Remeshing of Multiresolution Regular Grids

Ramakrishnan Kazhiyur-Mannar Rephael Wenger

Dept. of Computer and Information Science
The Ohio State University
Columbus, Ohio 43210
E-mail:{ramakris,wenger}@cis.ohio-state.edu

ABSTRACT

Isosurface reconstruction algorithm applied to adjacent rectangular regions of different sizes may create cracks in the isosurface between the regions. We avoid this cracking by contracting some edges of the smaller regions, repositioning hanging vertices and faces so that adjacent regions intersect properly. The convex hull based isosurface reconstruction algorithm applied to this mesh will create a crack-free isosurface. By adding a preprocessing splitting step, we can also guarantee that our edge contraction will not change the isosurface topology.

1 GRID COLLAPSE

Consider a regular d-dimensional rectangular region partitioned into axis-parallel rectangular regions called *boxes*. Adjacent boxes may have different sizes and the intersection of hypercubes may be different from their faces. Our goal is to modify this partition into a mesh so that the intersection of adjacent mesh elements is a face of each. Applying the convex hull based isosurface reconstruction algorithm to this mesh will produce a crack free surface.

Our grid contraction algorithm can be broken into three steps. The first is a "balancing" step which ensures that adjacent rectangles have dimensions which are not too different. The second is a splitting step which propogates the subdivision in each direction. The third is a vertex relocation which relocates "hanging" vertices so that adjacent partition elements intersect on their faces.

We say that a set of line segments in $\mathbb{R}^1$ is *balanced* if, whenever two line segments intersect at more than a point, either the two line segments are equal or the endpoints of one line segment are the endpoint and the midpoint of the other. In the latter case, one line segment has exactly half the length of the other.

A partition into boxes is *balanced* if, for every axis $\mathbf{x}_i$, the projection of the boxes onto $\mathbf{x}_i$ is balanced. In other words, whenever two boxes intersect along some dimension, the length of one box along that dimension is either equal to or half the length of the other and the boxes are aligned along that dimension.

We require that our partition into boxes be *balanced*. This requirement controls the geometric distortion in the resulting mesh. Our algorithm ensures that boxes are aligned along each dimension by only creating boxes on certain boundaries. It ensures the size restriction by splitting boxes which are too long compared to their neighbors.

A box vertex is called *hanging* if it lies on the boundary of some adjacent box but is not a vertex of that box. A box vertex is *proper* if it is not hanging. The second step in our algorithm splits boxes at hanging vertices.

For each box B containing a hanging vertex v which is not a vertex of B, we split B by d distinct axis parallel hyperplanes through the center of B into 2^d boxes. The splitting step may create new hanging vertices and does not solve the meshing problem. However, any of the original vertices are now proper. This step controls the amount of geometric distortion induced at the final step which moves only the hanging vertices.

Proposition 1. *Let P be a partition of a rectangular region in $\mathbb{R}^d$ into axis parallel boxes. If each box B containing a hanging vertex v which is not its vertex is split by the d distinct axis parallel hyperplanes through the center of B, then any vertex of P is a proper vertex of the new partition P'.*

Proof. Any $k > 0$ dimensional face f in P which contains a vertex v is split into 2^k faces. Vertex v is a vertex of each of those faces and of the boxes in P' containing those faces. Thus vertex v is not in the interior of any face of P' and is proper. □

The last step moves hanging vertices to nearby proper vertices. Each hanging vertex lies in the interior of one or more box faces. Since each box face is axis parallel, one vertex of the box has lowest coordinates among all face vertices. Let $L(f)$ be the vertex of face f with lowest coordinate. For each hanging vertex v, choose the face f with largest dimension which contains the hanging vertex in its interior and merge v with the vertex $L(f)$.

Vertex $L(f)$ may itself be a hanging vertex lying on some face f' of a box and thus merged with vertex $L(f')$. This merger implicitly merges v and $L(f')$. The three merged vertices $v, L(f)$ and $L(f')$ are located at the vertex with lowest coordinates $L(f')$. Of course, if $L(f')$ is also a hanging vertex, then it will be merged with some vertex with lower coordinates.

The moving of hanging vertices implicitly moves the edges with endpoints at those vertices. It also changes the higher dimensional faces containing those vertices. However, the moved vertices of a k-dimensional face may no longer lie on a k-dimensional affine subspace. For instance, the four vertices defining a cube's facet may no longer be co-planar. Thus the new k-face is no longer simply the convex hull of its vertices but is some deformation of the k-dimensional face of a hypercube. This is not different than curvilinear meshes in $\mathbb{R}^3$ where grid elements are combinatorially cubes but the facets are not necessarily planar.

We will show that the final mesh is proper, in the sense that intersection of adjacent mesh elements is a face of each. To prove this, we show that our final mesh can be produced by a specific set of box collapses applied to a regular mesh.

Given a partition P of a rectangular region into axis parallel boxes, let P^* be the partition formed by passing a hyperplane through every vertex of P. Partition P^* is combinatorially and topologically equivalent to a regular grid, although the the rectangular elements in P^* do not have uniform size. Each box in P is subdivided into a set of smaller boxes in P^*.

Each box B has two facets orthogonal to a given axis $\mathbf{x}_i$. Let $F^+(B, \mathbf{x}_i)$ be the facet orthogonal to $\mathbf{x}_i$ with larger x_i coordinates and $F^-(B, \mathbf{x}_i)$ be the orthogonal facet with smaller x_i coordinates. Define *collapsing* a box along axis $\mathbf{x}_i$ as mapping $F^+(B, \mathbf{x}_i)$ onto $F^-(B, \mathbf{x}_i)$. A box can be simultaneously collapsed along multiple axes. Similarly, for any face f of B which is not orthogonal to $\mathbf{x}_i$, $F^+(f, \mathbf{x}_i)$ and $F^-(f, \mathbf{x}_i)$ are the $(k-1)$-dimensional faces of f

orthogonal to $\mathbf{x}_i$. *Collapsing* f along $\mathbf{x}_i$ is mapping $F^+(f, \mathbf{x}_i)$ to $F^-(f, \mathbf{x}_i)$.

Collapsing a box along axis $\mathbf{x}_i$ is equivalent to collapsing all the box edges parallel to $\mathbf{x}_i$. The collapse of a block in P^* induces a collapse of the faces of adjacent blocks. It distorts the mesh but does not change the connectivity between mesh elements. In particular, mesh elements are still joined at their faces.

We define a particular set of collapses applied to the boxes in P^* and show that they generate a mesh equivalent to the one produced by moving hanging vertices. Each box $b \in P^*$ is contained in some box $B \in P$. Collapse box b along axis $\mathbf{x}_i$ if $F^+(b, \mathbf{x}_i)$ is not a subset of $F^+(B, \mathbf{x}_i)$. In other words, collapse b along $\mathbf{x}_i$ if b does not rest on facet $F^+(B, \mathbf{x}_i)$ of B. Let $\mathcal{C}P$ be the mesh produced after applying all these collapses to the boxes in P^*.

Note that the only box of P^* in B which does not collapse at all is the one sharing the vertex of B with largest coordinates is all directions.

2 Isosurface Reconstruction

The edge collapse described in the previous section eliminates hanging vertices from the mesh and guarantees that the elements of the image intersect properly in their faces. However, the edge collapse changes the elements of the mesh so that they are no longer hypercubes. In fact, they are no longer polyhedra and may not be convex. How do we reconstruct the isosurface in such a mesh?

One potential way to do so is to treat each mesh element as a hypercube with degenerate edges. Using the Marching Cubes lookup table [3] or its variants [4, 5, 1, 2], one could construct an isosurface patch for each mesh element and then move the vertices of the patch to correspond to the movement of vertices and edges caused by the edge collapse. Unfortunately, the edge collapse of the regular grid can induce edge collapses in the isosurface patches. These edge collapses can, in certain cases, destroy the manifold property of the isosurface. While it is potentially possible to check for such "bad" collapses and handle such cases specially, we propose an alternate solution.

Instead of constructing the isosurfaces from the uncollapsed hypercubes, we perform a collapse on each hypercube separately and then construct the isosurface patch within each collapsed hypercube. The collapsed hypercube is not necessarily a polyhedron. We show that nevertheless the edges of the original map to edges of the convex hull of the vertices of the collapsed hypercube. Thus, the algorithm in [1, 2] can be used to construct an isosurface patch in the collapsed hypercube.

Given an edge (p_1, p_2) of a hypercube, *collapsing* edge (p_1, p_2) means deleting edge (p_1, p_2) from the 1-skeleton of the hypercube and identifying vertices p_1 and p_2. We say that vertex p_1 *collapses* to vertex p_2 if edge (p_1, p_2) collapses and vertex p_1 is mapped to the location of p_2. We also say that vertex p_1 *collapses* to vertex p_k if there is a sequence of edges, (p_1, p_2), (p_2, p_3), ..., (p_{k-1}, p_k) such that p_i collapses to p_{i+1}) for $i \leq k - 1$.

Assume that the hypercube facets are normal to the coordinate axes, x_i. The hypercube edges are parallel to those axes. We say that vertex p_1 *collapses* in coordinate axis direction $\vec{x_i}$ if p_1 collapses to p_2 and edge (p_1, p_2) is parallel to axis x_i in the hypercube. Note that edges collapses may move p_1 or p_2 but we consider the original orientation of edge (p_1, p_2) in determining if it is parallel to axis x_i.

Collapsing edges of the hypercube deforms the hypercube faces, many of which will no longer be convex polyhedra. However, the faces remain cells (homeomorphic to the interior of a ball,) and thus their topological behaviour is the same. Collapsing multiple edges can also cause the identification of different edges of the hypercube.

Given a hypercube whose facets are normal to the coordinate axes, we say that vertex p_1 *dominates* vertex p_2 if every coordinate of vertex p_1 is greater than or equal to the corresponding coordinate of vertex p_2.

We claim that the set of hypercube edge collapses in the previous section satisfies the following conditions:

1. If vertex p_1 does not dominate vertex p_2, then p_1 does not collapse to p_2.

2. If p_1 collapses to p_2, then any point q which dominates p_2 and is dominated by p_1 also collapses to p_2.

3. For every hypercube edge (p_1, p_2), either p_1 collapses in every direction in which p_2 collapses, or p_2 collapses in every direction in which p_1 collapses.

Condition 1 is satisfied by the way edges are collapsed. Condition 2 is satisfied since p_1 collapses to $L(f)$ where f is the largest face containing p_1 on its interior. If q dominates $L(f)$ and is dominated by p_1, then some subface f' of f contains q in its interior. Since p_1 dominates q, $L(f')$ must equal $L(f)$ and so q maps to $L(f)$.

The step down operation ensures Condition 3. If either p_1 or p_2 are proper, then either p_1 or p_2 does not collapse and Condition 3 is trivially satisfied. If both p_1 and p_2 are hanging, then both vertices were created in the partition of box B at step down. Vertices p_1 and p_2 are located at the centers of faces f_1 and f_2 of B, respectively. Either f_2 is a subface of f_1 or vice versa. Without loss of generality, assume that f_2 is a subface of f_1. After step down, for any face which contains p_1 in its interior, there is a parallel face containing p_2 in its interior. Thus p_2 collapses in every direction in which p_1 collapses.

We show that under the three conditions, hypercube edges collapse to convex hull edges.

Proposition 2. *Let C be a set of hypercube edge collapses, and let $c(p_i)$ be the location of vertex p_i in the collapsed hypercube Given the above conditions on a set of hypercube edge collapses, if (p_1, p_2) is an uncollapsed hypercube edge, then line segment $(c(p_1), c(p_2))$ is an edge of the convex hull of the vertices of the collapsed hypercube.*

Proof. Let $V_C = \{c(p_i)\}$ be the set of vertices of the collapsed hypercube. Let (p_1, p_2) be a hypercube edge which does not collapse. Without loss of generality, assume that p_2 collapses in every direction in which p_1 collapses conforming to Condition 3. If p_1 and p_2 collapse in the same direction, then edge (p_1, p_2) is translated in that direction and identified with another edge of the hypercube. Thus we may assume that p_1 does not collapse in any direction.

Assume that p_2 collapses in exactly k directions to a vertex q. Line segment $(c(p_1), c(p_2)) = (p_1, q)$ is the diagonal of a $k+1$-dimensional hypercube face f. Face f has two k-dimensional subfaces, g_1 and g_2, which are orthogonal to (p_1, p_2). One of these subfaces, say g, contains p_1, while the other contains p_2 and q.

Let W be the set of hypercube vertices except for the vertices of g. By Condition 2, every vertex in g_2 collapses to q. Thus V_C is a subset of $W \cup \{q\}$. The boundary of the convex hull of $W \cup \{q\}$ contains a pyramid whose base is g_1 and whose apex is q. Line segment $(p_1, q) = (c(p_1), c(p_2))$ is an edge of this pyramid and thus of the convex hull of $W \cup \{q\}$. Since V_C is a subset of $W \cup \{q\}$, line segment $(p_1, q) = (c(p_1), c(p_2))$ is an edge of the convex hull of V_C. □

References

[1] Praveen Bhaniramka, Rephael Wenger, and Roger Crawfis. Isosurfacing in higher dimensions. In *Proceedings of Visualization 2000*, pages 267–273, 2000.

[2] Praveen Bhaniramka, Rephael Wenger, and Roger Crawfis. Isosurface construction in any dimension using convex hulls. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):130–141, 2004.

[3] W. Lorensen and H. Cline. Marching cubes: a high resolution 3d surface construction algorithm. *Comput. Graph.*, 21(4):163–170, 1987.

[4] Claudio Montani, Riccardo Scateni, and Roberto Scopigno. A modified look–up table for implicit disambiguation of Marching Cubes. *Visual Comput.*, 10:353–355, 1994.

[5] G.M. Nielson and B. Hamann. The Asymptotic Decider: Removing the ambiguity in Marching Cubes. In *Visualization '91*, pages 83–91, 1991.