CHAPTER 3

DUAL CONTOURING

Dual contouring[1] is an isosurface construction technique that places isosurface vertices inside mesh elements instead of on mesh edges. Isosurface vertices in adjacent mesh elements are connected by isosurface edges and facets.

In the simplest form, each mesh element intersected by the isosurface contains exactly one isosurface vertex. In more complicated versions, multiple isosurface vertices can appear within a single mesh element. Each such isosurface vertex lies on a different connected component of the mesh element's isosurface patch.

Dual contouring on regular grids produces quadrilateral patches instead of triangles. The ability to place a vertex anywhere within a mesh element allows better representation of sharp vertices and edges. The simpler versions of dual contouring, with one vertex per mesh element, extend easily to higher dimensional grids and multiresolution grids, as discussed in Sections 6.7 and 10.2. On the other hand, the placement of isosurface vertices is less precise in dual contouring. When dual contouring uses only one isosurface vertex per mesh element, the isosurface is usually not a manifold.

We present here two dual contouring algorithms, SURFACE NETS and DUAL MARCHING CUBES. SURFACE NETS is the first and simplest dual contouring algorithm, where each mesh element contains at most one isosurface vertex. DUAL MARCHING CUBES is a more sophisticated version, where an isosurface vertex is generated for each connected component in a mesh element's isosurface patch.

[1] Dual contouring constructs an isosurface that is a subset of the dual mesh of the original mesh, hence the name "dual contouring."

3.1 Definitions

Positive and negative vertices and faces. The definitions of positive, strictly positive, and negative vertices and edges given here are the same as in Section 2.1. However, the definitions are extended to grid facets and grid cubes.

Definition 3.1.

- A grid vertex is positive, "+", if its scalar value is greater than or equal to the isovalue σ.
- A grid vertex is negative, "−", if its scalar value is less than σ.
- A positive vertex is strictly positive if its scalar value does not equal to σ.

Definition 3.2.

- A grid face (vertex, edge, facet, or cube) is positive if all of its vertices are positive.
- A grid face is negative if all of its vertices are negative.
- A positive grid face is strictly positive if all of its vertices are strictly positive.

The definitions given in the beginning of this section apply not just to regular scalar grids but also to curvilinear grids. They also apply to the vertices and faces of polyhedral meshes such as tetrahedral and simplicial meshes.

Bipolar edges and active faces. The definitions of bipolar edges is the same as in Section 2.1. We add a similar term for grid squares, cubes, and their faces that contain one negative and one positive vertex.

Definition 3.3.

- A grid edge is bipolar if one endpoint is positive and one endpoint is negative.
- A grid square or cube or any face of a square or cube is active if it has at least one negative and one positive vertex.

Interior and boundary edges and cubes. We need to distinguish grid edges and cubes that are on the boundary of the grid from edges and cubes that are in the interior.

Definition 3.4.

- An interior grid edge is a grid edge that is contained in four grid cubes.
- A boundary grid edge is a grid edge that is contained in three or fewer grid cubes.

Definition 3.5.

- An interior grid cube is a grid cube whose edges are all interior grid edges.
- A boundary grid cube is a grid cube that contains at least one boundary grid edge.

Let Γ be a regular grid in $\mathbb{R}^3$. We use the notation Γ_{inner} to denote the regular subgrid of Γ consisting of the interior grid cubes of Γ. We let Γ_{outer} represent the st of boundary grid cubes of Γ.

Centroid. Let $P = \{p_1, p_2, \ldots, p_k\}$ be a set of k points in $\mathbb{R}^2$ or $\mathbb{R}^3$.

Definition 3.6. The centroid of P is

$$\frac{p_1 + p_2 + \ldots + p_k}{k}.$$

The centroid of p represents a point in the "center" of P.

3.2 Surface Nets

Gibson described the first dual contouring algorithm for regular grids, which she called Surface Nets [Gibson, 1998a, Gibson, 1998b]. We first describe a two-dimensional version of her algorithm.

3.2.1 2D Surface Nets

Let Γ be a two-dimensional regular grid with scalar values assigned to the vertices of Γ. Add an isocontour vertex $w_{\mathbf{c}}$ to each active grid square $\mathbf{c} \in \Gamma$. For each bipolar grid edge separating two squares, $\mathbf{c}$ and $\mathbf{c}'$, connect vertices $w_{\mathbf{c}}$ and $w_{\mathbf{c}'}$ by an isocontour edge (Figure 3.1(c)).

We would like to position the isocontour vertex within each grid square at the "center" of the isocontour patch in the grid square. To do so, we use linear interpolation as described in Section 1.7.2 to approximate the intersection of the isocontour and all the bipolar edges and then take the centroid of all the approximation points.

More specifically, let σ be the isovalue and s_p be the scalar value at a grid vertex p. For each bipolar edge $\mathbf{e} = [p, q]$, let $w_{\mathbf{e}}$ be the point $(1 - \alpha)p + \alpha q$ where $\alpha = (\sigma - s_p)/(s_q - s_p)$. If $w_{\mathbf{e}_1}, \ldots, w_{\mathbf{e}_k}$ are the interpolation points in the grid square $\mathbf{c}$, then locate vertex $w_{\mathbf{c}}$ at $(w_{\mathbf{e}_1} + \cdots + w_{\mathbf{e}_k})/k$, the centroid of $w_{\mathbf{e}_1}, \ldots, w_{\mathbf{e}_k}$.

The 2D Surface Nets algorithm is presented in Figure 3.2.

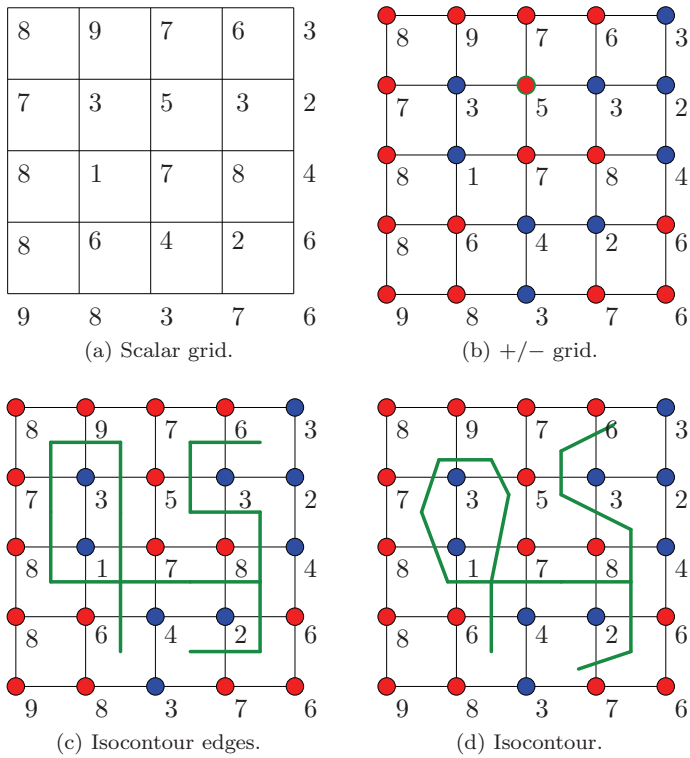


Figure 3.1. (a) 2D scalar grid. (b) Positive (red) or negative (blue) labels on grid vertices. Vertex v with scalar value s_v is positive (red) if $s_v \geq 5$ and negative (blue) if $s_v < 5$. (c) SURFACE NETS isocontour with vertices at square centers (before computing centroids). (d) SURFACE NETS isocontour with isovalue 5.

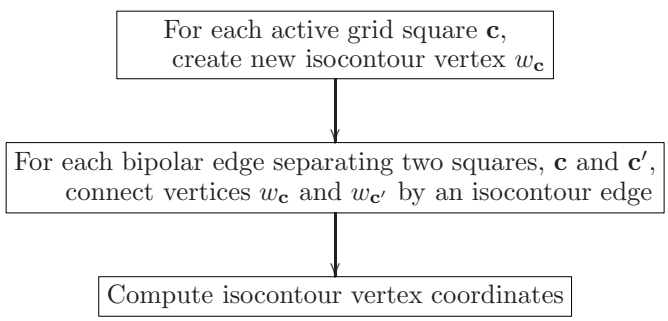


Figure 3.2. SURFACE NETS 2D.

Figure 3.1(d) contains an example of the isocontour produced by the algorithm. Note that an isocontour vertex may be incident on four isocontour edges. Thus the isocontour is not a simple closed curve or a 1-manifold.

3.2.2 3D Surface Nets

The three-dimensional Surface Nets algorithm is a direct generalization of the two-dimensional version. (See Figure 3.3.) Let Γ be a three-dimensional regular grid with scalar values assigned to the vertices of Γ. Add an isosurface vertex $w_{\mathbf{c}}$ to each grid cube $\mathbf{c}$ with at least one positive and at least one negative vertex. For each bipolar grid edge contained in four cubes, $\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{c}_3$, and $\mathbf{c}_4$, add an isosurface quadrilateral with vertices $w_{\mathbf{c}_1}$, $w_{\mathbf{c}_2}$, $w_{\mathbf{c}_3}$, and $w_{\mathbf{c}_4}$. The cyclic order of the vertices around the quadrilateral should match the cyclic order of the cubes around the edge.

To position the isosurface vertex within each grid cube, use linear interpolation as described in Section 1.7.2 to approximate the intersection of the isosurface and all the bipolar edges. For each bipolar edge $\mathbf{e} = [p, q]$, let $w_{\mathbf{e}}$ be the point $(1-\alpha)p+\alpha q$ where $\alpha = (\sigma - s_p)/(s_q - s_p)$. Take the centroid, $(w_{\mathbf{e}_1} + \cdots + w_{\mathbf{e}_k})/k$, of all the approximation points as the location of the isosurface vertex.

Figure 3.4 contains examples of the isosurface produced by the algorithm. If every grid edge is a bipolar edge (see configuration 4F in Figure 3.5), then an isosurface vertex may be incident on twelve isosurface quadrilaterals. Thus the isosurface is not necessarily a 2-manifold with boundary.

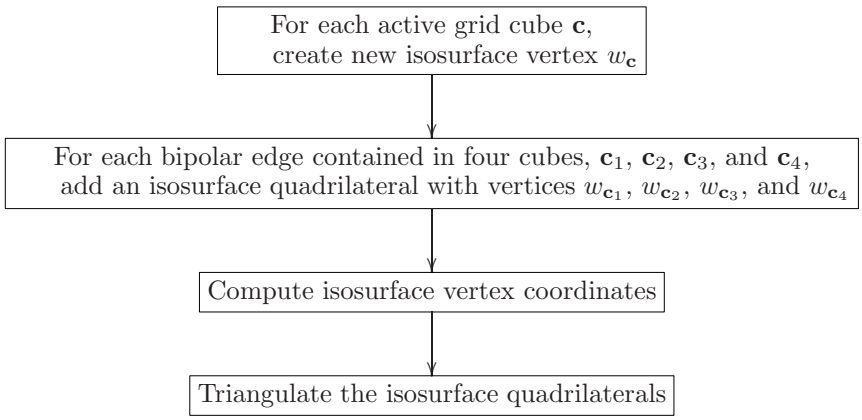
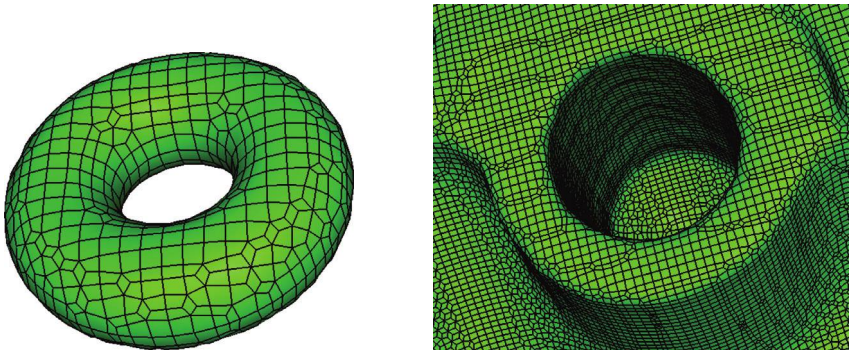


Figure 3.3. Surface Nets 3D.



(a) (b)

Figure 3.4. Examples of isosurfaces and quadrilateral meshes produced by SurfaceNets. (a) Isosurface (isovalue 3) forming a torus. Scalar data set is a $20 \times 20 \times 20$ regular grid with origin $(0, 0, 0)$ measuring the distance to a circle with radius 6 centered at $(9.5, 9.5, 9.5)$. (b) Isosurface (isovalue 80) of CT scan of an engine block (close-up). Data set created by General Electric.

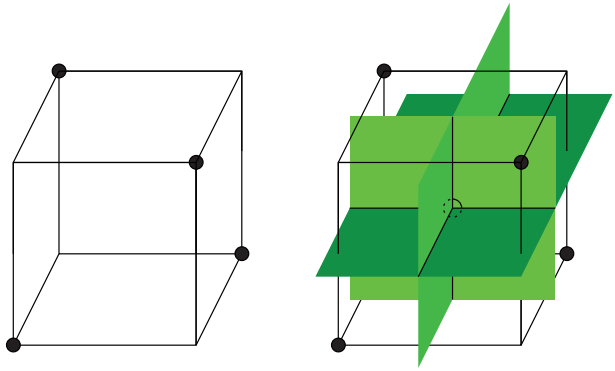


Figure 3.5. Configuration 4F. The isosurface vertex is incident on 12 isosurface quadrilaterals. The isosurface is not a manifold.

The algorithm just described uses the intersection of the isosurface and the bipolar edges to position the isosurface vertex within a grid cube. If the surface normals at the intersection points can also be computed, then these normals can be used in determining the location of the isosurface vertex. In particular, the normals can position isosurface vertices on sharp corners or edges in the isosurface, to accurately represent such corners or edges [Ju et al., 2002, Zhang et al., 2004].

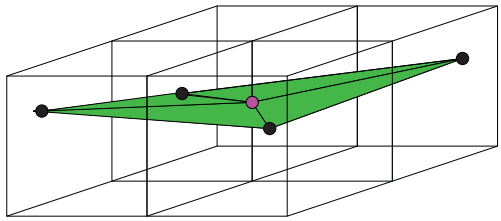
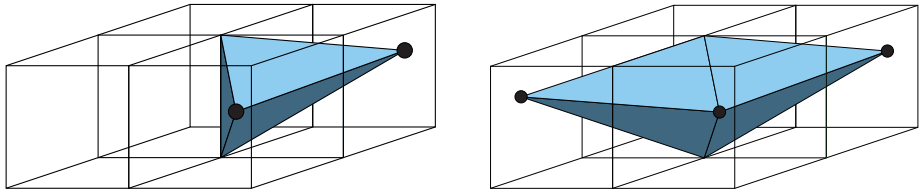


Figure 3.6. Triangle patch of four triangles whose boundary is a quadrilateral $\mathbf{q}$ around bipolar edge $\mathbf{e}$. Each triangle is formed by joining $w_{\mathbf{e}}$ (magenta vertex) to one of the four edges of $\mathbf{q}$.



(a) Tetrahedron. (b) Envelope of $\mathbf{e}$.

Figure 3.7. (a) Convex hull of bipolar edge $\mathbf{e}$ and a quadrilateral edge. (b) Union of the four tetrahedra around $\mathbf{e}$ forming the envelope of $\mathbf{e}$.

3.2.3 Triangulation

Dual contouring creates a quadrilateral mesh but these quadrilaterals are not necessarily planar, i.e., their four vertices do not necessarily lie on a single plane. In fact, unless isosurface vertices are placed at the center of each grid cube, it is highly unlikely that the four quadrilateral vertices will lie on a single plane. To create a piecewise linear surface, one needs to replace each quadrilateral with a patch of triangles whose boundary is the quadrilateral.

The simplest approach is to add a diagonal to each quadrilateral, splitting it into two triangles. Unfortunately, adding diagonals can create self-intersections in the isosurface [Ju and Udeshi, 2006]. The problem is that a diagonal from one quadrilateral can intersect the triangles of another quadrilateral.

Ju and Udeshi [Ju and Udeshi, 2006] solve this problem by breaking a quadrilateral into four, not two triangles. Consider the bipolar edge $\mathbf{e}$ contained in four cubes, $\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{c}_3$, and $\mathbf{c}_4$, listed in order around $\mathbf{e}$. Let $w_{\mathbf{e}}$ be the point on $\mathbf{e}$ constructed using linear interpolation as described in the previous section. The quadrilateral around $\mathbf{e}$ has four edges. Replace this quadrilateral with four triangles formed from joining $w_{\mathbf{e}}$ to each of the four edges (Figure 3.6).

Why does this new triangulation not have self-intersections? The convex hull of quadrilateral edge $(w_{\mathbf{c}_1}, w_{\mathbf{c}_2})$ and $\mathbf{e}$ is a tetrahedron (Figure 3.7(a)). Similarly,

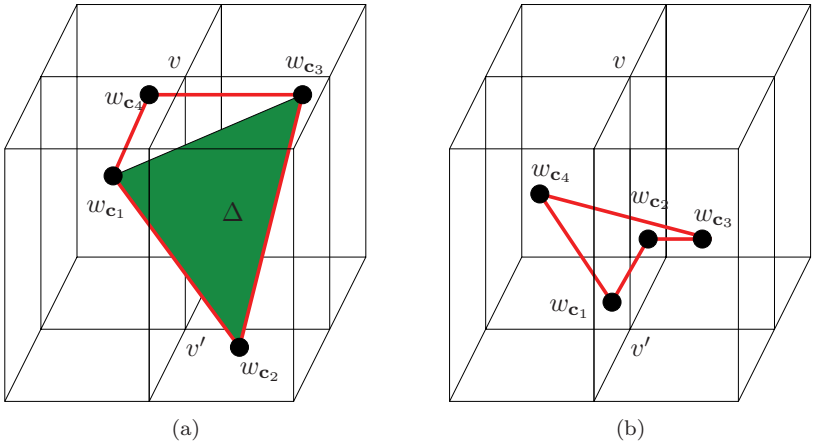


Figure 3.8. Isosurface triangles outside the envelope of (v, v'). Quadrilateral edges are colored red. (a) Plane containing triangle $\Delta = (w_{\mathbf{c}_1}, w_{\mathbf{c}_2}, w_{\mathbf{c}_3})$ does not strictly separate v from v'. (b) Plane containing v and quadrilateral diagonal $(w_{\mathbf{c}_1}, w_{\mathbf{c}_3})$ does not strictly separate $w_{\mathbf{c}_2}$ from $w_{\mathbf{c}_4}$.

the convex hull of the other quadrilateral edges and $\mathbf{e}$ are also tetrahedra. The union of these four tetrahedra is a polyhedron with eight triangular faces (Figure 3.7(b)), although this polyhedron is not necessarily convex. Ju and Udeshi call this polyhedron the envelope of $\mathbf{e}$. The four triangles sharing vertex $w_{\mathbf{e}}$ are contained in the envelope of $\mathbf{e}$.

The envelopes of two different grid edges intersect only on shared vertices, edges, or facets. Since the four triangles sharing vertex $w_{\mathbf{e}}$ are contained in the envelope for $\mathbf{e}$, they only intersect triangles from other quadrilaterals on shared vertices or edges. Thus, the resulting triangulated surface has no self-intersections. In contrast, the two triangles formed by adding a diagonal to the quadrilateral around $\mathbf{e}$ may not be contained in the envelope of $\mathbf{e}$ and may intersect the interior of triangles from other quadrilaterals.

Adding four triangles per quadrilateral instead of two increases the number of triangles representing the isosurface by a factor of two. If the two triangles formed by a diagonal lie inside the envelope, then there is no need to use four triangles. By the argument above, they will only intersect triangles from other quadrilaterals on shared edges or vertices.

Orientation tests can be used to determine if the two triangles formed by a diagonal lie inside the envelope of $\mathbf{e}$ [Wang, 2011]. Without loss of generality, assume the diagonal is $(w_{\mathbf{c}_1}, w_{\mathbf{c}_3})$. Let v and v' be the endpoints of $\mathbf{e}$. The two triangles $\Delta_1 = \Delta(w_{\mathbf{c}_1}, w_{\mathbf{c}_2}, w_{\mathbf{c}_3})$ and $\Delta_2 = \Delta(w_{\mathbf{c}_1}, w_{\mathbf{c}_3}, w_{\mathbf{c}_4})$ lie inside the envelope of (v, v') if and only if the planes through each triangle strictly separate v from v' and the plane through v and diagonal $(w_{\mathbf{c}_1}, w_{\mathbf{c}_3})$ strictly separates $w_{\mathbf{c}_2}$ from $w_{\mathbf{c}_4}$. Figure 3.8 illustrates violations of those two conditions.

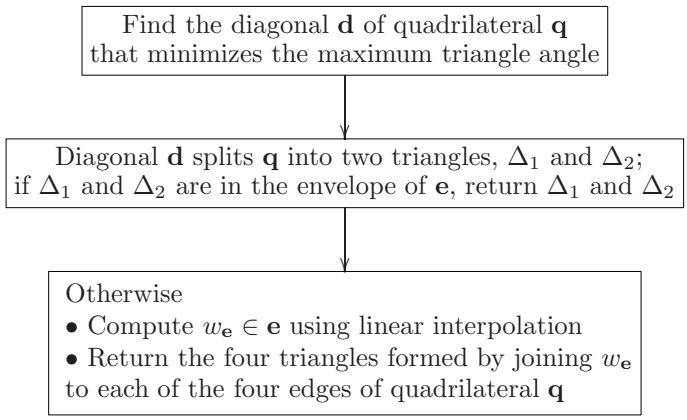


Figure 3.9. Triangulation of isosurface quadrilateral **q** around bipolar edge **e**.

Joining v and v' to triangles Δ_1 and Δ_2 gives four oriented tetrahedra: $\mathbf{t}_1 = (w_{\mathbf{c}_1}, w_{\mathbf{c}_2}, w_{\mathbf{c}_3}, v)$, $\mathbf{t}_2 = (w_{\mathbf{c}_1}, w_{\mathbf{c}_2}, w_{\mathbf{c}_3}, v')$, $\mathbf{t}_3 = (w_{\mathbf{c}_1}, w_{\mathbf{c}_3}, w_{\mathbf{c}_4}, v)$, and $\mathbf{t}_4 = (w_{\mathbf{c}_1}, w_{\mathbf{c}_3}, w_{\mathbf{c}_4}, v')$. The plane through Δ_1 strictly separates v from v' if and only if the orientations of $\mathbf{t}_1$ and $\mathbf{t}_2$ are nonzero and not equal. The plane through Δ_2 strictly separates v from v' if and only if the orientations of $\mathbf{t}_3$ and $\mathbf{t}_4$ are nonzero and not equal. The plane through v and diagonal $(w_{\mathbf{c}_1}, w_{\mathbf{c}_3})$ strictly separates $w_{\mathbf{c}_2}$ from $w_{\mathbf{c}_4}$ if and only if $\mathbf{t}_1$ and $\mathbf{t}_3$ have the same orientation. Thus, we can determine if the triangles Δ_1 and Δ_2 lie inside the envelope of (v, v') by computing and comparing the orientations of tetrahedra $\mathbf{t}_1$, $\mathbf{t}_2$, $\mathbf{t}_3$, and $\mathbf{t}_4$. (See Appendix B.6 on computing the orientation of four points.)

Each quadrilateral has two possible choices for a diagonal. We use the diagonal that minimizes the maximum triangle angle. Using this diagonal reduces the number of long, skinny triangles.

We select diagonal **d**, which minimizes the maximum triangle angle. Diagonal **d** splits quadrilateral **q** into two triangles, Δ_1 and Δ_2. If Δ_1 and Δ_2 are in the envelope of bipolar edge **e**, we replace quadrilateral **q** with triangles Δ_1 and Δ_2. If they are not, then we calculate the interpolated point $w_{\mathbf{e}}$ and form the four triangles with vertex $w_{\mathbf{e}}$ as described above. (See Figure 3.9.)

Forming a patch of triangles is one way of determining the surface bounded by a quadrilateral. Bilinear interpolation, Hermite surfaces, Bézier patches, and B-Splines can also be used to create smoother surfaces. As long as those surfaces are contained within the envelope of the corresponding bipolar edge, the resulting isosurface will not self-intersect.

3.2.4 Isosurface Properties

The SURFACE NETS algorithm produces an isosurface whose vertices may and usually do lie in the interior of the grid cubes. Thus the boundary of that isosurface is not usually on the boundary of the grid. A path connecting positive and negative grid vertices can bypass the isosurface in boundary grid cubes. However, if we restrict the path to interior grid cubes, then the path must intersect the isosurface. Restricted to interior grid cubes, the isosurface separates the positive and negative grid vertices.

If none of the grid vertices have scalar value equal to the isovalue, then the isosurface is minimal in the sense that it intersects interior bipolar grid edges exactly once and does not intersect positive or negative grid edges. The isosurface is not necessarily a manifold, even if none of the grid vertices have scalar value equal to the isovalue.

There are numerous variations on SURFACE NETS depending on the placement of the vertices within each grid cube and the construction of the surface bounded by each quadrilateral. The properties listed below apply to SURFACE NETS as described in the previous section, where vertices are placed at the centroid of edge-isosurface intersection points and the quadrilaterals are triangulated into two or four triangles based on the envelope.

SURFACE NETS returns a finite set, Υ, of triangles. The isosurface is the union of these triangles. The vertices of the isosurface are the triangle vertices. As defined in Section 3.1, Γ_{inner} is the subgrid of interior grid cubes of Γ.

The following properties apply to all isosurfaces produced by the SURFACE NETS algorithm.

Property 1. *The isosurface is piecewise linear.*

Property 2. *Set Υ contains at most $4m$ triangles where m is the number of bipolar grid edges.*

Property 3. *The isosurface does not contain any grid vertices whose scalar values do not equal the isovalue.*

Property 4. *The isosurface intersects at exactly one point every interior bipolar grid edge whose positive endpoint is strictly positive.*

Property 5. *The isosurface does not intersect any negative or strictly positive grid edges.*

Property 6. *The isosurface strictly separates the strictly positive vertices of Γ_{inner} from the negative vertices of Γ_{inner}.*

Properties 3–5 imply that the isosurface intersects a minimum number of grid edges.

The following properties apply to the SURFACE NETS isosurfaces whose isovalues do not equal the scalar value of any grid vertex.

Property 7. *Set Υ does not contain any zero-area triangles or duplicate triangles and the triangles in Υ form a triangulation of the isosurface.*

3.2.5 Proof of Isosurface Properties 1–5

Property 1. *The isosurface is piecewise linear.*

Proof: The isosurface is a finite union of triangles and thus is piecewise linear.□

Property 2. *Set Υ contains at most $4m$ triangles where m is the number of bipolar grid edges.*

Proof: Surface Nets constructs a quadrilateral for each bipolar edge $\mathbf{e}$ contained in four grid cubes and splits that quadrilateral into at most four triangles. Since the number of such grid edges is at most m, the number of triangles is at most $4m$. □

Property 3. *The isosurface does not contain any grid vertices whose scalar values do not equal the isovalue.*

Proof: Let v be a grid vertex whose scalar value does not equal the isovalue. If v were to lie on the isosurface, then it would have to be contained in some isosurface triangle formed from some bipolar edge $\mathbf{e}$. Edge $\mathbf{e}$ would have to be in a grid cube incident on v. We consider two cases depending upon whether $\mathbf{e}$ itself is incident on v.

Consider a bipolar edge $\mathbf{e}$ incident on v. Let P be the plane through v orthogonal to $\mathbf{e}$. Surface Nets uses linear interpolation to determine a point $w_{\mathbf{e}} \in \mathbf{e}$ that approximates the intersection of $\mathbf{e}$ and the isosurface. Since the scalar value of v does not equal the isovalue, this point is not v. For every grid cube $\mathbf{c}$ containing $\mathbf{e}$, the point $w_{\mathbf{c}}$ is the centroid of a set of points in $\mathbf{c}$ that includes point $w_{\mathbf{e}}$. Thus, $w_{\mathbf{e}}$ pulls $w_{\mathbf{c}}$ away from the plane P and none of the triangles formed from the quadrilateral around $\mathbf{e}$ contain v.

Now consider a bipolar edge $\mathbf{e}$ that is not incident on v. Let $\mathbf{c}$ be a grid cube containing $\mathbf{e}$ and v and let $\mathbf{f}$ be a facet of $\mathbf{c}$ that contains v but does not intersect $\mathbf{e}$. Let P be the plane containing $\mathbf{f}$. Let $w_{\mathbf{e}} \in \mathbf{e}$ be the linear interpolation point on $\mathbf{e}$. For every grid cube $\mathbf{c}'$ containing $\mathbf{e}$ and v, point $w_{\mathbf{e}}$ pulls $w_{\mathbf{c}'}$ away from plane P. Thus, none of the triangles formed from the quadrilateral around $\mathbf{e}$ contain v. □

As defined in Section 3.1, an interior grid edge $\mathbf{e}$ is a grid edge contained in four grid cubes. To prove Property 4, we first show that every interior bipolar grid edge intersects the isosurface.

Lemma 3.7. *If $\mathbf{e}$ is an interior bipolar grid edge, then $\mathbf{e}$ intersects the isosurface.*

Proof: Let $\mathbf{q}$ be the quadrilateral around $\mathbf{e}$. Let $\Sigma_{\mathbf{e}}$ be the isosurface patch generated by splitting $\mathbf{q}$ into triangles. By construction, $\Sigma_{\mathbf{e}}$ is contained in the envelope of $\mathbf{e}$ and its boundary is $\mathbf{q}$. Since $\mathbf{e}$ passes through $\mathbf{q}$ and has endpoints on the envelope of $\mathbf{e}$, edge $\mathbf{e}$ intersects $\Sigma_{\mathbf{e}}$. □

As defined in Section 3.1, a grid cube or grid face is active if it contains at least one negative and at least one positive vertex. For each active grid cube $\mathbf{c}$, algorithm SURFACE NETS constructs a point $w_{\mathbf{c}} \in \mathbf{c}$ by approximating the intersections of the isosurface and the cube edges and taking the centroid of these intersections.

Lemma 3.8. *If v is a strictly positive vertex of an active cube $\mathbf{c}$ and $\mathbf{f}$ is a facet of $\mathbf{c}$ containing v, then $w_{\mathbf{c}}$ does not lie on $\mathbf{f}$.*

Proof: Let $\mathbf{f}'$ be the facet of $\mathbf{c}$ parallel to $\mathbf{f}$. We consider three cases based on whether $\mathbf{f}'$ is active, negative, or positive.

Case I: Facet $\mathbf{f}'$ is active.

Since $\mathbf{f}'$ is active, it has a bipolar edge $\mathbf{e}'$. Some point $r' \in \mathbf{e}'$ approximates the intersection of the isosurface and $\mathbf{e}'$. Point $w_{\mathbf{c}}$ is the centroid of a set of points in $\mathbf{c}$ that includes point r'. Since r' is not on $\mathbf{f}$, point r' pulls $w_{\mathbf{c}}$ away from $\mathbf{f}$ and $w_{\mathbf{c}}$ does not lie on $\mathbf{f}$.

Case II: Facet $\mathbf{f}'$ is negative.

Let $\mathbf{e} = (v, v')$ be the edge from v to a vertex v' in $\mathbf{f}'$. Since $\mathbf{f}'$ is negative, vertex v' is negative and edge $\mathbf{e}$ is bipolar. Some point $r \in \mathbf{e}$ approximates the intersection of the isosurface and $\mathbf{e}$. Since v is strictly positive, point r does not equal v and does not lie on $\mathbf{f}$. Thus r pulls $w_{\mathbf{c}}$ away from $\mathbf{f}$ and $w_{\mathbf{c}}$ does not lie on $\mathbf{f}$.

Case III: Facet $\mathbf{f}'$ is positive.

Since cube $\mathbf{c}$ is active, cube $\mathbf{c}$ must contain some negative vertex v''. Since $\mathbf{f}'$ is positive, negative vertex v'' must lie on $\mathbf{f}$. Let $\mathbf{e} = (v'', v')$ be the edge from v'' to a vertex v' in $\mathbf{f}'$. Since $\mathbf{f}'$ is positive, edge $\mathbf{e}$ is bipolar. Some point $r \in \mathbf{e}$ approximates the intersection of the isosurface and $\mathbf{e}$. Since v'' is negative, point r' does not equal v and does not lie on $\mathbf{f}$. Thus r' pulls $w_{\mathbf{c}}$ away from $\mathbf{f}$ and $w_{\mathbf{c}}$ does not lie on $\mathbf{f}$. □

Property 4. *The isosurface intersects at exactly one point every interior bipolar grid edge whose positive endpoint is strictly positive.*

Proof: Let $\mathbf{e}$ be an interior edge with one endpoint scalar value greater than the isovalue and one endpoint scalar value less than the isovalue. Since neither scalar value equals the isovalue, the isosurface does not contain either endpoint (Property 3).

Let $\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{c}_3$, and $\mathbf{c}_4$ be the four grid cubes containing $\mathbf{e}$. By Lemma 3.7, the isosurface intersects $\mathbf{e}$. By Lemma 3.8, none of the $w_{\mathbf{c}_i}$ lie on facets containing $\mathbf{e}$. If the quadrilateral around $\mathbf{e}$ is split into four triangles, then each of these triangles intersect $\mathbf{e}$ at point $w_{\mathbf{e}}$, proving the property. If the quadrilateral around $\mathbf{e}$ is split into two triangles by a diagonal $\mathbf{d}$, then either both triangles intersect $\mathbf{e}$ at $\mathbf{d} \cap \mathbf{e}$ or one triangle intersects $\mathbf{e}$ at a single point and the other does not intersect $\mathbf{e}$. □

Property 5. *The isosurface does not intersect any negative or strictly positive grid edges.*

Proof: Let $\mathbf{e}$ be a grid edge whose scalar values are both greater or both less than the isovalue. Since the scalar values at the endpoints of $\mathbf{e}$ do not equal the isovalue, the isosurface does not contain any endpoint (Property 3).

Since $\mathbf{e}$ is not a bipolar edge, edge $\mathbf{e}$ does not generate any isosurface triangles. Let $\mathbf{e}'$ be a bipolar grid edge. By construction, all the triangles generated by $\mathbf{e}'$ are contained in the envelope of $\mathbf{e}'$. If $\mathbf{e}'$ does not intersect $\mathbf{e}$, the envelope of $\mathbf{e}'$ does not intersect $\mathbf{e}$ and none of the isosurface triangles generated by $\mathbf{e}'$ intersect $\mathbf{e}$. If $\mathbf{e}'$ shares an endpoint with $\mathbf{e}$, then the envelope of $\mathbf{e}'$ intersects $\mathbf{e}$ only at this endpoint. Since the isosurface does not contain this endpoint, none of the triangles generated by $\mathbf{e}'$ intersect $\mathbf{e}$. Thus $\mathbf{e}$ does not intersect the isosurface.□

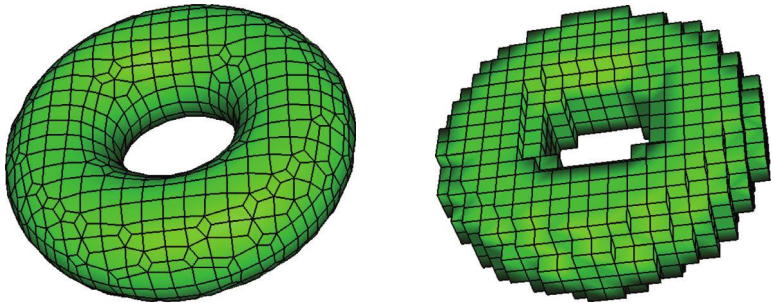
3.2.6 Proof of Isosurface Property 6

To prove Property 6, we first show that a simplified version of the SURFACE NETS isosurface separates positive from negative vertices. We then apply a homotopy map from the simplified version to the SURFACE NETS isosurface, which preserves the separation.

For each cube $\mathbf{c}$, let $a_{\mathbf{c}}$ be the center of $\mathbf{c}$. For each edge $\mathbf{e}$, let $a_{\mathbf{e}}$ be the midpoint of $\mathbf{e}$. Remap the set of triangles Υ by mapping each isosurface vertex $w_{\mathbf{c}}$ to the cube center $a_{\mathbf{c}}$ and each isosurface vertex $w_{\mathbf{e}}$ to the edge midpoint $a_{\mathbf{e}}$. Let $\widehat{\Upsilon}$ be the resulting set of triangles. Each interior bipolar edge corresponds to two or four triangles in $\widehat{\Upsilon}$ that together form a square that is orthogonal to the edge and passes through the edge midpoint. Set $|\widehat{\Upsilon}|$ is the union of all the triangles in $\widehat{\Upsilon}$. (See the example in Figure 3.10.)

Let a_0 be the center of the grid cube with the lowest x, y, and z coordinates and let a_1 be the center of the grid cube with the highest x, y, and z coordinates. If grid Γ is composed of unit cubes and has origin $(0, 0, 0)$ and vertex dimensions $n_x \times n_y \times n_z$, then a_0 is the point $(0.5, 0.5, 0.5)$ and a_1 is the point $(n_x - 0.5, n_y - 0.5, n_z - 0.5)$. Let $\mathbb{X}^*_\Gamma$ be the rectangular region from a_0 to a_1. (See Figure 3.11 for a 2D illustration.) The isosurface $|\widehat{\Upsilon}|$ is contained in $\mathbb{X}^*_\Gamma$. The boundary of this isosurface is contained in $\partial\mathbb{X}^*_\Gamma$.

We will show that a path that lies in $\mathbb{X}^*_\Gamma$ from a positive vertex of Γ_{inner} to a negative vertex of Γ_{inner} must intersect $|\widehat{\Upsilon}|$ (Lemma 3.13). We first consider a



(a) $|\Upsilon|$. (b) $|\widehat{\Upsilon}|$.

Figure 3.10. Dual contouring isosurfaces of torus. Only quadrilateral edges are displayed. (a) Isosurface $|\Upsilon|$ produced by SURFACE NETS. (b) Isosurface $|\widehat{\Upsilon}|$ produced by positioning vertices at cube centers and edge midpoints.

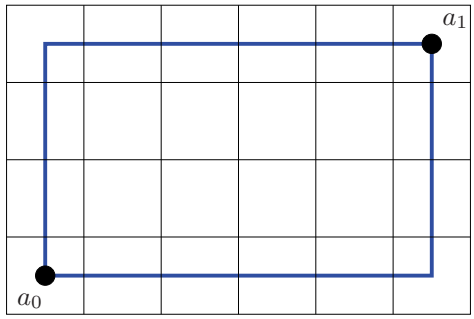
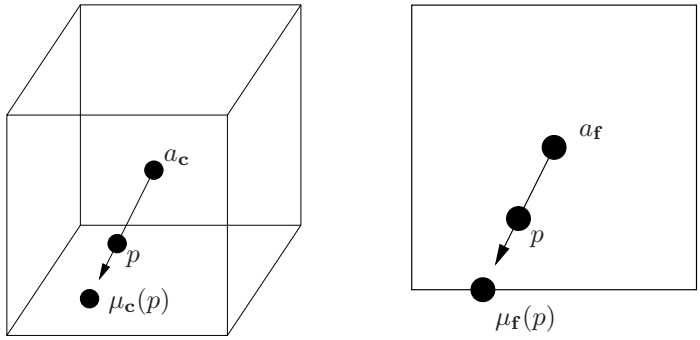


Figure 3.11. Two-dimensional illustration of rectangular region $\mathbb{X}^*_\Gamma$ (blue). Point a_0 is the center of the grid cube with the lowest x and y coordinates. Point a_1 is the center of the grid cube with the largest x and y coordinates.

path ζ in a single grid cube and show how it can be mapped to the edges and faces of the cube (Lemmas 3.9 and 3.10 and Corollary 3.11). We show that a path ζ from a negative vertex to a positive vertex of a grid cube will intersect $|\widehat{\Upsilon}|$ (Corollary 3.12). Finally, we divide grid cubes into negative and positive regions separated by $|\widehat{\Upsilon}|$ and prove that these regions agree on the faces between grid cubes. Any path in $\mathbb{X}^*_\Gamma$ from a negative vertex to a positive vertex of Γ_{inner} would have have to cross from a negative region to a positive one and would have to intersect $|\widehat{\Upsilon}|$ (Lemma 3.13).

We start by showing how paths in grid cube $\mathbf{c}$ can be mapped to faces and edges of $\mathbf{c}$. Define a mapping $\mu_{\mathbf{c}} : (\mathbf{c} - a_{\mathbf{c}}) \to \partial\mathbf{c}$ from $(\mathbf{c} - a_{\mathbf{c}})$ to the boundary of $\mathbf{c}$ as follows. For each point $x \in \mathbf{c} - a_{\mathbf{c}}$, there is a ray $\mathbf{r}_x$ from $a_{\mathbf{c}}$ through x that intersects $\partial\mathbf{c}$ at exactly one point. Let $\mu_{\mathbf{c}}(x)$ equal $\mathbf{r}_x \cap \partial\mathbf{c}$. Mapping $\mu_{\mathbf{c}}$ is called a radial projection of $(\mathbf{c} - a_{\mathbf{c}})$. (See Figure 3.12(a).)



(a) $\mu_{\mathbf{c}}(p)$. (b) $\mu_{\mathbf{f}}(p)$.

Figure 3.12. Radial projections. (a) Radial projection of $p \in \mathbf{c} - \{a_{\mathbf{c}}\}$ to $\mu_{\mathbf{c}}(p) \in \partial\mathbf{c}$. Point $a_{\mathbf{c}}$ is the center of cube $\mathbf{c}$. (b) Radial projection of $p \in \mathbf{f} - \{a_{\mathbf{f}}\}$ to $\mu_{\mathbf{f}}(p) \in \partial\mathbf{f}$. Point $a_{\mathbf{f}}$ is the center of face $\mathbf{f}$.

We define a similar mapping $\mu_{\mathbf{f}}$ for each grid facet $\mathbf{f}$. For each grid facet $\mathbf{f}$, let $a_{\mathbf{f}}$ be the center of $\mathbf{f}$. For each point $x \in \mathbf{f} - a_{\mathbf{f}}$, there is a ray $\mathbf{r}_x$ from $a_{\mathbf{f}}$ through x that intersects $\partial\mathbf{f}$ at exactly one point. Let $\mu_{\mathbf{f}}(x)$ equal $\mathbf{r}_x \cap \partial\mathbf{f}$. Mapping $\mu_{\mathbf{f}}$ is a radial projection of $(\mathbf{f} - a_{\mathbf{f}})$. (See Figure 3.12(b).)

Lemma 3.9.

1. *For all $x \in \mathbf{c} - a_{\mathbf{c}}$, $x \in |\widehat{\Upsilon}|$ if and only if $\mu_{\mathbf{c}}(x_{\mathbf{c}}) \in |\widehat{\Upsilon}|$.*
2. *For all $x \in \mathbf{f} - a_{\mathbf{f}}$, $x \in |\widehat{\Upsilon}|$ if and only if $\mu_{\mathbf{f}}(x_{\mathbf{f}}) \in |\widehat{\Upsilon}|$.*

Proof of 1: If x is in $|\widehat{\Upsilon}|$, then x lies in some square in $|\widehat{\Upsilon}|$ with vertex $a_{\mathbf{c}}$. Point $\mu_{\mathbf{c}}(x)$ also lies in this square so $\mu_{\mathbf{c}}(x)$ is in $|\widehat{\Upsilon}|$.

On the other hand, if $\mu_{\mathbf{c}}(x)$ is in $|\widehat{\Upsilon}|$, then $\mu_{\mathbf{c}}(x)$ lies in some square in $|\widehat{\Upsilon}|$ with vertex $a_{\mathbf{c}}$. Line segment $(a_{\mathbf{c}})$ is contained in this square and x lies on this lines segment, so x is in $|\widehat{\Upsilon}|$. □

The proof of Statement 2 in Lemma 3.9 is similar to the proof of Statement 1 and is omitted.

Let p be a point in $\mathbf{c} - a_{\mathbf{c}}$. Note that p is in $\mathbb{X}^*_\Gamma$ if and only if $\mu_{\mathbf{c}}(p)$ is in $\mathbb{X}^*_\Gamma$. Similarly, if p is a point in $\mathbf{f} - a_{\mathbf{f}}$, then p is in $\mathbb{X}^*_\Gamma$ if and only if $\mu_{\mathbf{f}}(p)$ is in $\mathbb{X}^*_\Gamma$.

The 1-skeleton of a cube is the union of all the vertices and edges of the cube.

Lemma 3.10. *Let $\mathbf{c}$ be a grid cube of Γ, let $\mathbf{e}$ be an edge of $\mathbf{c}$, and let ζ be a path in $\mathbf{c} \cap \mathbb{X}^*_\Gamma$ from a negative vertex of $\mathbf{c}$ to some point $p \in \mathbf{e} \cap \mathbb{X}^*_\Gamma$. If ζ does not intersect $|\widehat{\Upsilon}|$, then there is a negative endpoint v of $\mathbf{e}$ such that v is in $\mathbb{X}^*_\Gamma$ and the closed line segment $[v, p]$ does not intersect $|\widehat{\Upsilon}|$.*

Proof: If all the vertices in $\mathbf{c}$ are negative, then $|\widehat{\Upsilon}|$ does not intersect $\mathbf{c}$. The closed line segment from a vertex of $\mathbf{e}$ in $\mathbb{X}^*_\Gamma$ to p does not intersect $|\widehat{\Upsilon}|$, satisfying the conclusion.

Assume some vertex of $\mathbf{c}$ is positive. Since $\mathbf{c}$ contains at least one negative vertex, $|\widehat{\Upsilon}|$ intersects $\mathbf{c}$ and contains $a_\mathbf{c}$, the center of $\mathbf{c}$.

Let ζ be a path in $\mathbf{c} \cap \mathbb{X}^*_\Gamma$ that does not intersect $|\widehat{\Upsilon}|$ from a negative vertex v of $\mathbf{c}$ to point $p \in \mathbf{e} \cap \mathbb{X}^*_\Gamma$. By Lemma 3.9, there is a continuous map $\mu_\mathbf{c}$ that takes $\mathbf{c} - a_\mathbf{c}$ to $\partial\mathbf{c}$, maps points in $|\widehat{\Upsilon}|$ to points on $|\widehat{\Upsilon}|$ and points in $\mathbf{c} - |\widehat{\Upsilon}|$ to points in $\partial\mathbf{c} - |\widehat{\Upsilon}|$. Since path ζ does not intersect the $|\widehat{\Upsilon}|$, path ζ does not contain $a_\mathbf{c}$. Apply the mapping $\mu_\mathbf{c}$ to ζ to get a path ζ' lying in $\partial\mathbf{c}$ that connects v and p. Since ζ does not intersect $|\widehat{\Upsilon}|$, neither does ζ'. Since ζ is in $\mathbb{X}^*_\Gamma$, path ζ' is in $\mathbb{X}^*_\Gamma$.

Path ζ' may pass through the center $a_\mathbf{g}$ of some facets $\mathbf{g}$ of $\mathbf{c}$. If it does, then we can perturb it slightly to form a path $\zeta'' \subset \partial\mathbf{c}$ that does not pass through the center $a_\mathbf{g}$ of any facet $\mathbf{g}$ of $\mathbf{c}$ and remains in $\mathbb{X}^*_\Gamma$. If the perturbation is small enough, then ζ'' will not intersect $|\widehat{\Upsilon}|$.

By Lemma 3.9, there is a continuous map $\mu_\mathbf{g}$ that takes $\mathbf{g} - a_\mathbf{g}$ to $\partial\mathbf{g}$, maps points in $|\widehat{\Upsilon}|$ to points on $|\widehat{\Upsilon}|$, and points in $\mathbf{f} - |\widehat{\Upsilon}|$ to points in $\partial\mathbf{f} - |\widehat{\Upsilon}|$. Since ζ'' does not contain $a_\mathbf{c}$, applying the mapping $\mu_\mathbf{g}$ to ζ'' for each facet $\mathbf{g}$ gives a path ζ''' that intersects neither the interior of $\mathbf{c}$ nor that of any facet of $\mathbf{c}$. Thus, path ζ''' is contained in the 1-skeleton of the cube. Since ζ'' is in $\mathbb{X}^*_\Gamma$, path ζ''' is also in $\mathbb{X}^*_\Gamma$. Path ζ''' connects a negative vertex of $\mathbf{c}$ to p.

Every bipolar edge of $\mathbf{c}$ intersects $|\widehat{\Upsilon}|$ at its midpoint. Since path ζ''' does not intersect $|\widehat{\Upsilon}|$, it does not contain any bipolar edge of $\mathbf{c}$. To reach p, path ζ''' must pass through a sequence of grid vertices $(v_1, v_2, \ldots, v_k)$ of $\mathbf{c}$. Since ζ''' does not contain any bipolar edge of $\mathbf{c}$ and v_1 is negative, each of the vertices v_i must be negative. In particular, vertex v_k must be negative. Since v_k is the last grid vertex contained in ζ''', it must be an endpoint of edge $\mathbf{e}$. Since ζ''' is in $\mathbb{X}^*_\Gamma$, vertex v_k is in $\mathbb{X}^*_\Gamma$.

Path ζ''' contains a subpath $\zeta_\mathbf{e} \subset \mathbf{e}$ that connects v_k to p. Since $\zeta_\mathbf{e}$ does not intersect $|\widehat{\Upsilon}|$, the closed line segment $[v_k, p]$ does not intersect $|\widehat{\Upsilon}|$, proving the lemma. □

Corollary 3.11. *Let $\mathbf{c}$ be a grid cube of grid Γ, let $\mathbf{f}$ be a face of $\mathbf{c}$, and let ζ be a path in $\mathbf{c} \cap \mathbb{X}^*_\Gamma$ from a negative vertex of $\mathbf{c}$ to some point $p \in \mathbf{f} \cap \mathbb{X}^*_\Gamma$. If ζ does not intersect $|\widehat{\Upsilon}|$, then there is a path in $\mathbf{f} \cap \mathbb{X}^*_\Gamma$ that does not intersect $|\widehat{\Upsilon}|$ from a negative vertex of $\mathbf{f}$ to p.*

Proof: We consider two cases depending upon whether p is at the center of $\mathbf{f}$.

Case I: Point p is not at the center of $\mathbf{f}$.

Since p is not at the center of $\mathbf{f}$, the map μ can be applied to p. Point $\mu_\mathbf{f}(p)$ is on an edge $\mathbf{e}$ of $\mathbf{f}$. Since p is in $\mathbb{X}^*_\Gamma$, so is $\mu_\mathbf{f}(p)$. By Lemma 3.10, edge $\mathbf{e}$ has a negative endpoint v such that v is in $\mathbb{X}^*_\Gamma$ and the closed line

segment $[v, \mu_{\mathbf{f}}(p)]$ does not intersect $|\widehat{\Upsilon}|$. Since p is not in $|\widehat{\Upsilon}|$, line segment $[\mu_{\mathbf{f}}(p), p]$ does not intersect $|\widehat{\Upsilon}|$. Since p and $\mu_{\mathbf{f}}(p)$ are in $\mathbb{X}^*_\Gamma$, line segment $[\mu_{\mathbf{f}}(p), p]$ is in $\mathbb{X}^*_\Gamma$. Thus the path $(v, \mu_{\mathbf{f}}(p), p)$ is in $\mathbf{f} \cap \mathbb{X}^*_\Gamma$ and does not intersect $|\widehat{\Upsilon}|$.

Case II: Point p is at the center of $\mathbf{f}$.

Perturb p slightly to a point p' such that closed line segment $[p, p']$ does not intersect $|\widehat{\Upsilon}|$ and p' is in $\mathbb{X}^*_\Gamma$. By Case I, there is a path $\zeta_{\mathbf{f}}$ in $\mathbf{f} \cap \mathbb{X}^*_\Gamma$ from some negative vertex v of $\mathbf{f}$ to p'. Adding $[p', p]$ to this path gives a path ζ in $\mathbf{f} \cap \mathbb{X}^*_\Gamma$ from v to p that does not intersect $|\widehat{\Upsilon}|$. □

Corollary 3.12. *Let $\mathbf{c}$ be a grid cube of Γ. If ζ is a path in $\mathbf{c} \cap \mathbb{X}^*_\Gamma$ from a negative vertex to a positive vertex of $\mathbf{c}$, then ζ intersects $|\widehat{\Upsilon}|$.*

Proof: Assume ζ was a path in $\mathbf{c} \cap \mathbb{X}^*_\Gamma$ from a negative vertex of $\mathbf{c}$ to a positive vertex v of $\mathbf{c}$ that did not intersect $|\widehat{\Upsilon}|$. Let $\mathbf{e}$ be an edge of $\mathbf{c}$ containing v. By Lemma 3.10, edge $\mathbf{e}$ has a negative endpoint v' such that v' is in $\mathbb{X}^*_\Gamma$ and the closed line segment $[v, v']$ does not intersect $|\widehat{\Upsilon}|$. Since v is positive and v' is negative, vertices v and v' must be distinct and (v, v') must equal $\mathbf{e}$. Since $\mathbf{e}$ has a positive endpoint v and a negative endpoint v', edge $\mathbf{e} = (v, v')$ intersects $|\widehat{\Upsilon}|$ at the midpoint of $\mathbf{e}$, a contradiction. Thus ζ intersects $|\widehat{\Upsilon}|$. □

Note that Corollary 3.12 requires that ζ be contained in $\mathbb{X}^*_\Gamma$. If ζ is not in $\mathbb{X}^*_\Gamma$, then there may be a path that does not intersect $|\widehat{\Upsilon}|$ from a negative vertex to a positive vertex in $\mathbf{c}$.

Lemma 3.13. *Every path ζ in $\mathbb{X}^*_\Gamma$ from a positive vertex of Γ_{inner} to a negative vertex of Γ_{inner} intersects $|\widehat{\Upsilon}|$.*

Proof: For each grid cube $\mathbf{c}$ of grid Γ, let the negative region for $\mathbf{c} \cap \mathbb{X}^*_\Gamma$ be the set of points that can be reached by a path in $\mathbb{X}^*_\Gamma$ from a negative vertex such that the path does not intersect $|\widehat{\Upsilon}|$. Let the positive region for $\mathbf{c} \cap \mathbb{X}^*_\Gamma$ be all the other points in $\mathbf{c} \cap \mathbb{X}^*_\Gamma$. The positive region contains $|\widehat{\Upsilon}| \cap \mathbf{c}$ and is closed. Any point within the positive region that does not lie on $|\widehat{\Upsilon}|$ has a neighborhood in $\mathbf{c} \cap \mathbb{X}^*_\Gamma$ contained in the positive region.

By construction of $\widehat{\Upsilon}$, no grid vertex is contained in $|\widehat{\Upsilon}|$. Since a zero-length path connects a negative vertex to itself, every negative vertex of $\mathbf{c}$ in $\mathbf{c} \cap \mathbb{X}^*_\Gamma$ is contained in the negative region. On the other hand, by Corollary 3.12 any path in $\mathbf{c} \cap \mathbb{X}^*_\Gamma$ from a negative vertex of $\mathbf{c}$ to a positive vertex $\mathbf{c}$ must intersect $|\widehat{\Upsilon}|$.

We claim that positive and negative regions agree on shared boundaries of grid cubes. Let R_1^- and R_2^- be the negative regions of $\mathbf{c}_1 \cap \mathbb{X}^*_\Gamma$ and $\mathbf{c}_2 \cap \mathbb{X}^*_\Gamma$ for two adjacent grid cubes, $\mathbf{c}_1$ and $\mathbf{c}_2$. Let $\mathbf{f}$ be the intersection of the two grid cubes. Note that $\mathbf{f}$ may be a grid vertex, a grid edge, or a square grid facet.

By Corollary 3.11, if point p lies in $R_1^- \cap \mathbf{f}$, then some path in $\mathbf{f} \cap \mathbb{X}_\Gamma^*$ does not intersect $|\widehat{\Upsilon}|$ and connects a negative vertex to p. Since $\mathbf{f}$ is a face of $\mathbf{c}_2$, this path also lies in $\mathbf{c}_2$ and so p is in R_2^-. A symmetric argument shows that if p is in R_2^-, then p is in R_1^-. Thus the positive and negative regions agree on shared boundaries of grid cubes.

Let R^+ be the union of the positive regions of $\mathbf{c} \cap \mathbb{X}_\Gamma^*$ over all the grid cubes $\mathbf{c} \in \Gamma$. Consider a path ζ in $\mathbb{X}_\Gamma^*$ from a positive grid vertex to a negative one. The positive grid vertex lies in R^+ while the negative one does not. Thus ζ must intersect some point p on the boundary of R^+ where it crosses out of R^+. Every neighborhood of p must contain points that are not in R^+.

Since R^+ is closed, point p lies in R^+. Since positive regions agree on the grid cube boundaries, point p lies in the positive region of each grid cube containing p. Assume p is not in $|\widehat{\Upsilon}|$. Within each grid cube containing p, some neighborhood of p is contained in the positive region for that grid cube. The union of those neighborhoods is a neighborhood of p within the grid and is contained in R^+. Thus ζ does not cross out of R^+ at p. We conclude that p must lie on $|\widehat{\Upsilon}|$ and that ζ intersects $|\widehat{\Upsilon}|$. □

Set $|\Upsilon|$ is the union of all the triangles in Υ. Therefore, set $|\Upsilon|$ is the set of points on the isosurface created by Surface Nets.

Property 6. *The isosurface strictly separates the strictly positive vertices of* Γ_{inner} *from the negative vertices of* Γ_{inner}.

Proof: Each vertex of a triangle in $\widehat{\Upsilon}$ is the midpoint $a_{\mathbf{e}}$ of some bipolar edge $\mathbf{e}$ or the center $a_{\mathbf{c}}$ of some active cube $\mathbf{c}$. For bipolar edges $\mathbf{e}$ and active cubes $\mathbf{c}$, define the homotopy

$$\eta(a_{\mathbf{e}}, \alpha) = (1 - \alpha)a_{\mathbf{e}} + \alpha w_{\mathbf{e}},$$
$$\eta(a_{\mathbf{c}}, \alpha) = (1 - \alpha)a_{\mathbf{c}} + \alpha w_{\mathbf{c}}.$$

Linearly extend the homotopy η to all the triangles in $\widehat{\Upsilon}$ giving a homotopy $\eta : |\widehat{\Upsilon}| \to |\Upsilon|$.

By Lemma 3.13, every path in $\mathbb{X}_\Gamma^*$ from a positive vertex of Γ_{inner} to a negative vertex of Γ_{inner} intersects set $|\widehat{\Upsilon}|$. Set $|\widehat{\Upsilon}|$ does not contain any vertices of Γ_{inner}. Map η is a homotopy map that never passes through any negative or strictly positive vertices in Γ_{inner}. Since $w_{\mathbf{c}}$ is contained in cube $\mathbf{c}$, set $\{\eta(x, \alpha) : x \in |\widehat{\Upsilon}| \cap \partial\mathbb{X}_\Gamma^*\}$ is a subset of $\mathbb{R}^3 - |\Gamma_{\text{inner}}|$. By Lemma B.19 in Appendix B, every path in $|\Gamma_{\text{inner}}|$ from a negative to a strictly positive vertex of Γ_{inner} intersects $|\Upsilon|$. Thus $|\Upsilon|$ separates the negative vertices of Γ_{inner} from the strictly positive vertices of Γ_{inner}. By Property 3, isosurface $|\Upsilon|$ does not contain any negative or strictly positive vertices of Γ_{inner}. Thus $|\Upsilon|$ strictly separates the negative vertices of Γ_{inner} from the strictly positive vertices of Γ_{inner}. □

3.2.7 Proof of Isosurface Property 7

Property 7. *Set Υ does not contain any zero-area triangles or duplicate triangles and the triangles in Υ form a triangulation of the isosurface.*

Proof: Let $\mathbf{q}$ be a quadrilateral around edge $\mathbf{e}$. Since the isovalue does not equal the scalar value of any grid vertex, the points $w_{\mathbf{c}}$ are in the interior of their respective cubes $\mathbf{c}$. Thus no two vertices of $\mathbf{e}$ coincide and no four vertices of $\mathbf{e}$ are colinear. Three vertices can be colinear, but the degenerate triangle containing the three vertices has a 180-degree angle and so the diagonal splitting this triangle will be selected. Thus, if $\mathbf{q}$ is split into two triangles, neither triangle will have zero area.

Let $(w_{\mathbf{c}_1}, w_{\mathbf{c}_2})$ be an edge of $\mathbf{q}$. Since no point $w_{\mathbf{c}}$ lies on a grid facet, points $w_{\mathbf{c}_1}$, $w_{\mathbf{c}_2}$, and $w_{\mathbf{e}}$ are not colinear. Thus triangle $\Delta(w_{\mathbf{c}_1}, w_{\mathbf{c}_2}, w_{\mathbf{e}})$ does not have zero area.

Each quadrilateral around an edge $\mathbf{e}$ is replaced by two or four triangles. By construction, this triangular patch is contained in the envelope of bipolar edge $\mathbf{e}$. Since the envelopes of two different bipolar edges intersect only at shared vertices, edges, or facets, the triangles in different triangular patches intersect only at shared vertices or edges. Thus Υ forms a triangulation of the isosurface. □

3.2.8 Non-Manifold Isosurface Patches

Let $\mathbf{c}$ be some active grid cube. Since all the isosurface quadrilaterals created by bipolar edges of $\mathbf{c}$ are incident on $w_{\mathbf{c}}$, there are no topological choices about the construction of the isosurface patch in $\mathbf{c}$. (There is a choice of the triangulation, but this choice affects the geometry, not the topology, of the isosurface.) Thus, the ambiguous Marching Cubes configurations listed in Section 2.3.5 do not create any ambiguity for the Surface Nets algorithm.

However, the ambiguous Marching Cubes configurations do create another problem for Surface Nets. Surface Nets applied to any of these configurations creates a non-manifold in the neighborhood of $w_{\mathbf{c}}$. The non-manifold around $w_{\mathbf{c}}$ may be viewed as the price to pay for avoiding any ambiguity.

The non-manifold configurations and their isosurface patches are illustrated in Figure 3.13. Note that even the isosurface patches for configurations 2C and 6C are not manifolds, even though none of the facets of these configurations are ambiguous. Note also that the isosurface quadrilaterals are artificially truncated. They actually extend further into adjacent cubes.

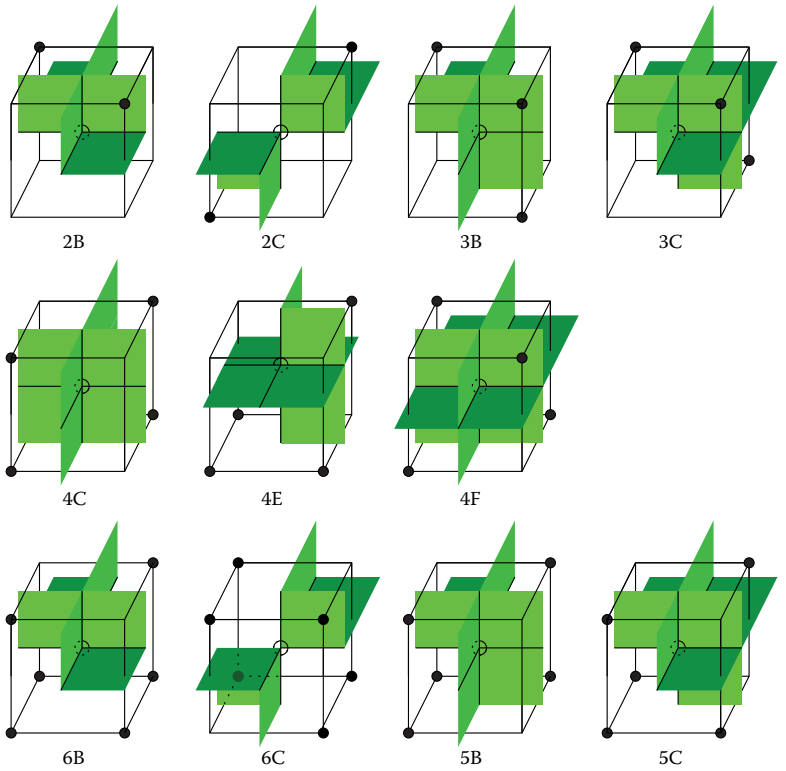


Figure 3.13. Non-manifold isosurface patches. Black vertices are positive.

3.2.9 Convex Polyhedral Meshes

Algorithm SURFACE NETS extends easily to convex polyhedral meshes. A mesh element is active if it contains at least one negative and at least one positive vertex. For each active mesh element $\mathbf{c}$, construct an isosurface vertex $w_{\mathbf{c}}$. For each bipolar mesh edge contained in mesh elements $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k$, add an isosurface polygon with vertices $w_{\mathbf{c}_1}, w_{\mathbf{c}_2}, \ldots, w_{\mathbf{c}_k}$. Note that this polygon is not necessarily a quadrilateral.

To position the isosurface vertices within each grid cube, use linear interpolation as described in Section 1.7.2 to approximate the intersection of the isosurface and all the bipolar edges. Take the centroid of all the approximation points as the location of the isosurface vertex.

The isosurface polygon $(w_{\mathbf{c}_1}, w_{\mathbf{c}_2}, \ldots, w_{\mathbf{c}_k})$ is not necessarily planar. To create a piecewise linear surface, triangulate the isosurface polygons.

3.3 Dual Marching Cubes

The Surface Nets algorithm places a single isosurface vertex in each active grid cube. This single vertex causes the surface to be a non-manifold for some configurations. The Dual Marching Cubes algorithm by Nielson [Nielson, 2004] attempts to remedy this problem by sometimes adding more than one vertex per cube.

For each bipolar grid edge, Dual Marching Cubes creates an isosurface quadrilateral connecting four isosurface vertices in the four incident grid cubes. If the grid cube contains more than one isosurface vertex, the appropriate vertex is chosen by using a lookup table. The isosurface quadrilaterals are then split into two triangles.

As will be discussed in Section 3.3.3, there is a correspondence between the Dual Marching Cubes lookup table and the Marching Cubes lookup table (Figure 2.16). The name Dual Marching Cubes comes from this correspondence between the two lookup tables.

For most configurations in Figure 3.13, Dual Marching Cubes creates two, three or four isosurface vertices in the grid cube. However, for two configurations, 2B and 3B, Dual Marching Cubes still adds only a single isosurface vertex. As discussed in Section 3.3.4, these configurations can create non-manifold surfaces.

A small modification of Dual Marching Cubes changes the isosurface patches for some instances of configurations 2B and 3B, avoiding non-manifold edges created by those configurations. The modified algorithm, called Manifold Dual Marching Cubes, produces an isosurface which is guaranteed to be a manifold if the isovalue does not equal the scalar value of any grid vertex. The modified algorithm is described in Section 3.3.5.

We first describe a two dimensional version of the algorithm, which we call Dual Marching Squares.

3.3.1 Dual Marching Squares

Input to the Dual Marching Squares algorithm is an isovalue and a set of scalar values at the vertices of a two-dimensional regular grid. The algorithm has three steps. (See Figure 3.14.) Read in the isocontour lookup table from a preconstructed data file. For each interior bipolar grid edge $\mathbf{e}$, retrieve from the lookup table the two isocontour vertices associated with $\mathbf{e}$ and add an isocontour edge between those vertices. Assign geometric locations to the isocontour vertices based on the scalar values at the square edge endpoints. We explain the last two steps below.

Since each square vertex is either positive or negative and a square has four vertices, there are $2^4 = 16$ different configurations of square vertex labels. These configurations are listed in Figure 2.2 in Chapter 2.

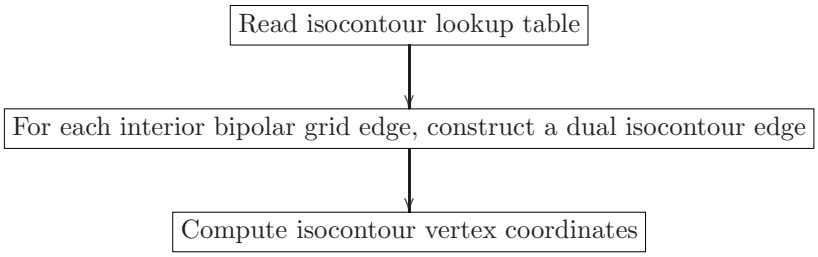


Figure 3.14. Dual Marching Squares.

The number and connectivity of the isocontour vertices within each square $\mathbf{c}$ is determined from the configuration of the square's vertex labels. If the configuration has two bipolar edges, then the configuration has a single vertex, $w_{\mathbf{c}}^1$, associated with both bipolar edges. This vertex will be an endpoint of the isocontour edges dual to each bipolar edge. If the configuration has four bipolar edges, then the configuration has two isocontour vertices, $w_{\mathbf{c}}^1$ and $w_{\mathbf{c}}^2$, one for each positive vertex. Associate the isocontour vertex corresponding to negative vertex v with the two square edges incident on v. Similarly, associate the isocontour vertex corresponding to negative vertex v' with the two square edges incident on v'. Each vertex will be an endpoint of two isocontour edges dual to the associated square edges. The isocontour generated for each configuration is illustrated in Figure 3.18.

The isocontour lookup table stores the association between isocontour vertices and square edges. An entry in the table is referenced as $\mathsf{Table}[\kappa, e]$, where κ is a configuration of square vertex labels and $\mathbf{e}$ is a bipolar edge of κ. Each entry is 1 or 2 corresponding to isocontour vertex w^1 or vertex w^2, respectively. If a configuration κ has two bipolar edges, $\mathbf{e}_1$ and $\mathbf{e}_2$, then $\mathsf{Table}[\kappa, \mathbf{e}_1] = \mathsf{Table}[\kappa, \mathbf{e}_2] = 1$. representing the single isocontour vertex, w^1, for configuration κ. If a configuration κ has four bipolar edges, $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$, and $\mathbf{e}_4$, and $\mathbf{e}_1$ and $\mathbf{e}_2$ share a negative vertex, and $\mathbf{e}_3$ and $\mathbf{e}_4$ share a negative vertex, then $\mathsf{Table}[\kappa, \mathbf{e}_1] = \mathsf{Table}[\kappa, \mathbf{e}_2] = 1$, representing vertex w^1, and $\mathsf{Table}[\kappa, \mathbf{e}_3] = \mathsf{Table}[\kappa, \mathbf{e}_4] = 2$, representing w^2. Vertices w^1 and w^2 are the two isocontour vertices for configuration κ.

The algorithm for positioning the isocontour vertices, $w_{\mathbf{c}}^m$, is similar to the one in Surface Nets, but uses only the bipolar edges associated with $w_{\mathbf{c}}^m$. For each bipolar edge $\mathbf{e}$, approximate the intersection $w_{\mathbf{e}}$ of $\mathbf{e}$ and the isocontour using linear interpolation as described in Section 1.7.2. Point $w_{\mathbf{e}}$ equals $(1-\alpha)p + \alpha q$ where $\alpha = (\sigma - s_p)/(s_q - s_p)$. Locate isocontour vertex $w_{\mathbf{c}}^m$ at the midpoint of line segment $(w_{\mathbf{e}}, w_{\mathbf{e}'})$, where $\mathbf{e}$ and $\mathbf{e}'$ are the edges associated with $w_{\mathbf{c}}^m$.

The Dual Marching Squares algorithm is presented in Figures 3.14–3.17. Function `LinearInterpolation`, called by this algorithm, is defined in Algorithm 1.1 in Section 1.7.2.

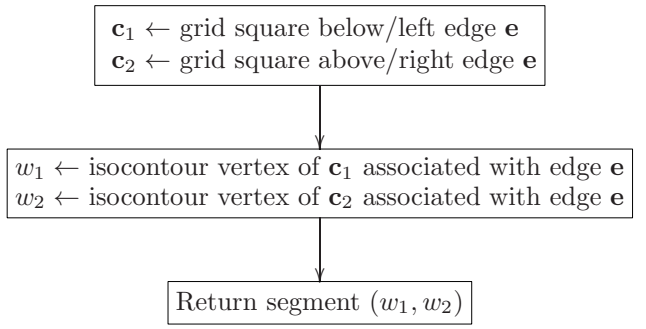


Figure 3.15. Construct isocontour edge dual to grid edge $\mathbf{e}$.

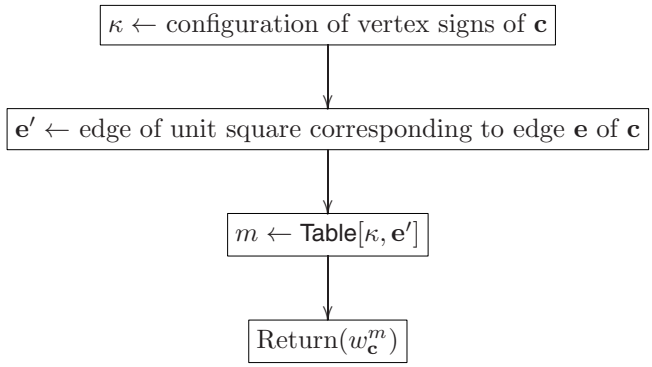


Figure 3.16. Return isocontour vertex of grid square $\mathbf{c}$ associated with edge $\mathbf{e}$.

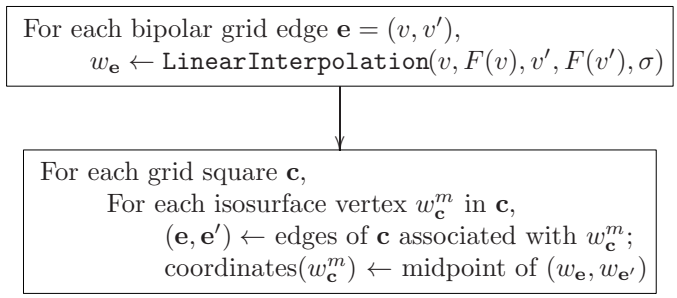


Figure 3.17. Compute vertex coordinates. $F(v)$ is the scalar value of v and σ is the isovalue.

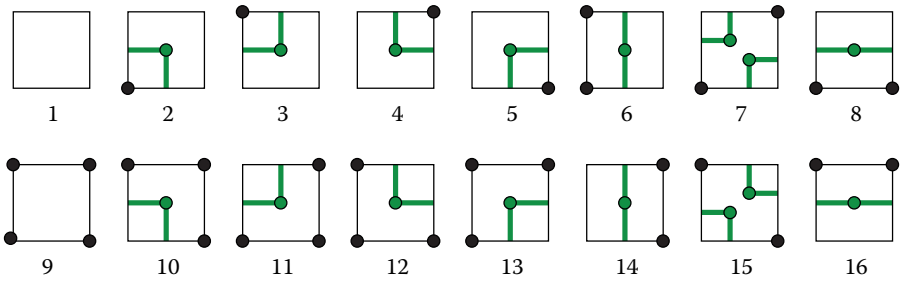


Figure 3.18. Dual Marching Squares isocontours.

3.3.2 3D Dual Marching Cubes

Input to the Dual Marching Cubes algorithm is an isovalue and a set of scalar values at the vertices of a three-dimensional regular grid. The algorithm has four steps. (See Figure 3.19.) Read in the isocontour lookup table from a preconstructed data file. For each interior bipolar grid edge **e**, retrieve from the lookup table the four isosurface vertices associated with **e** and add an isosurface quadrilateral between those vertices. Assign geometric locations to the isocontour vertices based on the scalar values at the square edge endpoints. Triangulate the isosurface quadrilaterals. We explain the last three steps next.

Since each cube vertex is either positive or negative and a cube has eight vertices, there are $2^8 = 256$ different configurations of cube vertex labels. These configurations are listed in Figure 2.15 in Chapter 2.

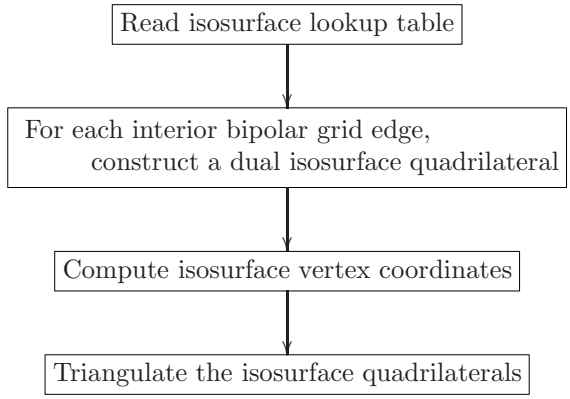


Figure 3.19. Dual Marching Cubes.

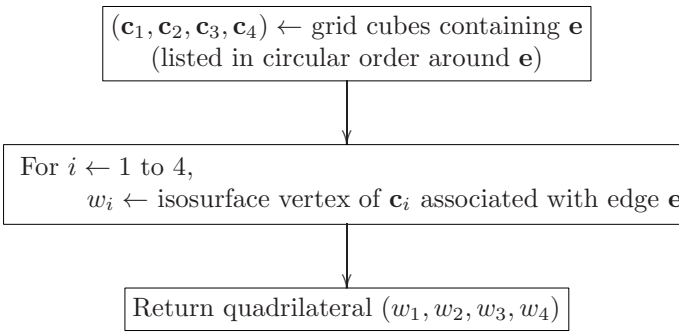


Figure 3.20. Construct isosurface quadrilateral dual to grid edge $\mathbf{e}$.

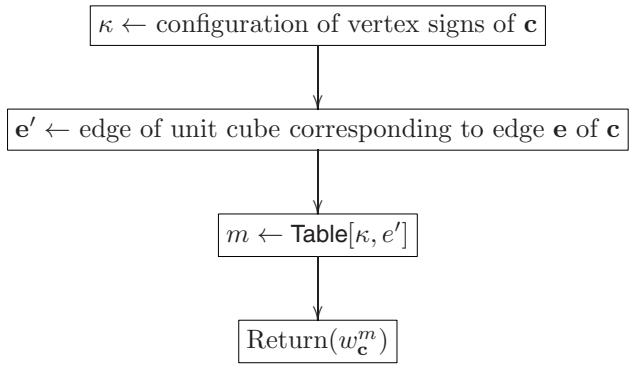


Figure 3.21. Return isosurface vertex of grid cube $\mathbf{c}$ associated with edge $\mathbf{e}$.

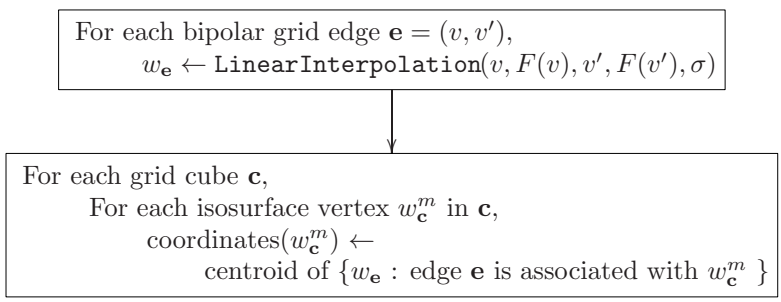


Figure 3.22. Compute vertex coordinates. $F(v)$ is the scalar value of v. σ is the isovalue.

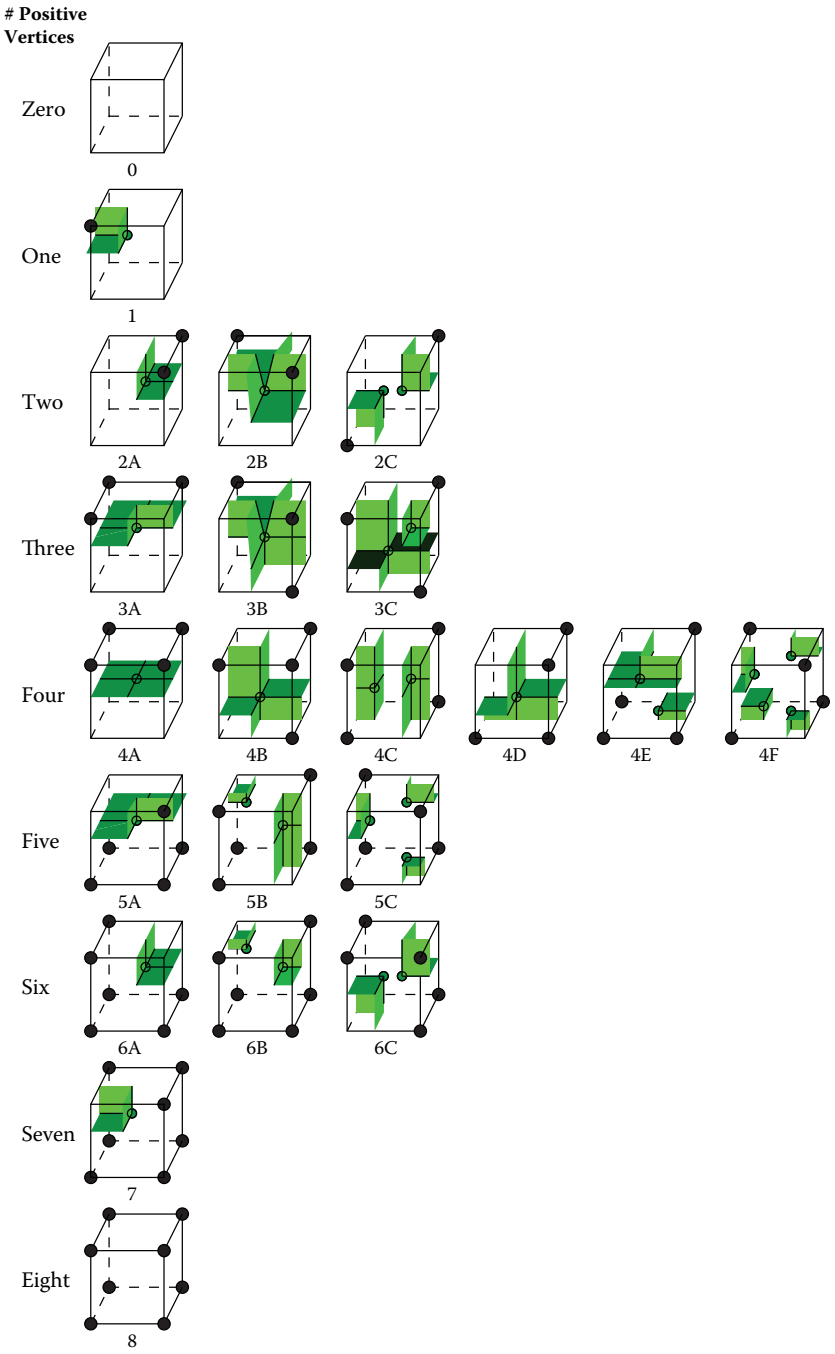


Figure 3.23. DUAL MARCHING CUBES isosurface patches.

The number and connectivity of the isosurface vertices within each cube is determined from the configuration of the cube's vertex labels. There are 22 distinct cube configurations after rotation and/or reflection. The isosurface vertices and edges for the twenty-two distinct configurations are represented in Figure 3.23.

Configuration 1 with a single positive vertex has a single isosurface vertex associated with its three bipolar edges. This vertex will be an endpoint of the isosurface edges dual to each bipolar edge. Configuration 2A with two adjacent positive vertices also has a single isosurface vertex associated with its four bipolar edges. Configuration 2C with two nonadjacent positive vertices, v and v', has two isosurface vertices. Isosurface quadrilaterals incident to one isosurface vertex intersect the three cube edges incident on v. Isosurface quadrilaterals incident to the other isosurface vertex intersect the three cube edges incident on v'. Configuration 4F generates four isosurface vertices, the most for any configuration.

The isosurface lookup table, Table stores the association between isosurface vertices and cube edges. An entry in the table is referenced as $\mathsf{Table}[\kappa, e]$, where κ is a configuration of cube vertex labels and $\mathbf{e}$ is a bipolar edge of κ. Each entry is 1, 2, 3, or 4 corresponding to isosurface vertex w_1, w_2, w_3, or w_4. An additional table, NumVertices, stores the number of isosurface vertices for configuration κ.

For instance, when κ is configuration 1 with three bipolar edges, $\mathbf{e}_1, \mathbf{e}_2$, and $\mathbf{e}_3$, then $\mathsf{Table}[\kappa, \mathbf{e}_1] = \mathsf{Table}[\kappa, \mathbf{e}_2] = \mathsf{Table}[\kappa, \mathbf{e}_3] = 1$ and $\mathsf{NumVertices}[\kappa] = 1$. When κ is configuration 2C with two positive grid vertices, v and v', with grid edges $\mathbf{e}_1, \mathbf{e}_2$, and $\mathbf{e}_3$ incident on v and grid edges $\mathbf{e}'_1, \mathbf{e}'_2$, and $\mathbf{e}'_3$ incident on v', then $\mathsf{Table}[\kappa, \mathbf{e}_1] = \mathsf{Table}[\kappa, \mathbf{e}_2] = \mathsf{Table}[\kappa, \mathbf{e}_3] = 1$ and $\mathsf{Table}[\kappa, \mathbf{e}'_1] = \mathsf{Table}[\kappa, \mathbf{e}'_2] = \mathsf{Table}[\kappa, \mathbf{e}'_3] = 2$ and $\mathsf{NumVertices}[\kappa] = 2$. (See Figure 3.24.)

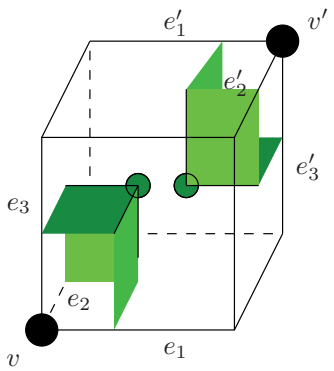


Figure 3.24. Configuration 2C: Grid edges intersected by the Dual Marching Cubes isosurface.

```
Input  : F is a 3D array of scalar values.
         Coord is a 3D array of (x, y, z) coordinates.
         σ is an isovalue.
Result : Set Quad of isosurface quadrilaterals and array VertexCoord of
         vertex coordinates.

DualMarchingCubes(F,Coord, σ, Quad)
1 Read Dual Marching Cubes lookup table into Table;
  /* Assign "+" or "−" signs to each vertex                          */
2 foreach grid vertex (i, j, k) do
3 |   if F[i, j, k] < σ then Sign[i, j, k] ← "−";
4 |   else Sign[i, j, k] ← "+";                        /* F[i, j, k] ≥ σ */
5 end
6 ConstructQuadrilaterals (Sign, Quad);
7 ComputeVertexCoordinates (F, Coord, σ, Sign, VertexCoord);
```

Algorithm 3.1. Dual Marching Cubes.

The algorithm for positioning the isosurface vertices is similar to the one in Surface Nets but uses only the bipolar edges associated with $w_{\mathbf{c}}^m$. For each bipolar edge $\mathbf{e}$, approximate the intersection $w_{\mathbf{e}}$ of $\mathbf{e}$ and the isosurface using linear interpolation as described in Section 1.7.2. Point $w_{\mathbf{e}}$ equals $(1-\alpha)p + \alpha q$ where $\alpha = (\sigma - s_p)/(s_q - s_p)$. Let $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k$ be the bipolar grid edges associated with isosurface vertex $w_{\mathbf{c}}^m$. Locate isosurface vertex $w_{\mathbf{c}}^m$ at the centroid, $(w_{\mathbf{e}_1} + w_{\mathbf{e}_2} + \cdots + w_{\mathbf{e}_k})/k$, of the points $w_{\mathbf{e}_1}, w_{\mathbf{e}_2}, \ldots, w_{\mathbf{e}_k}$, where $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k$ are the bipolar grid edges associated with isosurface vertex $w_{\mathbf{c}}^m$.

As with Surface Nets, we replace each quadrilateral around bipolar edge $\mathbf{e}$ by two or four triangles, depending on the envelope of $\mathbf{e}$. To construct the isosurface patch with two triangles, we use the diagonal that minimizes the maximum triangle angle.

The Dual Marching Cubes algorithm is presented in Algorithm 3.1. Function `LinearInterpolation`, called by this algorithm, is defined in Algorithm 1.1 in Section 1.7.2.

3.3.3 Lookup Table Duality

The Dual Marching Cubes lookup table in Figure 3.23 is dual to the Marching Cubes lookup table in Figure 2.16. Let κ be a configuration of positive and negative cube vertices.

Each connected component λ_{dual} of the isosurface patch for κ in the Dual Marching Cubes lookup table corresponds to a connected component λ of the isosurface patch for κ in the Marching Cubes lookup table. Each isosurface

```
ConstructQuadrilaterals(Sign, Quad)
Quad ← ∅;
foreach interior grid bipolar grid edge e do
    /* construct isosurface quadrilateral dual to e                    */
    for l = 1 to 4 do
        (i, j, k) ← index of l'th grid cube (in circular order) containing e;
        κ ← (Sign[i, j, k], Sign[i, j, k+1], ..., Sign[i+1, j+1, k+1]);
        m ← GetVertexIndex(i, j, k, e, κ);
        QuadVert[l] ← (i, j, k, m);
    end
    Quad ← Quad ∪ {(QuadVert[1], QuadVert[2], QuadVert[3], QuadVert[4])};
end

ComputeVertexCoordinates(F, Coord, σ, Sign, VertexCoord)
/* Approximate intersection of isocontour and bipolar edges            */
foreach bipolar grid edge e with endpoints (i1, j1, k1) and (i2, j2, k2) do
    w_e ← LinearInterpolation
          (Coord[i1, j1, k1], F[i1, j1, k1], Coord[i2, j2, k2], F[i2, j2, k2], σ);
end
/* Compute vertex coordinates                                          */
foreach grid cube (i, j, k) do
    κ ← (Sign[i, j, k], Sign[i, j, k+1], ..., Sign[i+1, j+1, k+1]);
    for m ← 1 to NumVertices[κ] do
        VertexCoord[i, j, m] ← (0, 0, 0);
        NumIncident[m] ← 0;
    end
    foreach bipolar edge e of grid cube (i, j, k) do
        m ← GetVertexIndex(i, j, k, e, κ);
        VertexCoord[i, j, k, m] ← VertexCoord[i, j, k, m] + w_e;
        NumIncident[m] ← NumIncident[m] + 1;
    end
    for m ← 1 to NumVertices[κ] do
        VertexCoord[i, j, m] ← VertexCoord[i, j, m]/NumIncident[m];
    end
end

Function GetVertexIndex(i, j, k, e, κ)
e' ← edge of unit cube corresponding to edge e in cube (i, j, k);
m ← Table[κ, e'];
return (m);
```

Algorithm 3.2. Subroutines for Dual Marching Cubes.

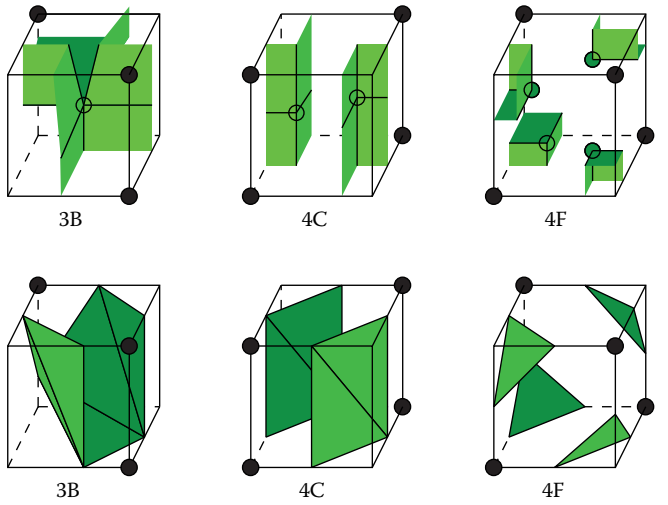


Figure 3.25. Examples of isosurface patches from the Dual Marching Cubes lookup table for configurations 3B, 4C, and 4F and the corresponding patches for Marching Cubes.

quadrilateral in λ_{dual} intersecting bipolar edge $\mathbf{e}$ corresponds to an isosurface vertex in λ lying on bipolar edge $\mathbf{e}$.

On the other hand, λ_{dual} contains one and only one isosurface vertex. Thus, each isosurface vertex of the isosurface patch for κ in the Dual Marching Cubes lookup table corresponds to a connected component of the isosurface patch for κ in the Marching Cubes lookup table.

Figure 3.25 contains three examples of isosurface patches for the Dual Marching Cubes lookup table and the corresponding isosurface patches for Marching Cubes.

3.3.4 Non-Manifold Isosurface Patches

By adding more than one isosurface vertex per grid cube, Dual Marching Cubes avoids many of the Surface Nets non-manifold constructions discussed in Section 3.2.8. Unfortunately, Dual Marching Cubes still has some non-manifold constructions. As shown in Figure 3.26, if cubes with configurations 2B and 3B are adjacent, then four isosurface quadrilaterals can meet in a single isosurface edge. The problem is that the two near-vertical edges in configurations 2B and 3B in Figure 3.23 have the same two endpoints, and so merge into a single isosurface edge.

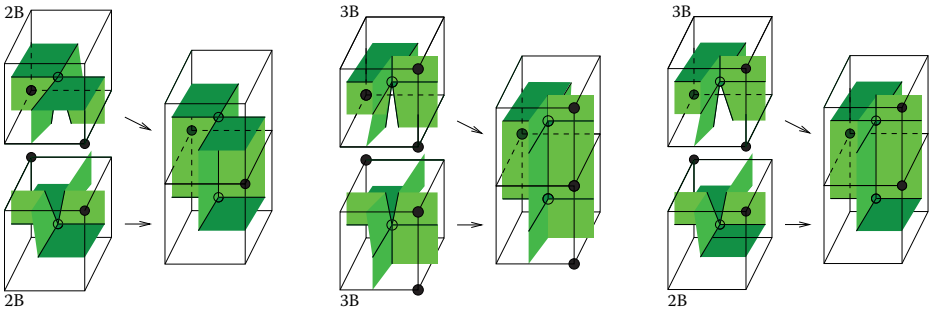


Figure 3.26. Dual Marching Cubes non-manifold constructions.

3.3.5 Manifold Dual Marching Cubes

The non-manifold problem described in the previous section could be solved by adding more complex isosurface patches for configurations 2B and 3B. However, there is a simple solution that avoids adding complexity to the isosurface patches.

Note that configurations 2B and 3B contain exactly one ambiguous facet. They are the only configurations in Figure 3.23 containing exactly one ambiguous facet. The only case in which four isosurface quadrilaterals share an edge is when a grid cube with configuration 2B or 3B shares its ambiguous facet with another grid cube with configuration 2B or 3B.

As discussed in Section 2.3.5 and illustrated in Figure 2.22, every ambiguous configuration has two "natural" Marching Cubes isosurface patches, one separating negative vertices and one separating positive vertices. Similarly, every ambiguous configuration has two "natural" Dual Marching Cubes isosurface patches. The two Dual Marching Cubes isosurface patches for configurations 2B and 3B are illustrated in Figure 3.27. For each configuration, one isosurface patch has a single connected component and the other isosurface patch has two connected components.

Manifold Dual Marching Cubes avoids creating non-manifold edges by using isosurface patches 2B-II and 3B-II instead of 2B-I and 3B-I whenever two cubes with configurations 2B and 3B share their ambiguous facets. Note that the isosurface patches for both cubes must be replaced, otherwise the patches will not align along their boundaries as discussed in Section 2.3.5. Moreover, if a cube $\mathbf{c}$ with configuration 2B or 3B shares an ambiguous facet with a cube $\mathbf{c}'$ with configuration other than 2B or 3B, isosurface patch 2B-I or 3B-I should be used. Cube $\mathbf{c}'$ will have two isosurface vertices connected by two distinct isosurface edges to the isosurface vertex in $\mathbf{c}$, and each isosurface edge will have only two incident quadrilaterals. If isosurface patch 2B-II or 3B-II were used for cube $\mathbf{c}$, then there would be misalignment between the isosurface patches in $\mathbf{c}$ and $\mathbf{c}'$.

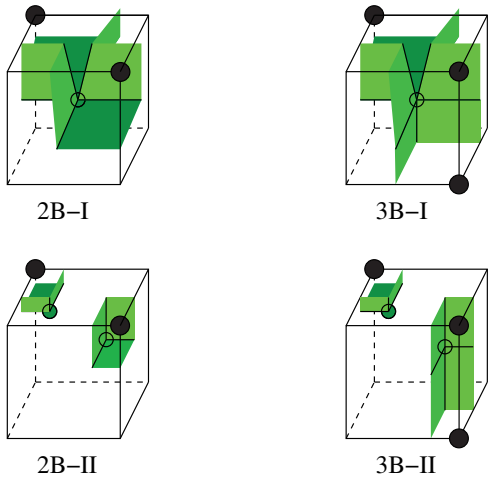


Figure 3.27. Isosurface patches 2B-I, 2B-II, 3B-I, and 3B-II for Manifold Dual Marching Cubes.

Manifold Dual Marching Cubes uses a modification of the algorithm in Figure 3.21 for retrieving the isosurface vertex of grid cube $\mathbf{c}$ associated with edge $\mathbf{e}$. Let Table1 be the Dual Marching Cubes isosurface lookup table as described in Section 3.3.2. Let Table2 be the same as Table1, except that configurations 2B and 3B use isosurface patches 2B-II and 3B-II. If cube $\mathbf{c}$ with configuration type 2B or 3B shares its ambiguous facet with a cube with configuration type 2B or 3B, then return Table2$[\kappa, e']$ instead of Table1$[\kappa, e']$ (Figure 3.28). All other routines are exactly the same.

3.3.6 Isosurface Properties

Properties 1–6 of the Manifold Dual Marching Cubes isosurface are the same as Properties 1–6 of the Surface Nets isosurface. The isosurface produced by Manifold Dual Marching Cubes depends on the placement of the vertices within each grid cube and the construction of the surface bounded by each quadrilateral. The properties listed below apply to Manifold Dual Marching Cubes where vertices are placed at the centroid of edge-isosurface intersection points and each quadrilateral is triangulated into two or four triangles. A quadrilateral around grid edge $\mathbf{e}$ is triangulated into two triangles, Δ_1 and Δ_2, using the diagonal that minimizes the maximum triangle angle. A quadrilateral is triangulated into four triangles if Δ_1 or Δ_2 are not in the envelope of $\mathbf{e}$.

Manifold Dual Marching Cubes returns a finite set, Υ, of triangles. The isosurface is the union of these triangles. The vertices of the isosurface are

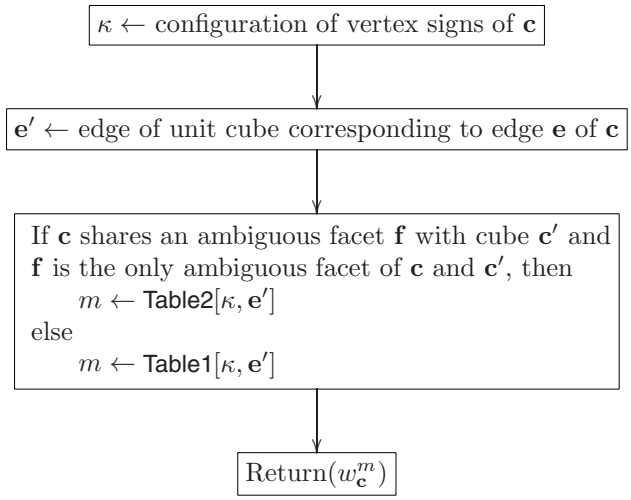


Figure 3.28. MANIFOLD DUAL MARCHING CUBES: Return isosurface vertex of grid cube $\mathbf{c}$ associated with edge $\mathbf{e}$.

the triangle vertices. As defined in Section 3.1, Γ_{inner} is the subgrid of interior grid cubes of Γ.

The following properties apply to all isosurfaces produced by MANIFOLD DUAL MARCHING CUBES.

Property 1. *The isosurface is piecewise linear.*

Property 2. *Set Υ contains at most $4m$ triangles where m is the number of bipolar grid edges.*

Property 3. *The isosurface does not contain any grid vertices whose scalar values do not equal the isovalue.*

Property 4. *The isosurface intersects at exactly one point every interior bipolar grid edge whose positive endpoint is strictly positive.*

Property 5. *The isosurface does not intersect any negative or strictly positive grid edges.*

Property 6. *The isosurface strictly separates the strictly positive grid vertices of Γ_{inner} from the negative vertices of Γ_{inner}.*

Properties 3–5 imply that the isosurface intersects a minimum number of grid edges.

As defined in Section 3.1, the Γ_{outer} is the set of boundary grid cubes of Γ. The following properties apply to a Manifold Dual Marching Cubes isosurface whose isovalues do not equal the scalar value of any grid vertex.

Property 7. *The isosurface is a piecewise linear, orientable 2-manifold with boundary.*

Property 8. *The boundary of the isosurface is contained in* Γ_{outer}.

Property 9. *Set* Υ *does not contain any zero-area triangles or duplicate triangles and the triangles in* Υ *form a triangulation of the isosurface.*

3.3.7 Proof of Isosurface Properties

The proof of Properties 1–6 are similar to the proofs in Sections 3.2.5–3.2.7 for Surface Nets and are omitted.

We first prove that Υ is a triangulation of the isosurface (Property 9). To prove this property, we need some lemmas about the separation of isosurface vertices and about the separation of envelopes around bipolar edges.

Lemma 3.14. *Let* $\mathbf{e}$ *and* $\mathbf{e}'$ *be distinct bipolar edges of cube* $\mathbf{c}$. *Let* w *and* w' *be distinct isosurface vertices in cube* $\mathbf{c}$ *where* w *is associated with edge* $\mathbf{e}$ *and* w' *is associated with edge* $\mathbf{e}'$. *If no vertex of* $\mathbf{c}$ *has scalar value equal to the isovalue, then some plane separates* $w \cup \mathbf{e}$ *from* $w' \cup \mathbf{e}'$ *and does not contain* w *or* w'.

Proof: Cube $\mathbf{c}$ must have a configuration that contains at least two distinct isosurface vertices. For configurations 2B, 2C, 3B, 3C, 4E, 4F, 5B, 5C, 6B, and 6C, either w or w' are incident on exactly three isosurface quadrilaterals. (Configurations 2B and 3B have two isosurface vertices when we use isosurface patches 2B-II and 3B-II.) Without loss of generality, assume that w is incident on exactly three isosurface quadrilaterals.

Isosurface vertex w is the centroid of exactly three points on three cube edges, and these three cube edges are incident on a single grid vertex v. The convex hull of v and its three incident cube edges are a tetrahedron. Let $\mathbf{f}$ be the facet of this tetrahedron that does not contain v, and let $\mathbf{h}$ be the plane containing $\mathbf{f}$. Plane $\mathbf{h}$ separates $w \cup \mathbf{e}$ from $w' \cup \mathbf{e}'$. Because no vertex of $\mathbf{c}$ has scalar value equal to the isovalue, point w is the centroid of three points that lie on one side of $\mathbf{h}$, so w is not on $\mathbf{h}$. Similarly, point w' is the centroid of points that lie on the other side of $\mathbf{h}$, so w' is not on $\mathbf{h}$.

For configurations 4C, let $\mathbf{h}$ be the plane through the two positive grid edges. Plane $\mathbf{h}$ separates $w \cup \mathbf{e}$ from $w' \cup \mathbf{e}'$. Since no vertex of $\mathbf{c}$ has scalar value equal to the isovalue, point w is the centroid of four points that lie on one side of $\mathbf{h}$, and point w' is the centroid of four points that lie on the other side. Thus, neither w nor w' lie on $\mathbf{h}$. □

Each interior bipolar edge $\mathbf{e}$ is associated with four isosurface vertices, one in each grid cube containing $\mathbf{e}$. These four isosurface vertices determine an envelope around $\mathbf{e}$ as described in Section 3.2.3.

Lemma 3.15. *The envelopes of two different interior bipolar grid edges intersect only at shared vertices, edges, or facets.*

Proof: Let $\mathbf{e}$ and $\mathbf{e}'$ be two different interior bipolar grid edges. Let $\mathbf{c}$ be a grid cube containing both $\mathbf{e}$ and $\mathbf{e}'$. Let w be the isosurface vertex in cube $\mathbf{c}$ associated with $\mathbf{e}$. Let w' be the isosurface vertex in cube $\mathbf{c}'$ associated with $\mathbf{e}'$. By Lemma 3.14, if w does not equal w', then there is some plane that separates $w \cup \mathbf{e}$ from $w' \cup \mathbf{e}'$. This plane separates the two envelopes. The two envelopes may only intersect if they share isosurface vertices and/or a vertex at $\mathbf{e} \cap \mathbf{e}'$. If they share such vertices, then their intersection is a vertex, edge, or facet determined by such shared vertices. □

Property 9. *Set Υ does not contain any zero-area triangles or duplicate triangles and the triangles in Υ form a triangulation of the isosurface.*

The proof that set Υ does not contain any zero-area triangles or duplicate triangles is exactly the same as the proof in Section 3.2.7 for Surface Nets, and is omitted.

Proof that set Υ is a triangulation: Each envelope around an interior bipolar edge contains two or four isosurface triangles. The nonempty intersection of any two triangles within an envelope is a vertex or edge of each. By Lemma 3.15, envelopes of different bipolar edges intersect only at shared vertices, edges, or facets. Thus, the triangles in different envelopes intersect only at shared edges or vertices. □

Finally, we prove that the isosurface is a 2-manifold.

Property 7. *The isosurface is a piecewise linear, orientable 2-manifold with boundary.*

Property 8. *The boundary of the isosurface is contained in Γ_{outer}.*

Proof: Since Υ is a triangulation of the isosurface, we need only check that every isosurface edge is contained in at most two isosurface triangles and that the neighborhood of every isosurface vertex is a disk. Inspection of the isosurface patches in Figure 3.23 shows that this is true for all configurations except for configurations 2B and 3B. As discussed in Section 3.3.4, when a cube $\mathbf{c}$ with configuration 2B or 3B shares its ambiguous facet with another cube $\mathbf{c}'$ with configuration 2B or 3B, an isosurface edge is shared by four isosurface triangles. However, in this case, the Manifold Dual Marching Cubes algorithm replaces isosurface patches 2B-1 or 3B-I with 2B-II or 3B-II, respectively, in both

cubes. Each edge in these isosurface patches lies in only two triangles, and the neighborhood of each vertex is a disk. Because $\mathbf{c}$ and $\mathbf{c}'$ have only one ambiguous facet, the new isosurface patches in $\mathbf{c}$ and $\mathbf{c}'$ still align with all of their neighbors.

Let w be an isosurface vertex in grid cube $\mathbf{c} \in \Gamma_{\text{inner}}$. The triangles containing w form a topological disk, and w is in the interior of this disk. Every isosurface edge incident on w is contained in exactly two triangles. Thus, neither w nor any isosurface edge incident on w is on the boundary of the isosurface. The only vertices on the boundary of the isosurface are isosurface vertices contained in Γ_{outer}. The only edges on the boundary of the isosurface are edges connecting vertices in Γ_{outer} and are also contained in Γ_{outer}. Thus, the boundary of the isosurface is contained in Γ_{outer}. □

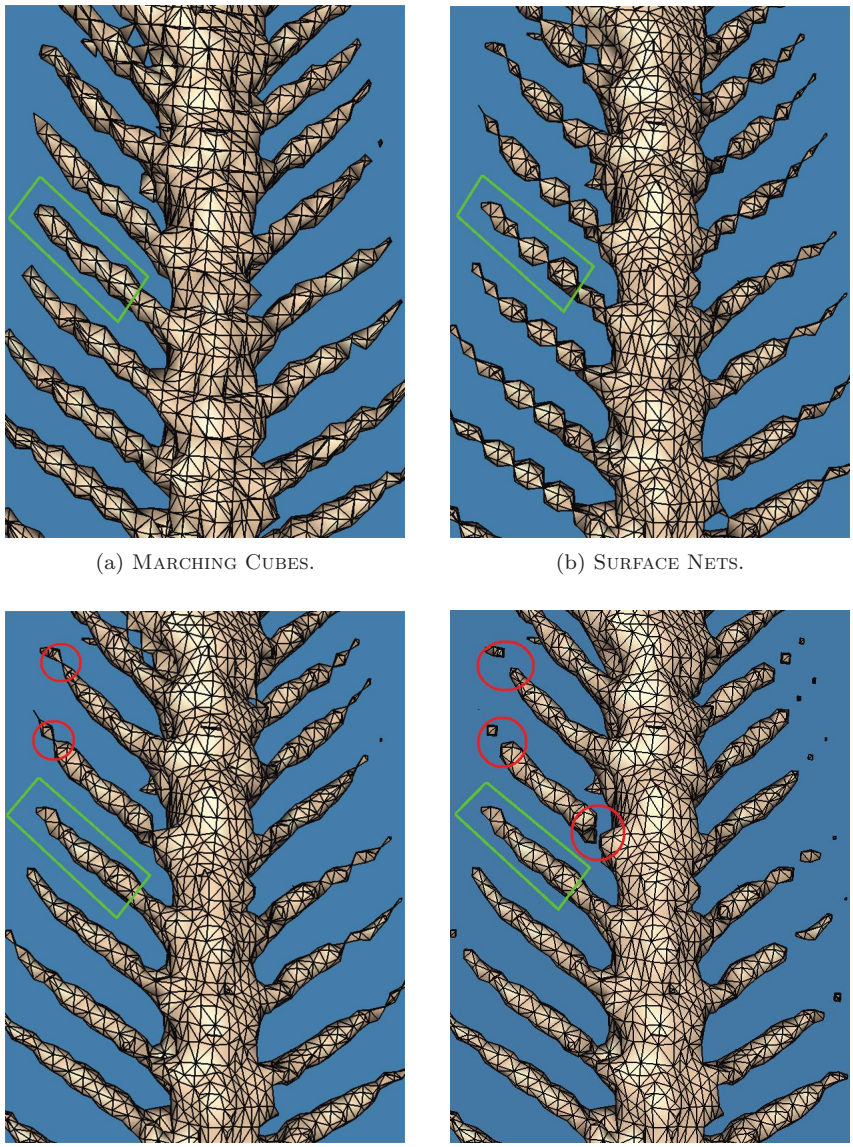
3.4 Comparison of Algorithms

Table 3.1 compares isosurfaces constructed by Marching Cubes, Surface Nets, and Manifold Dual Marching Cubes on engine, carp, and tooth data sets. The vertex and triangle counts for Nielson's original Dual Marching Cubes are almost exactly the same as for Manifold Dual Marching Cubes and are omitted. As can be seen, the number of isosurface vertices and triangles is about the same in each. The only significant difference is Tooth with isovalue 386. This is a particularly noisy isosurface with many small components of noise.

Figure 3.29 displays isosurfaces constructed by Marching Cubes, Surface Nets, Dual Marching Cubes, and Manifold Dual Marching Cubes.

Data set	Isovalue	Algorithm	# Vertices	# Triangles
Engine	80.5	Marching Cubes	298,000	594,000
Engine	80.5	Surface Nets	297,000	592,000
Engine	80.5	Manifold Dual MC	297,000	592,000
Carp	1150.5	Marching Cubes	670,000	1,342,000
Carp	1150.5	Surface Nets	662,000	1,340,000
Carp	1150.5	Manifold Dual MC	672,000	1,340,000
Tooth	386.5	Marching Cubes	3,374,000	6,898,000
Tooth	386.5	Surface Nets	2,904,000	6,690,000
Tooth	386.5	Manifold Dual MC	3,345,000	6,690,000
Tooth	640.5	Marching Cubes	64,000	128,000
Tooth	640.5	Surface Nets	64,000	128,000
Tooth	640.5	Manifold Dual MC	64,000	128,000

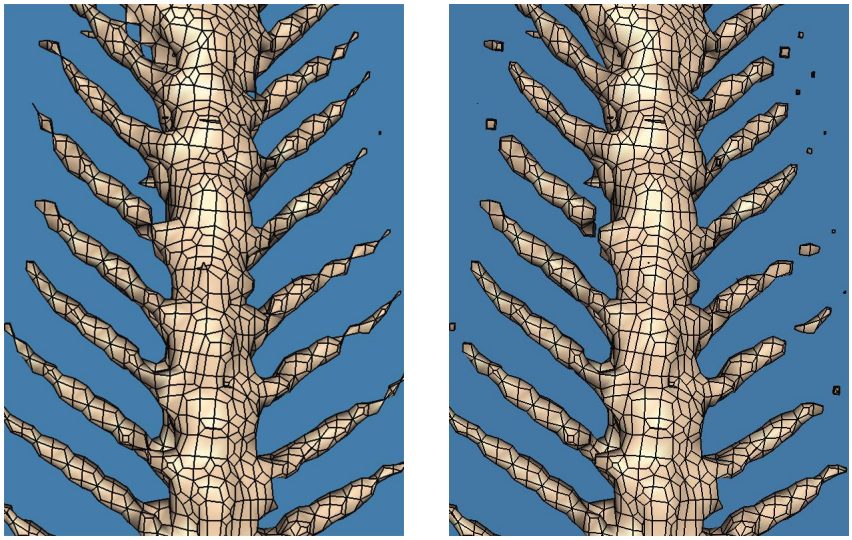
Table 3.1. Comparison of isosurfaces produced by Marching Cubes, Surface Nets, and Manifold Dual Marching Cubes, including the number of vertices and number of triangles in each isosurface.



(a) Marching Cubes. (b) Surface Nets.

(c) Dual Marching Cubes. (d) Manifold Dual MC.

Figure 3.29. Skeleton of the tail of a carp. Subsampled (8:1) carp data set. Isovalue 1150.5. (a) Marching Cubes isosurface. (b) Surface Nets triangulated isosurface. (c) Dual Marching Cubes triangulated isosurface. (d) Manifold Dual Marching Cubes triangulated isosurface. (Data set was created by Michael Scheuring, Computer Graphics Group, University of Erlangen, Germany.)



(a) Dual Marching Cubes. (b) Manifold Dual MC.

Figure 3.30. Isosurface of quadrilaterals from the carp data set, isovalue 1150.5. (a) Dual Marching Cubes isosurface. (b) Manifold Dual Marching Cubes isosurface.

The isosurface is the skeleton of a carp. The isovalue, 1150.5, does not equal the scalar value of any grid vertex, so the Marching Cubes and Manifold Dual Marching Cubes isosurfaces are manifolds. (The scalar values of the grid vertices are all integers.) The data set was subsampled (8:1) to better show the mesh edges and to illustrate the differences among the algorithms. The isosurface is particularly challenging for isosurface reconstruction algorithms because of the narrow bone structures. (The subsampling adds to the difficulty.) Figure 3.30 displays the quadrilaterals in the Dual Marching Cubes and Manifold Dual Marching Cubes isosurfaces.

Visual comparison of the Marching Cubes isosurface (Figure 3.29(a)) with the other isosurfaces (Figures 3.29(b), (c), and (d)) shows many more thin, small angle triangles in the Marching Cubes isosurface. The small, thin triangles create distracting visual artifacts in the isosurface. They also make the mesh unsuitable for finite element methods [Shewchuk, 2002].

The Marching Cubes isosurface is a manifold, whereas the Surface Nets and Dual Marching Cubes isosurfaces are not. Many of the long bones are pinched in the Surface Nets isosurface. (Compare the rectangular green regions in Figures 3.29(a), (b), and (c).) The Dual Marching Cubes isosurface does a much better job of modeling the fine bone structure than the Surface Nets algorithm does. However, even the Dual Marching Cubes has some pinching to a line segment. (See regions circled in red in Figure 3.29(c).) This

pinching is caused by the non-manifold cases discussed in Section 3.3.4 and illustrated in Figure 3.26.

The Manifold Dual Marching Cubes isosurface is a manifold. The non-manifold pinched regions have been eliminated by changes in the isosurface patches around those regions. However, those changes have also changed the isosurface topology, disconnecting some of the bones. (See regions circled in red in Figure 3.29(d).)

We should note that the pinching in Figure 3.29 is a result of the subsampling of the data set. As shown in Figures 3.31(b) and (c), the full-resolution Surface Nets and Dual Marching Cubes isosurfaces have no such pinching and look similar to the full-resolution Marching Cubes isosurface (Figure 3.29(a)). Similarly, the full-resolution Manifold Dual Marching Cubes isosurface in Figure 3.31(d) does not have disconnected bones and also looks similar to the other three isosurfaces.

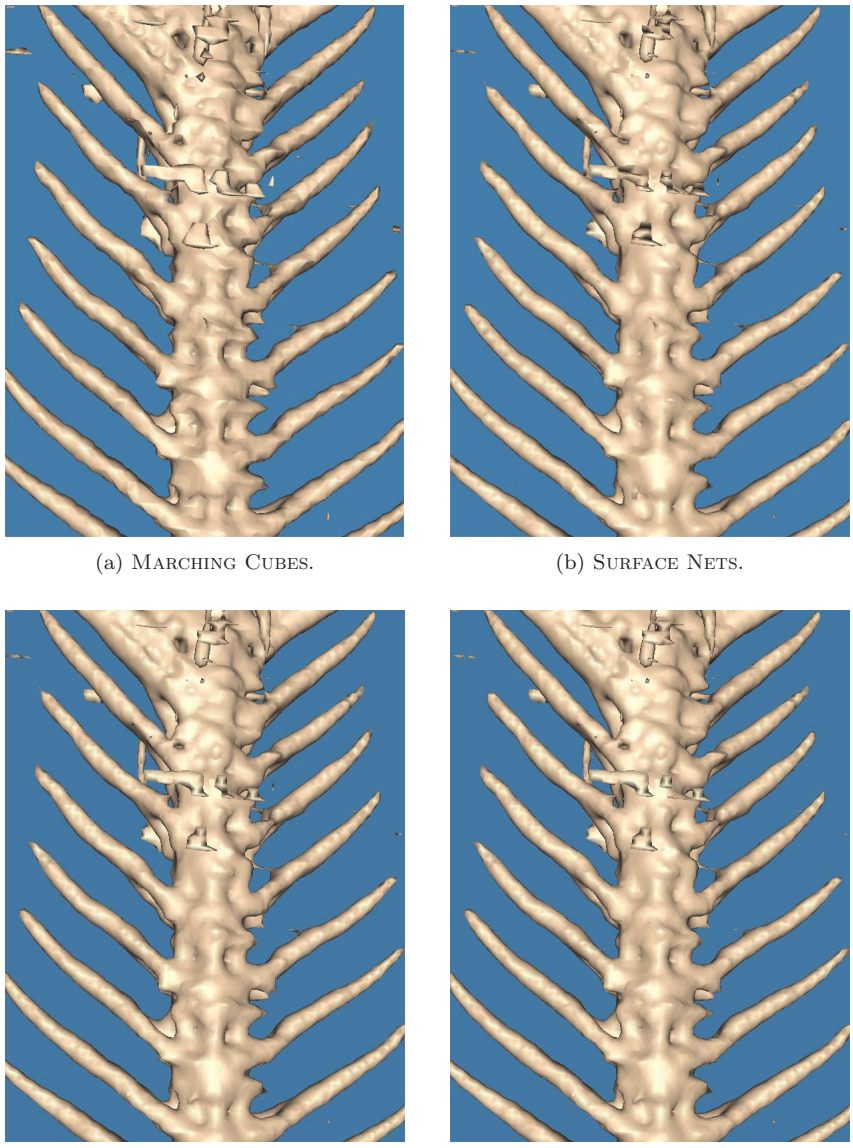
The pinching in Figure 3.29 is also a result of the long, narrow bone structure being modeled by the isosurface. For many isosurfaces, the subsampled Surface Nets isosurface has no such problems.

Even though the full-resolution Surface Nets isosurface in Figure 3.31(b) looks much better than its subsampled version, the isosurface still has about 8,000 non-manifold edges and 1,100 non-manifold vertices. Even the full-resolution Dual Marching Cubes isosurface has 875 non-manifold edges. The Manifold Dual Marching Cubes algorithm changes the isosurface patches in 1,750 ($= 2 \times 875$) cubes from 2B-I or 3B-I to 2B-II or 3B-II to avoid such non-manifold edges.

3.5 Notes and Comments

Gibson, in [Gibson, 1998a, Gibson, 1998b], introduced the Surface Nets algorithm. The term *dual contouring* was used in [Ju et al., 2002] to describe isosurface construction algorithms similar to Surface Nets, which place vertices inside grid cubes.

Surface normals at the intersection points of grid edges and the isosurface can be used to locate isosurface vertices on sharp corners or edges in the isosurface. Using these surface normals with dual contouring, Ju et al. [Ju et al., 2002] constructed isosurfaces with sharp features. Further details of their algorithm are included in [Schaefer and Warren, 2002]. To allow finer control over small features, Varadhan et al. [Varadhan et al., 2003] modified the dual contouring algorithm to model two intersections of the isosurface and each grid edge. Zhang, Hong, and Kaufman [Zhang et al., 2004] modified the algorithm further to model even more complicated structures within a grid cube. These papers assume that the input includes a continuous scalar function, either explicitly defined or



(a) Marching Cubes. (b) Surface Nets.

(c) Dual Marching Cubes. (d) Manifold Dual MC.

Figure 3.31. Full-resolution isosurfaces from the carp data set, isovalue 1150.5. (a) Full-resolution Marching Cubes isosurface. (b) Full-resolution Surface Nets isosurface. (c) Full-resolution Dual Marching Cubes isosurface. (d) Full-resolution Manifold Dual Marching Cubes isosurface.

defined as the signed distance to a polygonal mesh, and they compute the surface normals using that function.

Dual contouring algorithms are easily adaptable to multiresolution isosurface construction. Constructing multiresolution isosurfaces using dual contouring is discussed in Section 10.2.

Dual Marching Cubes by Nielson [Nielson, 2004] reads a "dual contouring" lookup table to place and connect isosurface vertices in grid cubes. The Dual Marching Cubes lookup table is "dual" to the Marching Cubes lookup table. Schaefer, Ju, and Warren [Schaefer et al., 2007] generalize Nielson's algorithm to multiresolution isosurfaces.

Ashida and Badler [Ashida and Badler, 2003] and Greß and Klein [Greß and Klein, 2004] give dual contouring algorithms that produce isosurfaces similar to the Dual Marching Cubes isosurface with multiple isosurface vertices in a grid cube. Their algorithms determine connectivity between isosurface vertices directly from the grid without using a lookup table.

Ju and Udeshi [Ju and Udeshi, 2006] discuss the problem of self-intersections in the dual contouring isosurface caused by triangulating the isosurface quadrilateral. They present a solution by adding a vertex on the bipolar edge surrounded by the quadrilateral and adding four triangles incident on this vertex. Wang [Wang, 2011] describes orientation tests for determining when four triangles are necessary.