite faces. For example for u_1, v_1 we have:

$$\begin{aligned} F(u_1, v_1, w) &= (1-w)G_{w=0}(u_1, v_1) + wG_{w=1}(u_1, v_1) \\ &= (1-w)i0 + wi0 = i0, \quad \forall w \end{aligned} \tag{5}$$

Therefore,

$$\frac{\partial F}{\partial w} = 0, \qquad \text{along the line} \quad \{u_1, v_1, w\} \quad \forall w \tag{6}$$



(a) case 7 (b) Intersection of the hyperbolas

Figure 3: Level set with a tunnel for case 7 and intersection of the hyperbolas at two points, gray is inside, white is outside the level set, dark gray is the parallel projection of the surface onto the face.

These straight lines lie on the level set. Similar equations can be written for the other two pair of opposite faces, i.e. faces 3 and 4, and faces 5 and 6. In the following we associate to each pair of opposite faces a quadratic equation of the form (4), and depending on the context with solution of the quadratic equation we mean the intersection point of the hyperbolic arcs or the corresponding pair the 3D points. In order to distinguish between the different pairs of points we introduce the following notation:

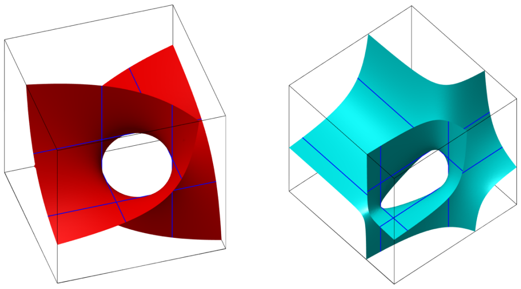
$(u_{w,i}, v_{w,i})$,	$i = 1,2$	opposite faces 1 and 2
$(u_{v,i}, w_{v,i})$,	$i = 1,2$	opposite faces 3 and 4
$(v_{u,i}, w_{u,i})$,	$i = 1,2$	opposite faces 5 and 6

Figure 3 shows a level set for case 7 and the intersection of the hyperbolic arcs from opposite faces. Chernyaev [Che95] pointed out that in order to resolve internal ambiguities hyperbolas corresponding to opposite faces can be tested for intersection. Besides this single comment we did not find in the literature any further analysis of the consequences of equation (4) on the topology and the geometry of level sets of trilinear functions. In the following we derive a set of conditions which can be used to determine the topology of the intersection of the level set with the unit cube.

Proposition 1. *The intersection of a level set of the trilinear function (1) with a unit cube is homeomorphic to a cylinder, i.e. it is a* tunnel, *if and only if for all three pairs of opposite faces the corresponding hyperbolas have two intersection points within the range* $[0,1]^2$ *and the intersection points are always at the same side of the corresponding asymptotes.*

Proof Assume that the hyperbolas of opposite faces intersect at two points within the rage $[0,1]^2$, i.e. within the faces of the unit cell, for all three pairs of opposite faces. According to equation (5) the interpolant is constant along lines connecting solutions to the quadratic equations (4) at opposite faces. Therefore, these straight lines are completely lying on the level set and at the one side is inside and at the other outside the surface, gray resp. white regions in Figure 3b. Because this is true for the three pairs of opposite faces, the level set has a tunnel within the unit cell.

Let's assume that the level set has a tunnel. For a pair of opposite faces it must be possible to *look through* the level set from one face to the other, otherwise, when moving along a line parallel to the coordinate axis the interpolant will intersect the level set twice and therefore change sign twice which is not possible for a linear function. Because the tunnel separates the space into inside and outside and it is a single piece of surface, the projection of the isosurface onto one of the faces will intersect at two points along the hyperbolic arcs, separating the empty regions into inside and outside. The intersection points must be at the same side of the asymptotes, otherwise it is not possible to move from one intersection point to the other within the empty region corresponding to the tunnel. This must be true when looking from the three main directions along the coordinate axes $\{u, v, w\}$. Therefore, the quadratic equations have 3 pairs of solutions within the range $[0,1]^2$. □



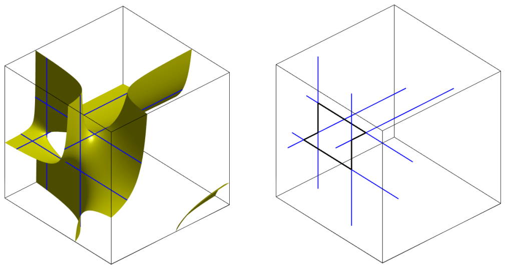
(a) case 4 with tunnel (b) case 13 with tunnel

Figure 4: The six straight lines intersect at six points.

The proposition shows that if the level set has a tunnel, there are three pairs of real solutions to the quadratic equations (4) within the face. Hence, there are three pairs of lines connecting opposite faces. These lines intersect pairwise within the unit cell at six points [Nie03, RWY05]. These six points can be connected building a closed polygonal path. In the following we call this polygonal path the *inner hexagon*, Figure 4. The inner hexagon lies on the surface and its edges are parallel to the coordinate axes.

Proposition 2. *If the level set has a tunnel within the cell, the six straight lines connecting the corresponding points at opposite faces intersect pairwise at six different points within the unit cube building the inner hexagon. Each straight line will intersect with one straight line of the other two pairs of opposite faces respectively. Once one intersection point was found, the intersection pattern, i.e. the order of the intersection points along the hexagon, is fixed and can be computed from the three pairs of solutions to the quadratic equations (4) for the hyperbolas.*

Proof The lines connecting solutions at opposite faces lie on the level set. Therefore, the hexagon obtained from the six intersection points lies on the level set as well. Consider first the pair of solutions $(u_{w,\{1,2\}}, v_{w,\{1,2\}})$ corresponding to the pair of opposite faces 1 and 2. Sort these solutions with respect to the coordinate u so that $(u_{w,1}, v_{w,1})$ has the smallest u. Thus, when moving along the u axis one crosses the level set from outside to inside at $(u_{w,1}, v_{w,1})$. Consider now the solutions $(u_{v,\{1,2\}}, w_{v,\{1,2\}})$ corresponding to the pair of opposite faces 3 and 4 and sort them again with respect to the coordinate u, hence $(u_{v,1}, w_{v,1})$ has the smallest u. Let $l_{u_{w,1}}(w) = (u_{w,1}, v_{w,1}, w)$ be the line connecting the pair of opposite faces 1 and 2 at $(u_{w,1}, v_{w,1}, 0)$ and $(u_{w,1}, v_{w,1}, 1)$. Move along this line until $w = w_{v,1}$. At this point turn and move parallel to the



(a) case 13, asymptotes (b) inner hexagon in black

Figure 5: The six straight lines intersect at six points building a closed polygonal path, which we call *inner hexagon*.

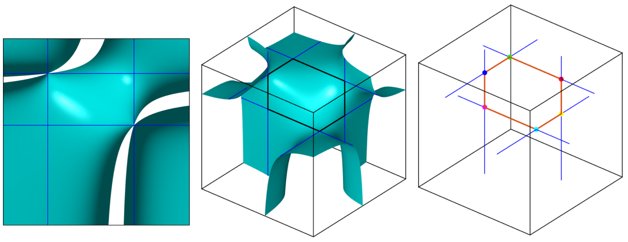
v–axis along the line $l(v) = (u_{w,1}, v, w_{v,1})$. This line will intersect the pair of opposite faces 3 and 4 at the point $(u_{w,1}, w_{v,1})$. The coordinate $w_{v,1}$ is a solution to the quadratic equation for this pair of opposite faces. It follows that $(u_{w,1}, w_{v,1})$ is on the corresponding hyperbolic arcs and is also a solution to the quadratic equation, thus $u_{w,1} = u_{v,1}$ and the line $l(v) = (u_{w,1}, v, w_{v,1})$ is one of the straight lines connecting the faces which completely lies on the level set. Applying this construction method to the remaining intersection points we can construct the inner hexagon by just reading the coordinates of the intersection points of the hyperbolic are, i.e. the solutions to the quadratic equations (4). The intersection pattern, i.e. the order in which points have to be connected to build up the hexagon, can be easily obtained by sorting the pairs of solutions to the quadratic equations with respect to one coordinate axis, Figure 5. □

Corollary 1. *If the quadratic equations have six solutions, the inner hexagon exits. Furthermore, if the corresponding branch of the level set does not build a tunnel, i.e. its intersection with the unit cube is homeomorphic to a disk, the contour has 12 vertices, and the solutions of the quadratic equations are not all at the same side of the corresponding asymptotes at the faces of the unit cube.*

Figure 6a shows the intersection of the hyperbolic arcs for MC case 13, which are at different sides of the corresponding asymptotes. The blue lines are the projections of the straight lines on the face. The region between the intersection points is not connected. Figure 6b shows case 13 without a tunnel and the intersection of the six straight lines. Figure 6c shows the inner hexagon in black with its vertices.

4.2. Level set without tunnel

The intersection with the unit cube of a branch of the level set, which is not a tunnel, has the topology of a disk. The geometry of the level set can be described by analyzing the solutions the the quadratic equations (4) for all three pairs of opposite faces. Along the straight lines connecting points at opposite faces, the trilinear function is constant. These lines are always parallel to one of the coordinate axis. If two such lines intersect within the unit cube, the intersection point is a *saddle point*, see equation (6). This fact has a strong impact on the geometry of the level set within the unit cube.



(a) hyperbolic arcs (b) level set (c) inner hexagon

Figure 6: Case 13: The hyperbolic arcs intersect at both sides of the asymptotes.

Proposition 3. *If one face is ambiguous and the two quadratic equations which are not associated to this face have one solution each, then one branch of the level set intersect this face twice, see Figures 10a, 10b.*

Proof [Sketch of a proof] Assume without loss of generality that the quadratic equations associated to the pair of opposite faces 1 and 2 and the pair of opposite faces 3 and 4 has only a single solution each within the face range. Then, there is one line connecting the solutions at faces 1 and 2 parallel to the w axis, and one line connecting the solutions at faces 3 and 4 parallel to the v-axis. These lines lie on the level set. The trilinear function is constant along the lines. Thus, the intersection of these lines is a saddle point, see Proposition 2. At this saddle point the branch of the level set *winds back* and intersects twice the face of the opposite faces 5 and 6 which is ambiguous, Figures 10a, 10b, 10d, and 10e. If both faces are ambiguous, the corresponding quadratic equation has 2 solutions and the branch of the level set intersect both faces twice, Figure 10c. □

We present more properties of the geometry of the level set within the unit cube without proof.

Corollary 2. *If only two of the three quadratic equations have respectively one single solution, and the third equation has no solution within the range $[0,1]^2$, the contour has 6 or 7 vertices. The contour intersects the ambiguous face twice (see Figures 10a, 10b).*

Corollary 3. *If the three quadratic equations have only one solution each, then the contour has 8 vertices, Figure 10d. The contour intersects two ambiguous faces twice each and the ambiguous faces share a common edge. If two of the quadratic equations have only one solution each, and the other equation has two solutions, the contour has 8 or 9 vertices. If it has 8 vertices, the ambiguous faces are opposite, Figures 10c, 10e.*

Corollary 4. *If the three quadratic equations have less than two solutions within the range $[0,1]^2$ or if only one of the equations has two solutions within this range, then the contour has 3, 4, 5 or 6 vertices and does not intersect an ambiguous face.*

Corollary 5. *If there is a tunnel and three contours, the contour with does not belong to the tunnel has 3 vertices.*

Corollary 6. *If two contours build a tunnel, one of the contours can have at most 6 vertices and the other 3 vertices.*

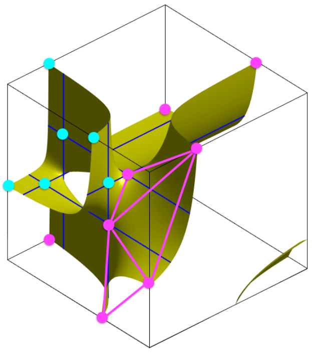


Figure 7: Correspondence between the vertices of the contours and the vertices of the inner hexagon. The triangulation of two consecutive contour vertices with the inner triangle is shown in magenta.

Corollary 7. *If there are three contours and two of them build a tunnel, the coordinates of the vertices of the remaining third contour are at one side with respect to the solutions to the three quadratic equations.*

These corollaries provide a description of the geometry of level sets within the unit cube. Based on these corollaries we introduce in the next section a set of rules for their triangulation.

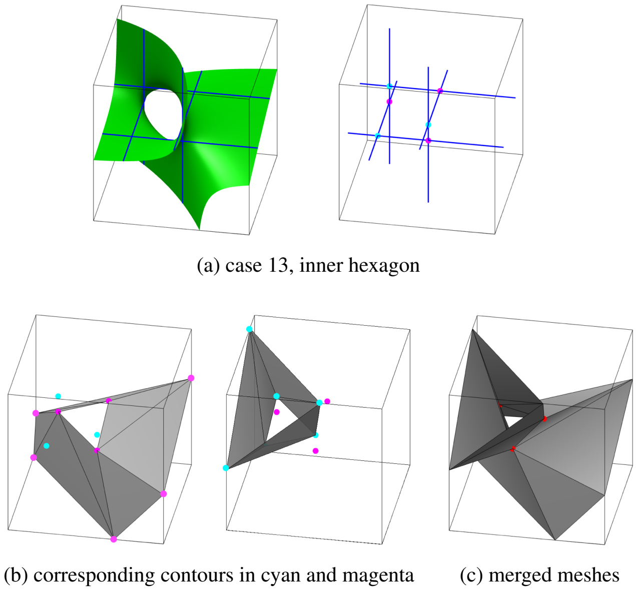
5. Topologically correct and manifold triangulation

The triangulation of level sets within the unit cube is pretty simple. There are two major cases that have to be distinguished: the three quadratic equations have 6 solutions, i.e. there is a tunnel or a contour with 12 vertices, or there are less than 6 solution which corresponds to the triangulation of a single contour into a disk like mesh.

5.1. Triangulating the tunnel

If there is a tunnel the quadratic equations have 6 solutions and the inner hexagon exists. Two branches of the level set might intersect the unit cube in this case, see Figure 5a. Discriminating this branch is pretty simple, as stated in Corollary 7. Its triangulation corresponds to the case of a single contour with the topology of a disk. The intersection of the tunnel with the unit cube consists of two closed curves, which are approximated by two contours. By a naive triangulation of these two contours into a tunnel additional edges at the faces of the unit cube will be introduced, and the triangulation will be non-manifold [CEPS13, GW94]. In order to obtain a manifold triangulation of the tunnel, an *inner contour* within the unit cube has to be constructed. The triangle mesh for the tunnel is obtained by triangulating these three contours. We construct an inner contour which has only three vertices to minimize the number of additional vertices and triangles. The additional *inner* vertices lie on the level set.

The first step by the triangulation of the tunnel is the computation of the inner hexagon. According to Proposition 2 the vertices of the inner hexagon can be determined as follows:



(a) case 13, inner hexagon

(b) corresponding contours in cyan and magenta

(c) merged meshes

Figure 8: Processing steps to correctly triangulate a tunnel

$$
\begin{array}{ll}
\text{if} \quad w_{v,1} = w_{u,1} \quad \text{then} & \\
\mathbf{p}_1 = (u_{w,1}, v_{w,1}, w_{v,1}) & \mathbf{p}_2 = (u_{w,1}, v_{u,1}, w_{v,1}) \\
\mathbf{p}_3 = (u_{w,2}, v_{u,1}, w_{v,1}) & \mathbf{p}_4 = (u_{w,2}, v_{u,1}, w_{v,2}) \\
\mathbf{p}_5 = (u_{w,2}, v_{u,2}, w_{v,2}) & \mathbf{p}_6 = (u_{w,1}, v_{u,2}, w_{v,2}) \\
\text{else if} \quad w_{v,1} = w_{u,2} & \\
\mathbf{p}_1 = (u_{w,1}, v_{w,1}, w_{v,1}) & \mathbf{p}_2 = (u_{v,1}, v_{u,2}, w_{v,1}) \\
\mathbf{p}_3 = (u_{w,2}, v_{u,2}, w_{v,1}) & \mathbf{p}_4 = (u_{w,2}, v_{u,2}, w_{v,2}) \\
\mathbf{p}_5 = (u_{v,2}, v_{u,1}, w_{v,2}) & \mathbf{p}_6 = (u_{w,1}, v_{u,1}, w_{v,2})
\end{array}
$$

We construct the three vertices of the inner contour by taking the midpoint of two consecutive vertices of the hexagon:

$$\mathbf{v}_1 = \frac{1}{2}(\mathbf{p}_1 + \mathbf{p}_2), \quad \mathbf{v}_2 = \frac{1}{2}(\mathbf{p}_3 + \mathbf{p}_4), \quad \mathbf{v}_3 = \frac{1}{2}(\mathbf{p}_5 + \mathbf{p}_6) \qquad (7)$$

The 6 vertices of the inner hexagon are saddle points. These saddle points and the corresponding straight lines connecting opposite faces *separate* the vertices of the contours. We *classify* the vertices of the contours by associating each vertex to the nearest vertex in the inner hexagon. It is easy to see, that all the vertices of one contour are assigned to three alternating vertices of the inner hexagon. The vertices of the other contour are associated to the other three vertices of the inner hexagon. In Figure 7 the mapping between vertices is colored in magenta and cyan. Each vertex of the inner hexagon has one or two nearest neighbors, e.g. in Figure 7 for the contour in cyan there are a one-to-one correspondence, the contour in magenta has 5 vertices and there are one-to-one and one-to-two correspondences. Next, a loop over each contour generates two triangle meshes. If a vertex and its consecutive vertex in the contour are assigned to the same inner vertex, we generate a triangle, otherwise we triangulate the corresponding quadrilateral. Figure 8b shows the resulting meshes. Finally, the two triangle mesh are merged by collapsing consecutive vertices of the inner hexagon according to equation (7). Figures 8a-8c demonstrate this process.

5.2. Triangulating the contour with 12 vertices

Because there can give at most 12 vertices intersecting the edges of the unit cube, a single branch of the level set intersects all the edges of the unit cube. This can happen only for MC case 13. The quadratic equations have 6 solutions and the inner hexagon exists, see Proposition 3 and Figure 6. The triangulation process is similar to the case of the tunnel. The loop runs over the single contour. At the end, vertices of the inner hexagon are merged into three inner vertices which generates the last triangle for the mesh. Figure 9 shows the level set intersecting the twelve edges of the unit cube and the corresponding triangulation.

5.3. Triangulating single closed polygons

The remaining branches of the isosurface are homeomorphic to a disk. Except for the case with 12 vertices, a contour can have between 3 and 9 vertices. The triangulation of a contour with 6 or fewer vertices satisfying the conditions of corollary 4 is equivalent to the triangulation of a flat polygon with 3, 4, 5 or 6 vertices, which can be trivially done without additional interior vertices and does not deserve any further discussion. For the triangulation of a general 3D polygon we refer to [BDE96] and [BS95].

Because a cell has 6 faces, contours with 8 or 9 vertices will intersect 2 or 3 ambiguous faces twice. Clearly, it is not possible to generate a manifold triangulation without adding an additional vertex in the interior of the cell, otherwise a third edge in one of the ambiguous faces has to be introduced, violating manifoldness across cells borders. For simplicity we present an unified triangulation scheme for ambiguous contours with 6, 7, 8 and 9 vertices: We add an additional inner vertex and triangulate the level set by connecting the vertices of the contour with this inner vertex. The additional vertex should lie on the surface and its computation must be efficient. The intersection of straight lines connecting solutions to the quadratic equations for opposite faces are saddle points, which are good candidates for the additional inner vertex. For geometric reasons we consider opposite faces for which the quadratic equation has a single solution within the face. According to corollaries 2 and 3 there can be two or three single straight lines If there are only two such lines, the additional vertex is the intersection point, see Figures 10a, 10b, 10c and 10e. If there are three such lines, the additional inner vertex is the midpoint between the two saddle points, see Figure 10d.

6. Comparison with other methods

In Lopes and Brodlie [LB03] approximation accuracy is improved by introducing additional *face* and *body shoulder* vertices which considerably increases the total number of vertices and triangles of the isosurface. The classification of the cells is based on a 3 stages decision process. First, the marching cube index is computed. Second, face and interior ambiguities are resolved and subcases are computed. Finally, subcases are mapped to one of 14 canonical boundary polygons. This process is carried out for each cell, which considerably increases computation time. Ambiguities are resolved by computing singular points of the trilinear interpolant. Our method introduces at most one additional interior vertex for ambiguous contours with 6, 7, 8 and 9 vertices and 3 additional interior vertices for a tunnel or a contour with 12 vertices. This is a very efficient and still accurate strategy. Furthermore, non-ambiguous cases are processed with the standard MC, which considerably increases the computational performance.

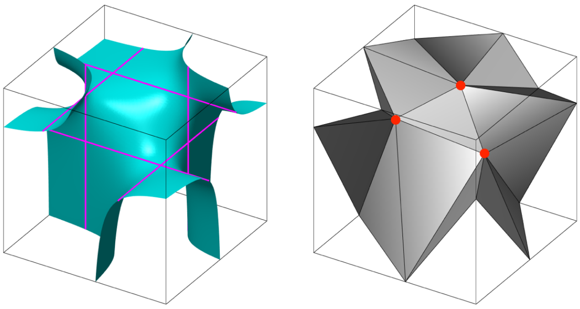


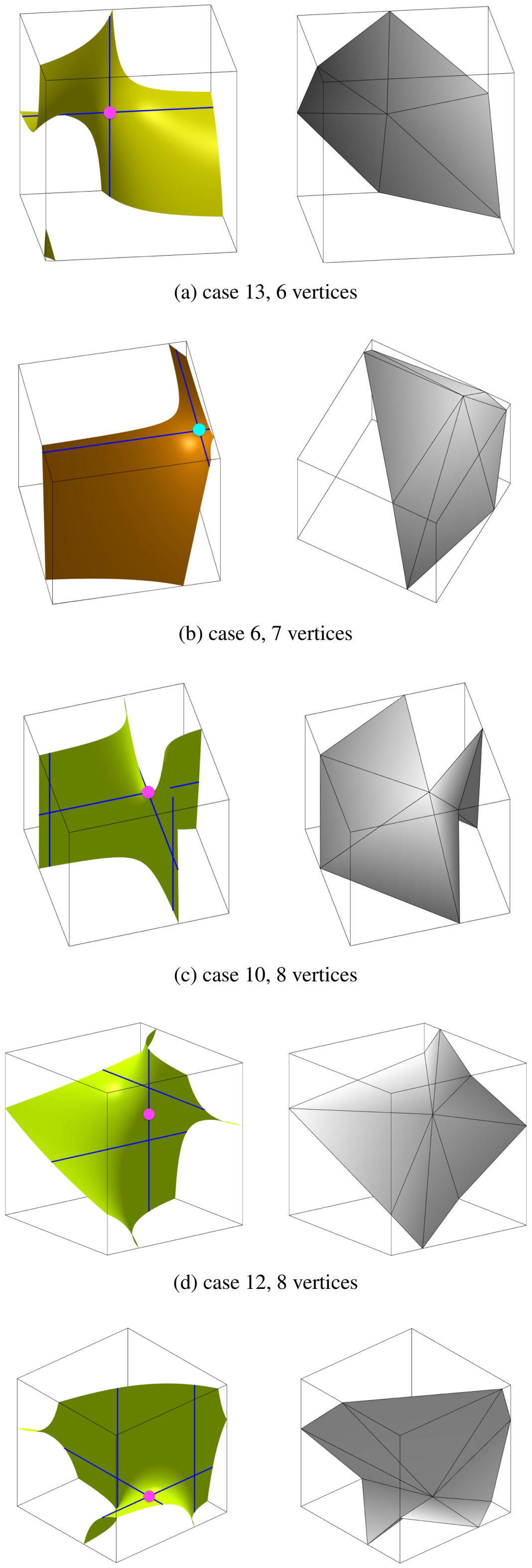
Figure 9: Triangulation of a contour with 12 vertices

The method proposed by Reno et al. [RWY05] is based on the computation and triangulation of boundary polygons. Triangulation schemes are similar to those presented by Lopes and Brodlie. In the case the intersection of the level set with the cell has two branches and one is a tunnel, i.e. there are three contours, the decider which selects the two contours building the tunnel is simply based on the Euclidean distance between the vertices of the contours and the inner hexagon. In our method, the selection of the contours constituting the tunnel is based on the geometric properties of the roots of the three quadratic equations, which delivers a theoretically founded procedure, see Corollary 7.

The algorithm proposed by Custodio et al. [CEPS13] is an improvement of Chernyaev's MC33. They do not really solve the manifold problem for case 13. Instead, they split neighbor cells to avoid consistency problems across cells borders. Our method solves manifoldness across cells based on the asymptotic decider adding additional interior vertices.

7. Results and Conclusions

The overall algorithm computes for each cell the MC index. If the cell is non-ambiguous, it is triangulated with the standard MC. If the cell is ambiguous it has to be processed separately. Following the scheme of Figure 2 the ambiguous cases are 3, 4, 6, 7, 10, 12 and 13. We computed the isosurface for two qualitative different data sets, a high resolution CT of a human skull with $512^2 \times 641$ voxels, and the Filigree data set embedded in a volume with 512^3 voxels, Figure 11. Table 1 shows the statistics counting the total number of ambiguous cells, and the total number of cases for each one of the configurations presented in the paper. The human skull has a complex topology which is reflected in the total number of ambiguous cells. The corresponding isosurface has a 2057 tunnels and 7 cases of a contour with 12 vertices. The filigree data set is



(a) case 13, 6 vertices

(b) case 6, 7 vertices

(c) case 10, 8 vertices

(d) case 12, 8 vertices

(e) case 7, 9 vertices

Figure 10: Triangulation of single ambiguous contours

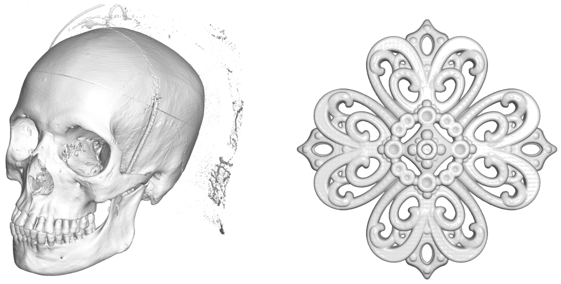


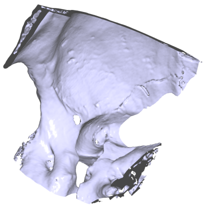
Figure 11: Isosurfaces from a CT of a human skull and the filigree data set

rather smooth and has much fewer special cases, it has no tunnel and no contour with 12 vertices. But even for this smooth data set, there are many ambiguous cases, that will generate a non-manifold triangle mesh, if ambiguous cells are not processed properly.

	CT of human skull	filigree
tot. nr. of vertices	5305717	1057602
tot. nr. of triangles	10701916	2115960
tot. nr. of ambiguous cells	296480	665
contour with 3 vts.	279196	549
contour with 4 vts.	78829	233
contour with 5 vts.	20127	0
contour with 6 vts.	1825	0
ambiguous cont. 6 vts.	41315	182
ambiguous cont. 7 vts.	49254	52
ambiguous cont. 8 vts.	14815	92
ambiguous cont. 9 vts.	1628	0
tunnels	2057	0
contour with 12 vts.	7	0

Table 1: Statistics on the ambiguous cases for the CT of a human being and the filigree data set

The algorithm we implemented computes the isosurface, generating an indexed face set data structure based on a hash table to store unique vertex indices. On a 3,5 GHz 6-Core Intel Xeon E5, 32 GB 1866 MHx DDR3 memory the computation of the isosurface for the human skull with the standard MC took 13613 ms, and with the method proposed in this paper 14021 ms, which means that for the complex topology of the human skull our method is only 3% slower than the standard MC. Testing manifoldness is pretty simple. The isosurfaces from the skull and the filigree data sets are closed. Therefore, a manifold and consistent triangulation will not have boundary edges. We computed the isosurface for the human skull for different isovalues and checked if the obtained triangle meshes had boundary edges. We repeated this procedure for the filigree. In all cases our method generated triangle meshes without boundary edges. Testing topology correctness is more difficult. We extracted a part of the human skull consisting of 128^3 voxels, starting at voxel $(64, 256, 384)$. The isosurface has 16 tunnels and 1 contour with 12 vertices.



We uniformly refined the data set using trilinear interpolation into two volumes of resolution 254^3 and 508^3. For all three data sets our method generated isosurface with the same number of connected components, the same number of boundaries and Euler characteristics for each component. We note, that after the first refinement step, there were no more contours with 12 vertices and only 5 tunnels. After two refinement steps there was only 1 tunnel left.

In this work we were concerned with the problem of generating manifold and topologically correct isosurfaces from a hexahedral mesh under the assumption of a trilinear interplant to reconstruct the scalar data. The algorithm proposed heavily depends on the solution of three quadratic equations, one for each pair of opposite faces in the cell. By an analysis of the solutions of these equations we classify the cell topology and derive a triangulation rule which guarantees a correct topology of the level set within the cell and manifoldness across cells borders.

References

[BDE96] BAREQUET G., DICKERSON M., EPPSTEIN D.: On triangulating three-dimensional polygons. In *Proceedings of the Twelfth Annual Symposium on Computational Geometry* (New York, NY, USA, 1996), SCG '96, ACM, pp. 38–47. 8

[BK02] BISCHOFF S., KOBBELT L. P.: Isosurface reconstruction with topology control. In *Computer Graphics and Applications, 2002. Proceedings. 10th Pacific Conference on Computer Graphics and Applications* (2002), pp. 246–255. 3

[BS95] BAREQUET G., SHARIR M.: Filling gaps in the boundary of a polyhedron. *Computer Aided Geometric Design 12*, 2 (1995), 207 – 229. 8

[CEPS13] CUSTODIO L., ETIENE T., PESCO S., SILVA C.: Practical considerations on marching cubes 33 topological correctness. *Computers Graphics 37*, 7 (2013), 840 – 850. 1, 2, 7, 8

[CGMS00] CIGNONI P., GANOVELLI F., MONTANI C., SCOPIGNO R.: Reconstruction of topologically correct and adaptive trilinear isosurfaces. *Computers Graphics 24* (2000), 399–418. 1, 2

[Che95] CHERNYAEV E. V.: *Marching Cubes 33: Construction of Topologically Correct Isosurfaces*. Tech. rep., 1995. 1, 2, 5

[Dür88] DÜRST M. J.: Re: Additional reference to "marching cubes". *SIGGRAPH Comput. Graph. 22*, 5 (Oct. 1988), 243–. 2

[ENS*12] ETIENE T., NONATO L., SCHEIDEGGER C., TIENRY J., PETERS T., PASCUCCI V., KIRBY R., SILVA C.: Topology verification for isosurface extraction. *IEEE Transactions on Visualization and Computer Graphics* (2012), 952–965. 2

[ESN*09] ETIENE T., SCHEIDEGGER C., NONATO L., KIRBY R., SILVA C.: Verifiable visualization for isosurface extraction. *Visualization and Computer Graphics, IEEE Transactions on 15*, 6 (Nov 2009), 1227–1234. 3

[GW94] GELDER A. V., WILHELMS J.: Topological considerations in isosurface generation. *ACM Transactions on Graphics 13*, 4 (1994), 337–375. 3, 7

[JLSW02] JU T., LOSASSO F., SCHAEFER S., WARREN J.: Dual contouring of hermite data. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2002)* (2002), ACM Press, pp. 339–346. 3

[KKDH07] KAZHDAN M., KLEIN A., DALAL K., HOPPE H.: Unconstrained isosurface extraction on arbitrary octrees. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (2007), pp. 125–133. 3

[LB03] LOPES A., BRODLIE K.: Improving the robustness and accuracy of the marching cubes algorithm for isosurfacing. *IEEE Transactions on Visualization and Computer Graphics 9* (2003), 2003. 1, 2, 3, 8

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)* (1987), pp. 163–169. 1

[LLVT03] LEWINER T., LOPES H., VIEIRA A. W., TAVARES G.: Efficient implementation of marching cubes' cases with topological guarantees. *Journal of Graphics Tools 8*, 2 (2003), 1–15. 2

[Mat99] MATVEYEV S. V.: Marching cubes: surface complexity measure. In *Proceedings of SPIE - The International Society for Optical Engineering* (1999), vol. 3643, pp. 220–225. 2

[MSS94] MONTANI C., SCATENI R., SCOPIGNO R.: A modified look-up table for implicit disambiguation of marching cubes. *The Visual Computer 10*, 6 (1994), 353–355. 2

[Nat94] NATARAJAN B. K.: On generating topologically consistent isosurfaces from uniform samples. *The Visual Computer 11*, 1 (1994), 52–62. 2

[NH91] NIELSON G. M., HAMANN B.: The asymptotic decider: resolving the ambiguity in marching cubes. In *Visualization '91* (1991), pp. 83–91. 1, 2, 3

[Nie03] NIELSON G. M.: On marching cubes. *IEEE Transactions on Visualization and Computer Graphics 9*, 3 (2003), 283–297. 1, 2, 3, 5

[NY06] NEWMAN T. S., YI H.: A survey of the marching cubes algorithm. *Computers & Graphics 30*, 5 (2006), 854 – 879. 2

[PPP88] PASKO A. A., PILYUGIN V. V., POKROVSKI V. V.: Geometric modeling in the analysis of trivariate functions. *Computers and Graphics*, 12 (1988), 3–4. 2

[RWY05] RENBO X., WEIJUN L., YUECHAO W.: A robust and topological correct marching cube algorithm without look-up table. In *Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on* (Sept 2005), pp. 565–569. 2, 5, 8

[SJW07] SCHAEFER S., JU T., WARREN J.: Manifold dual contouring. *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS 13*, 3 (2007). 3

[SW04] SCHAEFER S., WARREN J.: Dual marching cubes: Primal contouring of dual grids. In *Computer Graphics and Applications, 2004. PG 2004. 12th Pacific Conference on* (2004), IEEE Computer Society, pp. 70–76. 3

[VKKM03] VARADHAN G., KRISHNAN S., KIM Y. J., MANOCHA D.: Feature-sensitive subdivision and isosurface reconstruction. In *Visualization, 2003. VIS 2003. IEEE* (Oct 2003), pp. 99–106. 3

[ZHK04] ZHANG N., HONG W., KAUFMAN A.: Dual contouring with topology-preserving simplification using enhanced cell representation. In *Proceedings of the Conference on Visualization '04* (2004), pp. 505–512. 3