

Constructing Material Interfaces from Data Sets with Volume-Fraction Information

*K. Bonnell, M.A. Duchaineau, D.R. Schikore, B. Hamann
and K.I. Joy*

*This article was submitted to
Institute of Electrical and Electronics Engineers Visualization
Salt Lake City, UT
October 8-13, 2000*

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

March 29, 2000

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (423) 576-8401
<http://apollo.osti.gov/bridge/>

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161
<http://www.ntis.gov/>

OR

Lawrence Livermore National Laboratory
Technical Information Department's Digital Library
<http://www.llnl.gov/tid/Library.html>

CONSTRUCTING MATERIAL INTERFACES FROM DATA SETS WITH VOLUME-FRACTION INFORMATION

Kathleen S. Bonnell¹

Mark A. Duchaineau²

Daniel R. Schikore³

Bernd Hamann⁴

Kenneth I. Joy⁴

Abstract

We present a new algorithm for material boundary interface reconstruction from data sets containing volume fractions. We transform the reconstruction problem to a problem that analyzes the dual data set, where each vertex in the dual mesh has an associated barycentric coordinate tuple that represents the fraction of each material present. After constructing the dual tetrahedral mesh from the original mesh, we construct material boundaries by mapping a tetrahedron into barycentric space and calculating the intersections with Voronoi cells in barycentric space. These intersections are mapped back to the original physical space and triangulated to form the boundary surface approximation. This algorithm can be applied to any grid structure and can treat any number of materials per element/vertex.

Keywords: Eulerian flow, material boundary surface, barycentric coordinates, volume fraction, Voronoi diagram.

¹Corresponding Author, Center for Image Processing and Integrated Computing (CIPIC), Department of Computer Science, University of California, Davis, CA 95616-8562, USA; e-mail: bonnell@cs.ucdavis.edu

²Center for Advanced Scientific Computing (CASC), Lawrence Livermore National Laboratory, Livermore, CA 94551, USA; e-mail: duchaine@llnl.gov

³Computational Engineering International, Morrisville, NC 27560, USA; e-mail: schikore@ceintl.com

⁴Center for Image Processing and Integrated Computing (CIPIC), Department of Computer Science, University of California, Davis, CA 95616-8562, USA; e-mail: {hamann, joy}@cs.ucdavis.edu

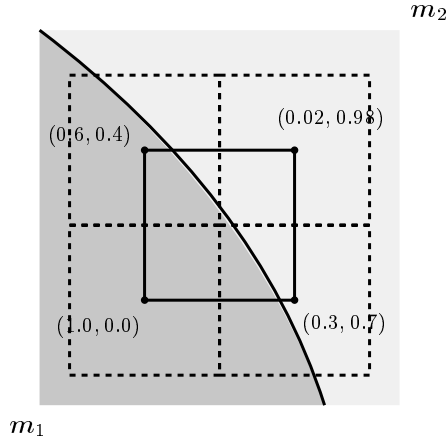


FIGURE 1: Grid and dual grid: The original grid (dashed lines) is replaced by a dual grid (solid lines), obtained by connecting the centers of the original elements. Barycentric coordinates are associated with each vertex of the dual grid. The barycentric coordinates represent the fractions of each material present in the original grid cells.

1. INTRODUCTION

There are numerous instances in which it is necessary to reconstruct or track the boundary surfaces (or “interfaces”) between multiple materials that commonly result from finite element simulations. Multi-fluid Eulerian hydrodynamics calculations require geometric approximations of fluid interfaces to form the equations of motion to advance these interfaces correctly over time. In typical simulations, the grid cells (finite elements) contain fractional volumetric information for each of the materials. Each cell C of a grid \mathcal{S} has an associated tuple $(\alpha_1, \alpha_2, \dots, \alpha_m)$ that represents the portions of each of m materials in the cell, *i.e.*, α_i represents the fractional part of material i . We assume that $\alpha_1 + \alpha_2 + \dots + \alpha_m = 1$. The problem is to find a (crack-free) piecewise two-manifold separating surface approximating the boundary surfaces between the various materials.

To solve this problem, we consider the dual data set constructed from the given data set, see Figure 1. In the dual grid, each cell is represented by a point (typically the center of the cell), and each point is associated with tuple $(\alpha_1, \alpha_2, \dots, \alpha_m)$, where m is the number of materials present in the data set and $\alpha_1 + \alpha_2 + \dots + \alpha_m = 1$. Thus, the boundary surface reconstruction problem reduces to constructing the material interfaces for a grid where each vertex has an associated barycentric coordinate representing the fractional parts of each material at the vertex. We use this “barycentric coordinate field” to approximate the material boundary surfaces.

Important applications of this problem occur for all grid types, *e.g.*, when the data points lie

on a rectilinear grid, curvilinear grid, or an unstructured grid. We therefore develop a solution strategy that is tailored to tetrahedral grids, as all other types of grid structures can be converted to this form, see Nielson [1]. In the case of rectilinear, curvilinear, or even hybrid polyhedral meshes, we preprocess a given grid by subdividing each polyhedral cell into tetrahedra, and apply our algorithm to the resulting tetrahedral grid.

If we have a data set containing m materials, we process each tetrahedral cell of the grid and map our tetrahedral elements into an m simplex representing m -dimensional barycentric space. Next, we calculate intersections with the edges of *Voronoi cells* [2] in the m -simplex. These Voronoi cells represent regions, where one material “dominates” the other materials locally. We map these intersections back to the original space and triangulate the resulting points to obtain the boundary.

Section 2 describes previous work dealing with material boundary surfaces. Section 3 describes the two-material case, which can be viewed as a simple extension of a marching cubes/tetrahedra algorithm [3, 4, 5]. Section 4 describes the three-material case. Here, material boundaries are calculated in barycentric space (a triangle) and mapped back to the original data set. The general m -material case is described in Section 5. In this case, intersections are calculated in a barycentric m -simplex and mapped back to the tetrahedra in the data set. Implementation details are described in Section 6. Section 7 presents results for various data sets, and Section 8 provides conclusions and describes possible future work.

2. RELATED WORK

The bulk of research in material interface reconstruction has been conducted in computational fluid dynamics (CFD) and hydrodynamics, where researchers are concerned with the movement of material boundaries during a simulation.

The Simple Line Interface Calculation (SLIC) algorithm by Noh and Woodward [6] is one of the earliest, describing a method for geometric approximation of fluid interfaces. Their algorithm is used in conjunction with hydrodynamics simulations to track the advection of fluids. Working only with two-dimensional grids, their algorithm produces an interface consisting of line segments, constructed parallel or perpendicular to a coordinate axis. Multi-fluid cells can be handled by grouping fluids together, calculating the interface between the groups, subdividing the groups, and iterating this process until interfaces are constructed. Since this algorithm only uses line segments that are parallel to the coordinate axes, the resulting interfaces are generally discontinuous.

The algorithm of Youngs [7] also operates on two-dimensional grids and uses line segments to approximate interfaces. In this algorithm, the line segments are not necessarily perpendicular or parallel to a coordinate axis. Instead, the neighbor cells of a cell C are used to determine the slope of a line segment approximating an interface in C . The exact location of the line segment is adjusted to preserve the volume fractions in a cell. Multiple materials are treated by grouping materials and determining interfaces on a two-material basis. Again, the interfaces produced by this method are generally discontinuous.

The algorithm of Gueffier [8] requires an estimate of the normal vector to the interface in order to reconstruct the interface. He utilizes differencing or least-squares methods to approximate this normal and adjusts a line segment perpendicular to this normal to generate a boundary surface in a cell. The surface is generally discontinuous, and it is unclear how multiple materials can be handled, or how one could generalize the algorithm to the case of three-dimensional grids.

Pilliod and Puckett [9] compare various volume-of-fluid interface reconstruction algorithms, including SLIC, noting differences in the surfaces reconstructed and demonstrating first-order or second-order accuracy. Their goal is to reproduce a linear interface.

Nielson and Franke [10] have presented a method for calculating a separating surface in an unstructured grid where each vertex of the grid is associated with one of several possible classes. Their method generalizes the marching cubes/tetrahedra algorithm, but instead of using a strict binary classification of vertices, it allows any number of classes. Edges in tetrahedral grids whose endpoints have different classifications are intersected by the separating surface. Similarly, the faces of a tetrahedron whose three vertices are classified differently, are assumed to be intersected by the surface in the middle of the face. When all four vertices of a tetrahedron have different classifications, the boundary surface intersects in the interior of the tetrahedron. The resulting “mid-edge”, “mid-face” and “mid-tetrahedron” intersections are triangulated to form the surface.

Our algorithm generalizes the above schemes. We utilize a grid that has a barycentric coordinate associated with each vertex. This allows us to generate material boundaries directly from the intersections calculated in “barycentric space.” Our algorithm handles multiple materials and can reconstruct layers and “Y-type” (non-manifold) interfaces with equal ease. Our algorithm does not rely on application-specific knowledge of hydrodynamics or other simulation codes, but solves the problem from a purely mathematical viewpoint.

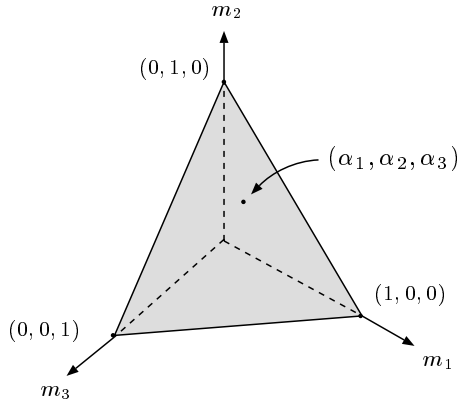


FIGURE 2: Barycentric triangle.

3. THE TWO-MATERIAL CASE

Consider a grid \mathcal{S} , where each vertex of \mathcal{S} has an associated barycentric coordinate α . In the case of two materials, α is the two-tuple $\alpha = (\alpha_1, \alpha_2)$ where $\alpha_1 + \alpha_2 = 1$. In this case, we define the material boundary to be the set of points where $\alpha_1 = \alpha_2 = \frac{1}{2}$.¹ We only need to find the point on each edge where $\alpha_1 = \alpha_2 = \frac{1}{2}$, using linear interpolation. By computing these points on all edges of a cell, we can use a marching tetrahedra implementation [5] to draw the boundary surface in a cell when only two materials are present. (If only one material is present in a cell, no boundary surface exists).

The two-material case reduces to an isosurface calculation, determining the isosurface $\alpha_1 = \frac{1}{2}$, which can easily be implemented by a marching tetrahedra method.

4. THE THREE-MATERIAL CASE

In the three-material case, each vertex has an associated 3-tuple $\alpha = (\alpha_1, \alpha_2, \alpha_3)$, where $\alpha_1 + \alpha_2 + \alpha_3 = 1$. Here, α_1 is the fraction of material m_1 , α_2 is the fraction of m_2 , and α_3 is the fraction of m_3 , respectively. The coordinate $(\alpha_1, \alpha_2, \alpha_3)$ lies on the equilateral triangle with vertices $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$, as shown in Figure 2. We partition this triangle into three regions, defined by the *Voronoi cells* \mathcal{V}_1 , \mathcal{V}_2 , and \mathcal{V}_3 , see Figure 3. The Voronoi cells \mathcal{V}_j are bounded by the edges of the triangle, and the three line segments l_{12} , l_{13} , and l_{23} , where

¹The rationale behind this decision is that an infinitesimally small cell whose center is on the boundary will contain approximately half of each material.

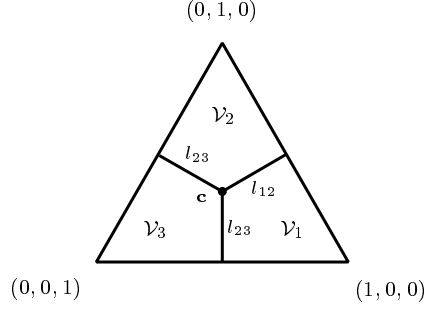


FIGURE 3: Partitioning the barycentric triangle into regions. The point c is the point $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, the center of the triangle. l_{12} , l_{13} , and l_{23} are the line segments bounding the Voronoi cells \mathcal{V}_j in the interior of the triangle.

$\alpha_1 = \alpha_2$ and $\alpha_3 \leq \frac{1}{3}$, $\alpha_1 = \alpha_3$ and $\alpha_2 \leq \frac{1}{3}$, or $\alpha_2 = \alpha_3$ and $\alpha_1 \leq \frac{1}{3}$, respectively.

For two-dimensional triangular grids, we map the associated barycentric coordinates of a triangle T onto a triangle T' in barycentric space. We use the intersections of the edges of T' with the edges of the Voronoi cells in the barycentric triangle to define material interfaces in T' . These intersections are then mapped back to points in T , using the same linear parameters to determine the intersections on the edges of T . There are three cases:

- The triangle T' does not intersect l_{12} , l_{13} , or l_{23} . In this case, we assume that no material boundary exists in T .
- The triangle T' intersects at least one of the line segments l_{12} , l_{13} , or l_{23} , and the center c of the barycentric triangle does not lie inside T' . In this case, we calculate intersections on the edges of T , corresponding to the intersections of T' with l_{12} , l_{13} , and l_{23} , respectively. (The triangle T' may intersect at most two of these lines.) The material boundary line segments inside T are then defined as the line segments that connect the corresponding edge intersections in T . Figures 4a and 4c illustrate these cases.
- The point c lies inside T' . In this case, we calculate three edge intersections for T , corresponding to the intersections of T' with l_{12} , l_{13} , and l_{23} , respectively, and a point in the interior of T , corresponding to the point c in T' . The material boundary line segments are defined as the three lines connecting the edge intersections and the face point. Figure 4b illustrates this case.

If one of the α_i values is zero for each of the three vertices of a triangle, then all points map to an edge of the barycentric triangle. Thus, the situation reduces to the two-material case. If only one material is present at all three vertices, then no intersections are calculated.

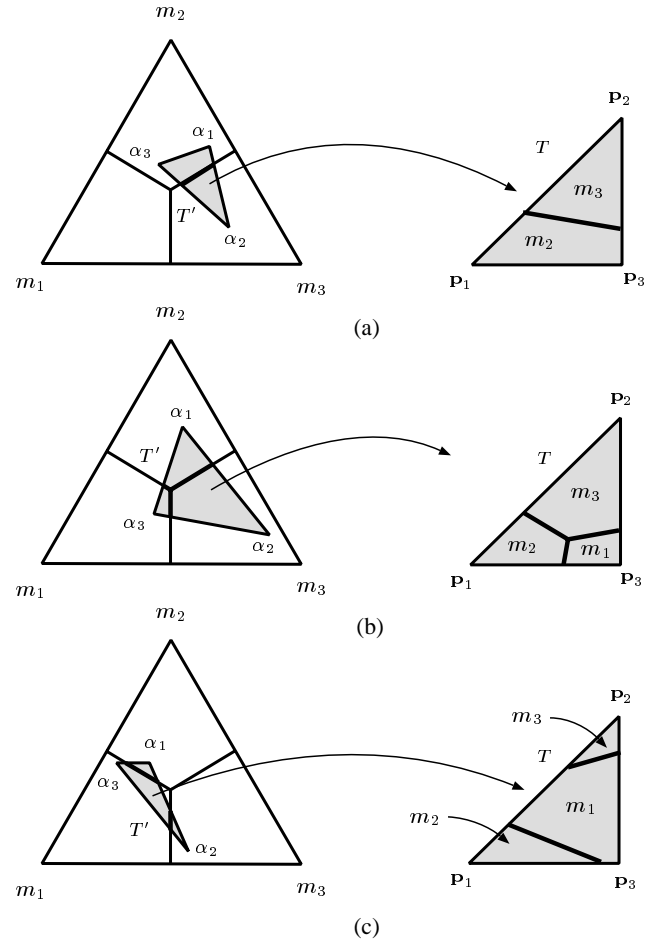


FIGURE 4: Mapping intersections from barycentric space to the triangle T : The images on the left show the triangle T' in barycentric space, and the images on the right show the material boundary line segments mapped from barycentric space to the original triangle T in physical space.

For three-dimensional tetrahedral grids, we use the associated barycentric values of the vertices of each face of a tetrahedron T and map the tetrahedron to an image T' of T in barycentric space. Intersections are calculated separately for each face of T' and mapped back to T . There are three cases:

- No edge of the tetrahedron T' intersects the line segments l_{12} , l_{13} , or l_{23} . In this case, no material boundaries exist in the tetrahedron T .
- The edges of the tetrahedron T' intersect at least one of the line segments l_{12} , l_{13} , or l_{23} , but the point $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, the center of the barycentric triangle, does not lie inside any of the faces of T' . In this case, we calculate the intersection line segments for each triangular face of T and determine a triangulation from these segments by following the marching tetrahedra algorithm [5]. Figures 5a and 5b illustrate the possible cases.
- The center point of the barycentric triangle lies inside two faces of T' . In this case, two faces have a single material boundary line segment connecting two edge intersection points, and two faces have three material boundary line segments meeting in the interior of two faces. We map the intersections back to the tetrahedron T , using linear interpolation. Using the material boundary line segments for each face, and the line segment connecting the two points in the interior of two faces of T , we can determine a valid triangulation of the boundary surface. Figure 5c illustrates this case.

5. THE GENERAL CASE

In the case of four materials, each vertex has an associated barycentric coordinate given by the four-tuple $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$, where $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$. By considering the tetrahedron having vertices $(1, 0, 0, 0)$, $(0, 1, 0, 0)$, $(0, 0, 1, 0)$, and $(0, 0, 0, 1)$ in four-dimensional space, we can construct a partition of this tetrahedron similarly to the three-material case. Again, we use the Voronoi cells for the decomposition of the barycentric tetrahedron. The boundaries of these cells include parts of the faces of the tetrahedron and six planar pieces, which are defined by $\alpha_1 = \alpha_2$, $\alpha_1 = \alpha_3$, $\alpha_1 = \alpha_4$, $\alpha_2 = \alpha_3$, $\alpha_2 = \alpha_4$, and $\alpha_3 = \alpha_4$. This Voronoi partition is shown in Figure 6a.

For two-dimensional grids, we map the four-dimensional barycentric coordinates associated with the vertices of a triangle T into a triangle T' in barycentric space. We use a clipping algorithm to generate the intersections in the triangle T' , clipping against the six planes defining the boundaries of the Voronoi cells of the barycentric tetrahedron. The tetrahedron is

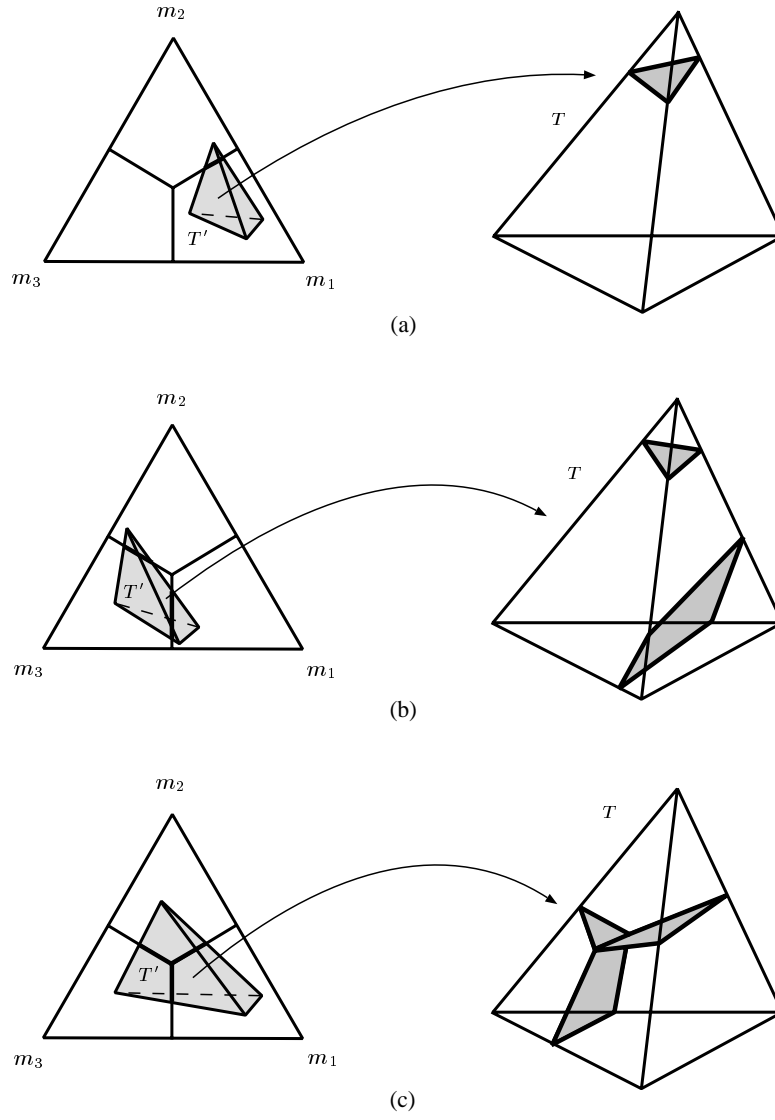


FIGURE 5: Examples of material boundary surface determination for tetrahedral grids.

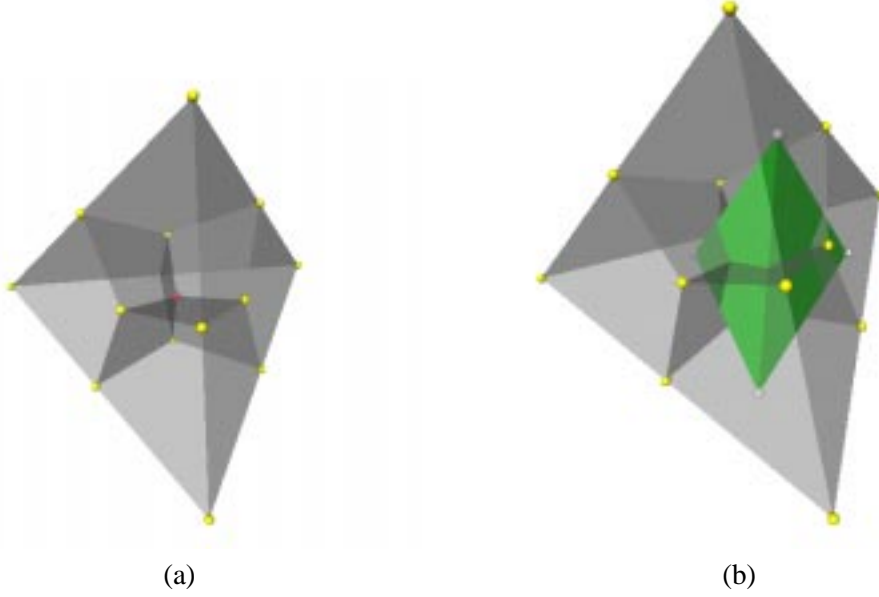


FIGURE 6: Voronoi cell decomposition in the four-material case: The figure illustrates a three-dimensional projection of the barycentric tetrahedron from four-dimensional space. The tetrahedron is segmented into four Voronoi cells in (a). In (b), a tetrahedron, mapped from physical space, is shown inside the barycentric tetrahedron.

stored in a binary space partitioning (BSP) tree, and we apply the clipping algorithm described by Samet [11]. Once the intersections are determined by the clipping algorithm, the material boundary line segments can be determined for the triangle T .

For three-dimensional tetrahedral grids, we use a similar clipping algorithm for the image T' of a tetrahedron T . This enables us to calculate the boundary surfaces inside the tetrahedron T' , which we then map back to the tetrahedron T in physical space.

In the general case of m materials, we map a tetrahedron T to a tetrahedron T' in an m -simplex in barycentric space. The m -simplex is partitioned into Voronoi cells whose boundaries consist of the faces of the m -simplex and the $\binom{m}{2}$ hyperplanes defined by $\alpha_i = \alpha_j$, where $i \neq j$ and $1 \leq i, j \leq m$. The material boundaries for T' are calculated by using a clipping algorithm and then are mapped back to physical space to form the material boundaries inside T . We utilize a BSP algorithm to perform the clipping.

6. DISCUSSION

The algorithm runs in effectively the same time as does the marching cubes/tetrahedra algorithm. We traverse the cells of a grid and calculate, for each cell, a polygonal representation of the material boundaries. Most grid cells in common examples contain only one material, and boundaries do not exist in these cells.

We note that the algorithm can miss material boundaries in tetrahedra. In any isosurface-type algorithm, it is possible for the isosurface to enter a tetrahedron, but only intersect one edge. In this case, the algorithm cannot detect the material boundary from only the information at the vertices.

In the three-material case, we have chosen the point $\mathbf{c} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ as the “center” of the barycentric triangle. This assumes that there are three distinct sectors in the barycentric triangle, subdividing the triangle in a “Y” fashion, and that a cell of infinitesimally small size contains about one-third of each material in the cell. This is not always the case. For example, consider a “T intersection,” where any small cell would contain one-half of one material and one-quarter of the other two materials. We can adjust our segmentation of the barycentric triangle so that the point \mathbf{c} is at an arbitrary location in the triangle, and the edges that determine the intersections can be adjusted appropriately. This can be done by sampling in a larger neighborhood of a specific cell to understand how to weigh the materials about the “Y point.”

In the four-material case, the center of the tetrahedron can also be adjusted. However, this implies that the “center” vertices on the faces must also be adjusted so that the separating surfaces remain planar. In the m -material case, similar considerations also hold when adjusting the center of the m -simplex.

Our algorithm can be considered as a direct generalization of Nielson’s and Franke’s algorithm [10]. Each vertex of a grid \mathcal{S} has an associated barycentric coordinate $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)$, and by restricting material fractions such that exactly one $\alpha_i = 1$, we obtain the case where each vertex is only associated with one material. In this case, our algorithm produces the results produced by the Nielson-Franke algorithm.

7. RESULTS

We have implemented this algorithm and used it to generate material interfaces for a variety of data sets. Figure 7 illustrates the material interfaces for a data set consisting of three materials.



FIGURE 7: The boundary surfaces of two materials formed as concentric spherical layers.

The boundary of the region containing material 1 has a spherical shape, and the other two material regions are formed as concentric layers around material 1 – forming two material interfaces. The original grid is rectilinear-hexahedral consisting of $64 \times 64 \times 64$ cells. We constructed the dual grid, and then split each dual cell into six tetrahedra, see Nielson [1], creating 1,572,864 tetrahedra. Approximately 30% of the tetrahedra containing the material boundaries contain two boundary surfaces.

Figure 8 shows the material interfaces for a three-material data set of a simulation of a ball striking a plate consisting of two materials. The original data set rectilinear-hexahedral and has a resolution of $53 \times 23 \times 23$ cells. Again, we created the dual grid, and split each dual cell into six tetrahedra, creating 28,037 tetrahedra. Four instances of the data set are shown.

Figure 9 illustrates the material interfaces for a human brain data set. The original grid is rectilinear-hexahedral containing $256 \times 256 \times 124$ cells. Each cell contains a probability tuple giving the probability that each material is present at the point. The resulting dual data set contains over eight million tetrahedra.

8. CONCLUSIONS

We have presented a new algorithm for material boundary surface reconstruction from data sets containing material volume-fraction information. We transform a given grid to a dual grid, where each vertex has an associated barycentric coordinate that represents the fractions

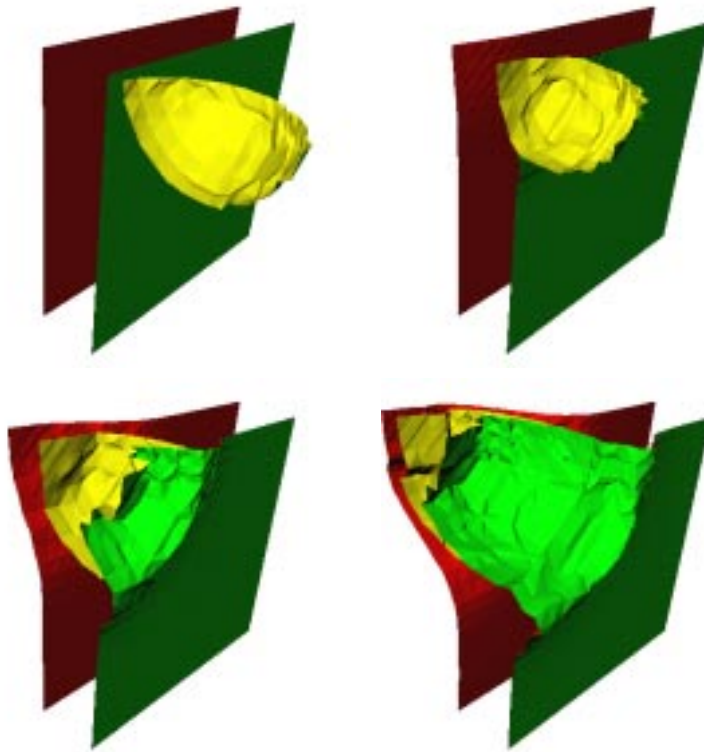


FIGURE 8: Simulation of a ball striking a plate consisting of two materials. The picture in the upper-left corner shows the initial configuration, and the following sequence of pictures shows the boundary surfaces as the ball penetrates the two-material plate.

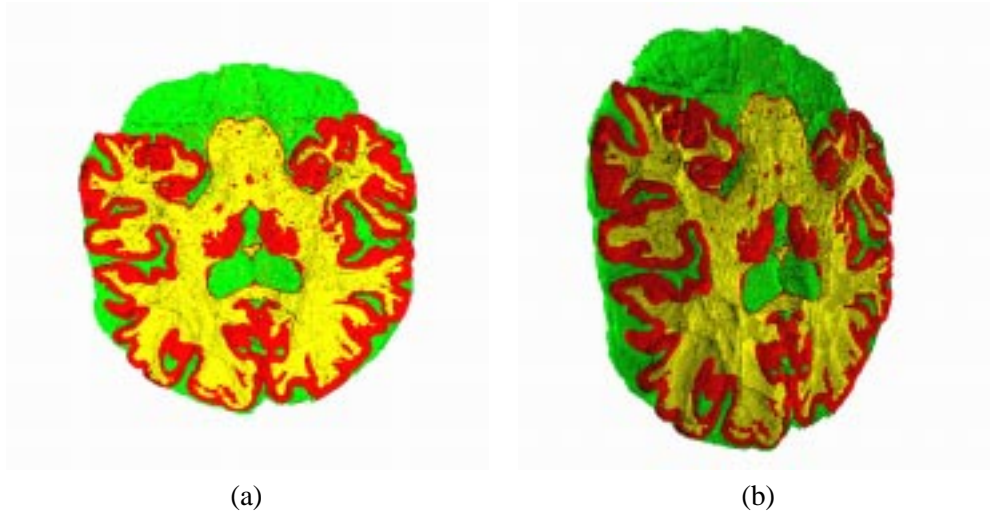


FIGURE 9: Illustrations of a brain data set. The material boundary surfaces are shown in red, green and yellow. The polygons forming the material boundaries were clipped to show the interior of the data set. Two views of the material boundary surfaces, rotated differently, are shown in (a) and (b).

of each material present. After tetrahedrizing the dual grid, we construct the material interfaces by mapping each tetrahedron to barycentric space, calculating the intersections with Voronoi cells in barycentric space. These intersection points are mapped back to physical space and triangulated to form the resulting boundary surface.

The algorithm can treat any number of materials per cell, and since it is based on tetrahedral grids, it can be used with any grid structure.

Concerning future work, we would like to insert a “measure-and-adjust” feature to the algorithm. Once an initial boundary surface approximation is calculated, we calculate (new) volume fractions for cells directly from this boundary surface. This will enable us to calculate the difference between the original volume fractions and the volume fractions as implied by our initial boundary surface approximation. It is then possible to adjust our material interfaces to minimize the volume fraction deviations.

We also plan to extend this algorithm to multidimensional grids.

9. ACKNOWLEDGMENTS

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract no. W-7405-Eng-48. It was also supported

by the National Science Foundation under contracts ACI 9624034 and ACI 9983641 (CA-REER Awards), through the Large Scientific and Software Data Set Visualization (LSSDSV) program under contract ACI 9982251, and through the National Partnership for Advanced Computational Infrastructure (NPACI); the Office of Naval Research under contract N00014-97-1-0222; the Army Research Office under contract ARO 36598-MA-RIP; the NASA Ames Research Center through an NRA award under contract NAG2-1216; the Lawrence Livermore National Laboratory under ASCI ASAP Level-2 Memorandum Agreement B347878 and under Memorandum Agreement B503159; and the North Atlantic Treaty Organization (NATO) under contract CRG.971628 awarded to the University of California, Davis. We also acknowledge the support of ALSTOM Schilling Robotics, Chevron, General Atomics, Silicon Graphics, and ST Microelectronics, Inc. We would like to thank Randy Frank of the Center for Applied Scientific Computing at Lawrence Livermore National Laboratory for supplying several of the data sets presented in this paper. We also thank the members of the Visualization Group at the Center for Image Processing and Integrated Computing (CIPIC) at the University of California, Davis for their support.

References

- [1] G. M. Nielson, “Tools for triangulations and tetrahedrizations and constructing functions defined over them,” in *Scientific Visualization: Overviews, Methodologies, and Techniques* (G. M. Nielson, H. Hagen, and H. Müller, eds.), (Los Alamitos), pp. 429–525, IEEE Computer Society Press, 1997.
- [2] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tesselations — Concepts and Applications of Voronoi Diagrams*. Chichester: Wiley, 1992.
- [3] W. E. Lorensen and H. E. Cline, “Marching cubes: a high resolution 3D surface construction algorithm,” in *Computer Graphics (SIGGRAPH '87 Proceedings)* (M. C. Stone, ed.), vol. 21, pp. 163–170, July 1987.
- [4] G. M. Nielson and B. Hamann, “The asymptotic decider: Removing the ambiguity in marching cubes,” in *Proceedings of IEEE Visualization '91*, pp. 83–91, 1991.
- [5] Y. Zhou, B. Chen, and A. Kaufman, “Multiresolution tetrahedral framework for visualizing regular volume data,” in *IEEE Visualization '97* (R. Yagel and H. Hagen, eds.), pp. 135–142, IEEE, Nov. 1997.
- [6] W. F. Noh and P. Woodward, “SLIC (Simple line interface calculation),” in *Lecture Notes in Physics* (A. I. van der Vooren and P. J. Zandbergen, eds.), pp. 330–340, Springer-Verlag, 1976.
- [7] D. L. Youngs, “Time-dependent multi-material flow with large fluid distortion,” in *Numerical Methods for Fluid Dynamics* (K. W. Morton and J. J. Baines, eds.), pp. 273–285, Academic Press, 1982.
- [8] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski, “Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows,” *Journal of Computational Physics*, vol. 152, pp. 423–456, 1999.
- [9] J. E. Pilliod and E. G. Puckett, “Second-order accurate volume-of-fluid algorithms for tracking material interfaces,” technical report, Lawrence Berkeley National Laboratory, 2000.
- [10] G. M. Nielson and R. Franke, “Computing the separating surface for segmented data,” in *Proceedings of IEEE Visualization '97* (R. Yagel and H. Hagen, eds.), (Los Alamitos), pp. 229–234, IEEE Computer Society Press, Oct. 1997.

- [11] H. Samet, *The Design and Analysis of Spatial Data Structures*. Series in Computer Science, Reading, Massachusetts, U.S.A.: Addison-Wesley, reprinted with corrections ed., Apr. 1990.