Multiple Transparent Material-enriched Isosurfaces

Ravindra L. Kanodia* Lars Linsen*† Bernd Hamann*

* Institute for Data Analysis and Visualization (IDAV)
Department of Computer Science
University of California, Davis
Davis, CA 95616, U.S.A.

† Department of Mathematics and Computer Science
Ernst-Moritz-Arndt-Universität Greifswald
Greifswald, Germany

ABSTRACT

Isosurface extraction is a standard method for visualizing scalar volume data that can be used to render a specific material boundaries inherent in multi-material data sets. Multiple transparent isosurfaces can thus be used to visualize multiple material boundaries, but still fail to capture any data in between the boundary layers. We describe how isosurfaces can be "enriched" with surrounding material information. By visualizing surrounding material, both material boundary information and gradient - or change in density - information of the scalar field are represented. Visualizing multiple transparent material-enriched isosurfaces leads to a fairly effective volumetric impression. Thus, our approach approximates results obtained from direct volume rendering.

The visualization of multiple transparent isosurfaces requires a back-to-front rendering of the typically triangulated surface components. The order of the surfaces' triangles is imposed by the location of the convex cells they are extracted from, which supports fast rendering of multiple isosurface.

Keywords

Multiple Transparent Isosurfaces, Volume Rendering, Material-enriched Isosurfaces.

1 Introduction

Isosurface extraction and direct volume rendering are the most common techniques used for visualizing scalar-valued volume data. The volumetric data sets typically result from numerical simulations or from 3D scanning and imaging processes. The computed or measured scalar fields represent material properties such as pressure, temperature, density, etc.

Isosurface extraction methods explicitly compute the three-dimensional geometry of material boundaries in the form of two-manifold surfaces. Once an isosurface has been extracted, surface rendering is efficient, especially when exploiting acceleration mechanisms using graphics hardware. Moreover, the rendered surface is of high quality, as a sharp material boundary is computed and photo-realistic shading algorithms can be applied. On the other hand, one isosurface represents only one material boundary and fails to capture most of the volumetric information.

Direct volume rendering methods provide a "richer" visualization, as the whole range of material information is incorporated into the rendered image. Transfer functions are used to control color and opacity for the depiction of the various materials present in the volume data. Direct volume rendering results provide a strong sense of the overall content of a data set. Unfor-

*ravi@cyberman.com, hamann@cs.ucdavis.edu
†linsen@uni-greifswald.de

Conference proceedings ISBN 80-903100-7-9
WSCG 2005, January 31-february 4, 2005
Plzen, Czech Republic.

tunately, the computational costs are high. Hardware-accelerated approaches exist and are commonly used, but are limited in their applicability to large data sets.

Multiple transparent isosurface rendering can be used to overcome the single isosurface's drawback by adding more information to rendered images, while maintaining the advantages in speed and quality. In Section 3, we describe how a standard isosurface extraction algorithm can be extended to extract multiple isosurfaces into a data structure that allows for fast and correct rendering. We exploit the properties of marching isosurface-extraction algorithms to compute the occlusion properties of the multiple isosurfaces "on-the-fly", i. e., without applying an expensive post-processing step for polygon sorting.

Unfortunately, visualization of multiple transparent isosurfaces do not quite establish the same-quality volumetric impression of the visualized data set as direct volume rendering techniques do. Multiple transparent isosurface rendering can only approximate direct volume-rendered images when using "spiky" transfer functions, i. e., transfer functions with vanishing opacity everywhere except for a finite number of small separated intervals. The spikes or peaks in the opacity function correspond to individual isosurfaces. The individual isosurfaces are colored with respect to the color information of the respective material, which can be looked up in the transfer function. In between opacity peaks, however, there are gaps, which represent material that cannot be captured using multiple isosurfaces.

We present an approach to "fill" these gaps to a certain extent. We enrich the multiple transparent isosurfaces with information about the material in-between the isosurfaces. In Section 4, we describe how multiple material-enriched isosurfaces can be used to approximate volume rendering. We present and discuss our results in Section 5.

2 Related Work

Isosurfaces are commonly extracted from volumetric scalar fields using marching-cell algorithms. These algorithms go back to the marching-cubes approach [LC87], which operates on uniform rectilinear grids. The algorithm "marches" through the grid considering each cell of the grid once. Intersections of the isosurface with the edges of each cell are computed using linear interpolation of the values at the cell's corners. The intersection points are connected forming a triangular surface mesh. Many extensions and improvements have been made to the original approach including the solution of ambiguous cases [Ham91], better triangulations [Nie03], and a generic algorithm combining previous approaches [BL03]. When splitting the hexahedral cells into tetrahedral ones, ambiguous cases are resolved (while making certain assumptions though), and isosurfaces can be extracted using a marching-tetrahedra algorithm [GH95].

Multiple isosurfaces are extracted by the method of Wan et al. [WTTL96]. Gerstner [Ger02] developed a multiresolution approach for fast multiple isosurface extraction from adaptively refined tetrahedral grids. For a fast rendering of the multiple isosurfaces, Gerstner deployed a binary tree-based sorting method. The implementation of our multiple isosurface extraction algorithm is based on a marching-tetrahedra approach, but any other isosurface extraction approach that marches through a grid of convex polyhedra could be used instead.

Gerstner's method [Ger02] renders different isosurfaces with different colors and opacities, which leads to results similar to those one can obtain using direct volume rendering with spiky transfer functions. Information about the scalar field in between the isosurfaces is not captured. In particular, the transformation from one material, whose boundary is represented by one isosurface, to another material, whose boundary is represented by another isosurface, is not visualized, as it would be when using direct volume rendering with smoother transfer functions.

Direct volume rendering goes back to the work by Levoy [Lev88], who introduced the concept of volume ray casting for uniform rectilinear grids. Data values within a cell are computed using trilinear interpolation. Ray casting in an improved form is still a widely used approach for direct volume visualization, as it produces high-quality results. Another direct volume rendering approach with high-quality results is splatting [Wes90], where the trilinear interpolation is replaced by a Gaussian function. The shear-warp approach [LL94] is targeted toward higher speed at the expense of lower-quality results. Recent developments in graphics hardware made the 3D-texture-mapping approach favorable in terms of speed and, with floating-point precision, even in terms of quality. The uniform rectilinear grid is mapped to the 3D texture memory and trilinear interpolation is performed by the hardware components [CCF94]. One obvious limitation is the restriction to the size of the 3D texture memory.

When comparing isosurface extraction with direct volume rendering, isosurface rendering is still faster and/or produces higher-quality images in terms of smoothness and illumination. The advantage in speed is given by the reduction of the volume data set to a two-manifold surface, whose triangular mesh repre-

sentation allows for fast rendering. Moreover, many fast and realistic lighting techniques for triangular mesh rendering exist, which allow for smooth shading without rendering artifacts. Also, the explicit computation of material boundaries facilitates quantitative analyses, which is particularly important for biomedical applications such as clinical measurements. On the other hand, the reduction of a volumetric model to a surface model obviously can lead to significant loss of information. Direct volume rendering incorporates all of the volume data and shows overlaying and internal features in a realistic fashion. We enrich multiple transparent isosurfaces with surrounding material information, such that the volumetric impression of a direct volume rendering is approximated while maintaining the advantages of isosurface rendering with respect to speed and quality.

3 Fast Multiple Isosurface Rendering

Let $f : D \to R$ be a trivariate scalar function with domain $D \subset \mathbf{R}^3$ and range $R \subset \mathbf{R}$. The values $f(\mathbf{x}) \in R$ of function f are known at discrete sample points $\mathbf{x} = (x,y,z) \in D$. Let the sample points be organized in a three-dimensional grid with convex grid cells. Moreover, let $v_0, \ldots, v_{n-1} \in R$ be an arbitrary number n of isovalues. The multiple isosurfaces are defined by $f(\mathbf{y}) = v_i$ for $i = 0, \ldots, n-1$ and $\mathbf{y} \in D$. Let F_i be the isosurface with respect to isovalue v_i, $i = 0, \ldots, n-1$.

A marching isosurface extraction algorithm determines an isosurface F_i by iterating through the three-dimensional grid once, i. e., each cell is considered once, and determining an isosurface component within each cell independently, if existent.

Typically, uniform rectilinear or tetrahedral grids are used for discrete data representation and a linear interpolation model for the determination of the isosurface components within each cell. The approach presented in this paper does not depend on the grid structure and the interpolation method used. It only requires the cells of the grid to be convex, which can easily be achieved in the unlikely case that they are not. For implementation purposes, we used uniform tetrahedral grids and linear interpolation within each tetrahedron, following the ideas in [GH95].

Multiple isosurfaces can be extracted just as single ones are, i. e., by iterating through the grid once and extracting for each cell all existent isosurface components within the cell for all isosurfaces F_i, $i = 0, \ldots, n-1$. Thus, the algorithm is still linear in the number of cells. Each cell stores up to n isosurface components. When using a linear interpolation method, the isosurface components are represented using a polygonal surface model.

When rendering multiple transparent isosurfaces, the isosurface components must be sorted in a back-to-front or front-to-back order according to depth from the view-point or viewing plane. We employ a fast yet simple back-to-front sorting method. It essentially exploits a loophole for depth-sorting: The isosurface components in a scene do not have to be truly sorted according to depth from viewing plane, as long as the rendering algorithm guarantees that no isosurface component is drawn after another isosurface component which occludes it.

Under the assumption that all grid cells are convex and non-overlapping polyhedra, we can uniquely determine whether one cell is in front of another cell with respect to a given viewing vector, or the other way round. Each isosurface component lying entirely in the back cell must be rendered before any isosurface component lying entirely in the front cell. Moreover, the "in-front-of" relation defines a partial order, as it fulfills the property of being transitive. Thus, the isosurface components can be sorted by sorting the cells.

The sorting of the cells can be done hierarchically by grouping neighbored cells to form larger convex and non-overlapping cells. For uniform rectilinear or uniform tetrahedral grids, cells can be grouped to rows of cells, and rows can further be grouped to slabs. Obviously, rows and slabs are, again, convex and non-overlapping.

Exploiting this three-step hierarchy, cells can be sorted in constant time. We only have to determine for each of the three axes of the grid's coordinate system whether high or low values are closer to the viewing plane, i. e., we have to determine the orientation of the grid with respect to the viewing direction. The sorting of the cells is implicitly given by grid order.

It remains to sort the isosurface components within each cell. If isosurface components are represented using a polygonal surface model, the number of triangles per cell is bounded by the maximum number t of triangles per cell generated by the single isosurface extraction algorithm. Therefore, for multiple isosurface extraction, the total number of triangles per cell is bounded by $n \cdot t$. As this number is small, we employ a standard depth-sort algorithm to sort the triangles within each cell, which does not slow down the performance of our multiple isosurface rendering algorithm noticeably.

4 Material-enriched Isosurfaces

Rendering multiple isosurfaces obviously exhibits additional information compared to a visualization using a single isosurface. However, this additional information partially gets lost when displaying the surfaces using same color and opacity. Since for performance reasons multiple isosurfaces are extracted simultaneously, they are also stored together and cannot be easily separated afterwards. Therefore, already during the extraction phase, we have to tag each isosurface component and even each triangle. The tag can be an identifier of the material whose boundary is represented by the isosurface. Multiple isosurfaces are rendered by assigning color and opacity to each material [Ger02].

By performing this assignment step, a transfer function, as known from direct volume rendering, is approximated. However, this use of multiple transparent isosurfaces to approximate direct volume rendering is rather limited. When using direct volume rendering, the transfer function allows for color and opacity assignments for every value $v \in R$. When using multiple isosurface rendering, color and opacity for only n values $v_0, \ldots, v_{n-1} \in R$ are assigned. Any other material with value $v \neq v_i$, $i = 0, \ldots, n-1$, is not visualized.

To add more material information to a visualization, we introduce the concept of material-enhanced isosurfaces. During isosurface extraction, we first compute the surface normal $\mathbf{n}$ at point $\mathbf{p}$ as the normalized gradient, which needs to be computed anyway if smooth shading algorithms are applied. We determine and store the material value v_{near} found at a short distance λ from point $\mathbf{p}$ along the normal direction $\mathbf{n}$, i. e.,

$$v_{near} = f(\mathbf{p} + \lambda \mathbf{n}) \; .$$

During rendering, this additional material information can be used to color the isosurface. The color associated with material value v_{near} is obtained from the used transfer function. Also, the distance λ can be determined from the transfer function: The opacity function has peaks at isovalues $v_0, \ldots, v_{n-1}$; the wider these peaks are, the greater is distance λ.

We have implemented two versions to color material-enriched isosurfaces. Let F_i be the isosurface to be rendered. The first option is to render the isosurface with the color of material value v_{near}. The second option is to render the isosurface by blending the color of material value v_{near} with the color of isovalue v_i. The former leads to a stronger volumetric impression, as more emphasis is placed on the material around each isosurface, while the latter leads to a more realistic impression, as the color is closer to the color assigned to the isovalue.

5 Results and Discussion

In Figures 1 and 2, we compare visualizations using material-enriched isosurfaces with standard isosurface visualizations in the context of single isosurface rendering. The isosurfaces are extracted using an extended marching-tetrahedra approach. The colors used in the renderings are obtained from a user-defined transfer function.

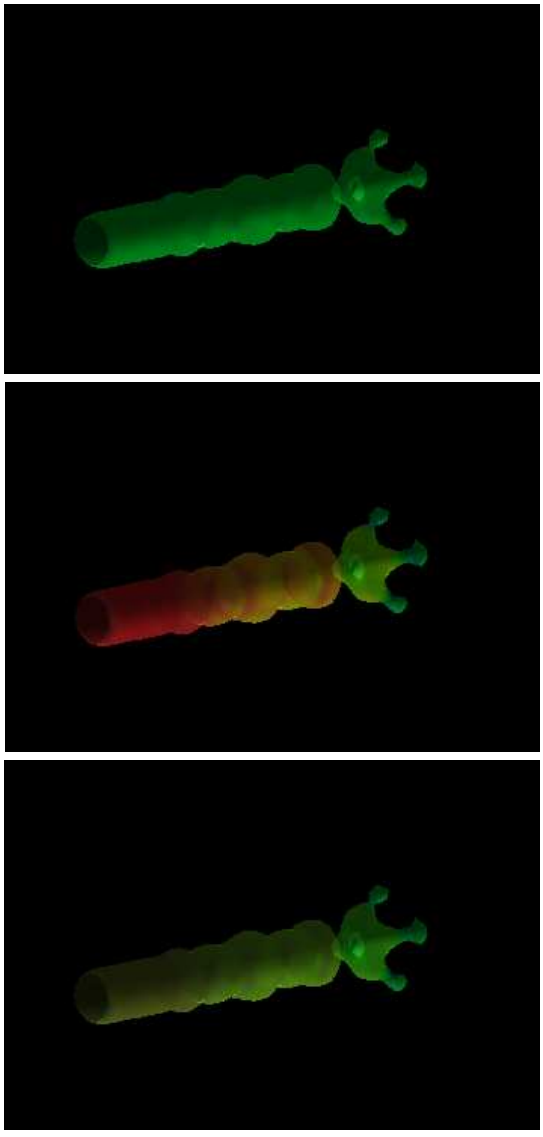

Figure 1: Single isosurface rendering applied to "fuel injection" data set: (a) standard isosurface; (b) material-enriched isosurface using color of material v_{near}; (c) material-enriched isosurface using color blending.

Figure 1 shows a simulation data set of fuel injected into a combustion chamber. The higher the density

value is, the less air is present.[1] Figure 1(a) shows a standard isosurface, while Figures 1(b) and 1(c) show the rendering of material-enriched isosurfaces. In Figure 1(b), the assigned color is defined by a transfer-function look-up table for material with value v_{near}, i. e., material near the surface. In Figure 1(c), the assigned color is a blending of the colors used in Figures 1(a) and 1(b).

The visualizations with material-enriched isosurfaces exhibit more information in the data set. A single material-enriched isosurface suffices to understand that the density field has the steepest gradient perpendicular to the extracted isosurface where the fuel is injected (left side in the figures), which decreases smoothly with increasing distance.

In Figure 2, we provide another example for single transparent material-enriched isosurface rendering. The data set represents a simulation of a two-body probability distribution of a nucleon in the atomic nucleus 16O, where the position of a second nucleon is known.[2] The material-enriched isosurface renderings in Figures 2(b) and 2(c) exhibit a high-density region (reddish color) - an information that cannot be perceived from the standard isosurface visualization in Figure 2(a).

In Figure 3, we show multiple transparent isosurfaces extracted from the "fuel injection" data set. In Figure 3(a), the multiple isosurfaces are rendered with the same color and opacity. It is difficult to distinguish the different layers of material boundary. In Figure 3(b), the visualization of the different isosurfaces uses colors and opacities obtained from an assigned transfer function. This visualization leads to results similar to the ones shown in [Ger02]. The different material layers are easy to identify, but no volumetric impression as known from direct volume rendering techniques can be achieved. In Figure 3(c), we use material-enriched isosurfaces, where color is assigned with respect to near material values. A strong volumetric impression is achieved. The visualization with multiple transparent material-enriched isosurfaces can indeed approximate a visualization using direct volume rendering. Figure 3(d) shows a visualization with material-enriched isosurfaces, where the color of the "iso-material" is blended with the color of the near material. The volumetric impression is not as strong as in Figure 3(c), but the visualization is more realistic, as the colors are close to the colors in Figure 3(b). For comparison, we also show a direct volume-rendered image in Figure 3(e).[3]

When comparing the multiple transparent material-enriched isosurface renderings with the direct volume rendering, we observe that the surfaces can approximate the direct volume rendering in terms of volumetric impression, but we also recognize differences in the resulting images. When using direct volume rendering, material boundaries are not as clearly visible. For example, the outer-most material boundary visible in the renderings in Figures 3(a)-(d) is hardly visible in Figure 3(e). This is probably due to the fact that this outer shell is a very thin layer of the chosen material. Even though we slightly "drift away" from our initial goal to approximate direct volume rendering as closely as possible, we consider the capability to add information about these thin outer layers as advantageous. On the other hand, we believe that these layers could also be visualized using direct volume rendering with appropriate higher-dimensional transfer functions and/or higher sampling rates.

Another example of multiple transparent isosurface rendering is given in Figure 4. The isosurfaces are extracted from the "nucleon" data set. Standard isosurfaces with uniform 4(a) or material color 4(b) are compared to material-enriched surfaces with colors for near material 4(c) or blended colors 4(d).[4] Although the geometry does not change, material-enhanced isosurfaces clearly can reveal the different rates of change around an isosurface. Thus, a more general transfer function can be implemented.

One disadvantage of multiple isosurfaces compared to direct volume rendering is their vulnerability to noise. In a volume rendering, noise appears as "dust" of "fog", while in a multiple-isosurface method, noise manifest itself as jarring, jagged triangle groups. A noise reduction algorithm can solve this problem, but would also affect the data. We plan to explore whether a feature detection employed in a preprocessing step can take care of this problem instead.

We have tested our implementation on a standard PC with an Athlon 733 MHz processor. Our unoptimized prototype achieved a rendering performance of three to five frames per second. Our triangle sorting method is extremely fast, but the triangle drawing method could be improved. The rendering algorithm steps through all cells of the triangle-storing grid one at a time, regardless whether they are filled or not. In practice, this does not turn out to be a major issue; rendering time is dominated by the very high number of material shifts that occur when rendering multiple material-enhanced isosurfaces - three per triangle.

[1] Data set courtesy of SFB 382 of the German Research Council (DFG).

[2] Data set courtesy of SFB 382 of the German Research Council (DFG).

[3] A visualization of this data set using a more sophisticated direct volume rendering technique can be found at `http://www.volvis.org`.

[4] A visualization using direct volume rendering can be found at `http://www.volvis.org`.

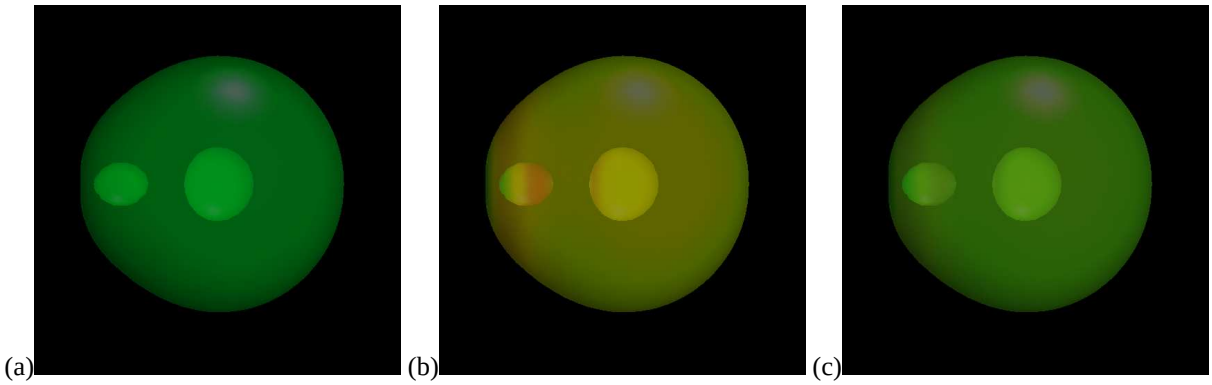

Figure 2: Single isosurface rendering applied to “nucleon” data set: (a) standard isosurface; (b) material-enriched isosurface using color of material v_{near}; (c) material-enriched isosurface using color blending.

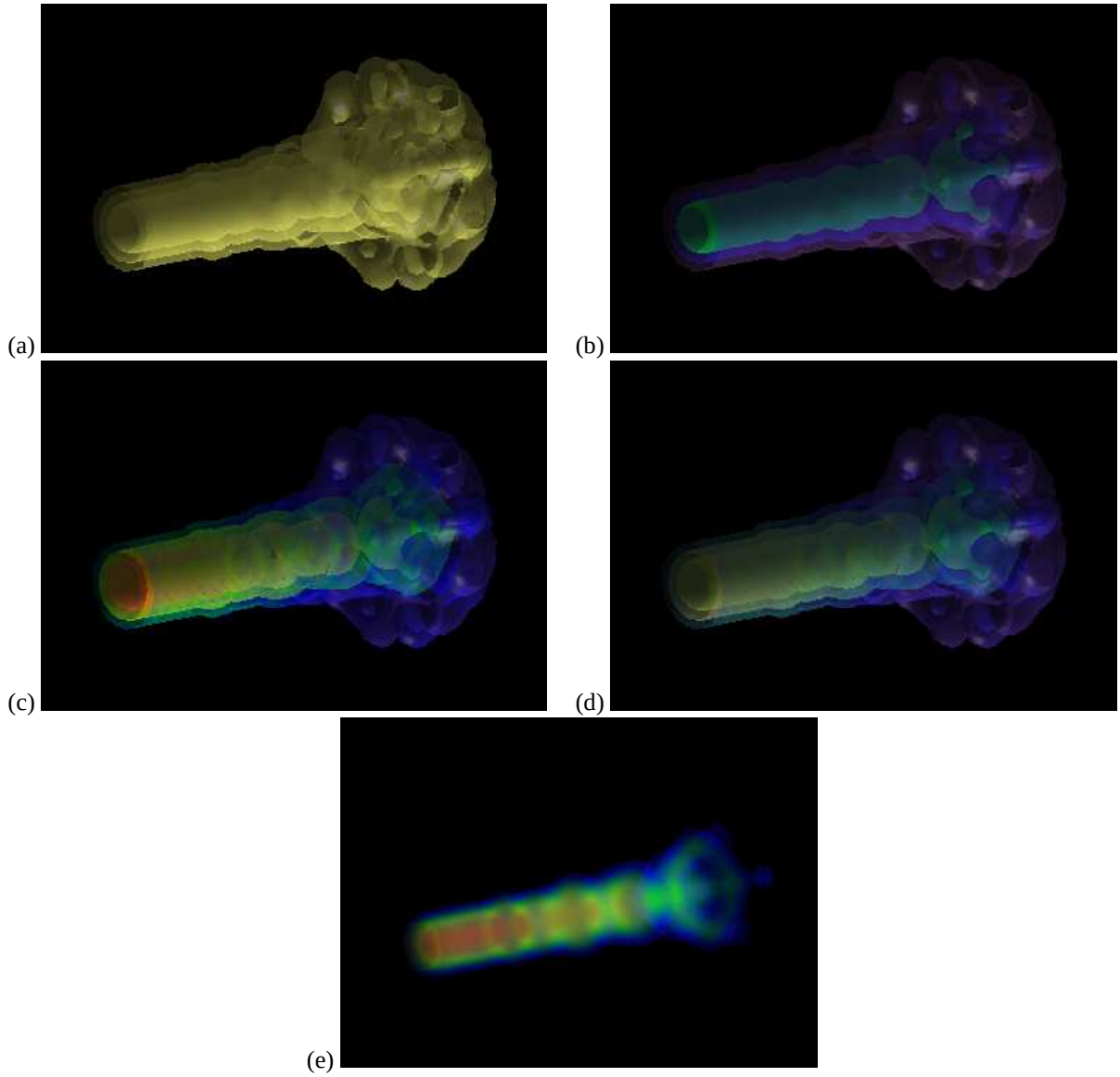

Figure 3: Multiple transparent isosurface rendering and direct volume rendering applied to “fuel injection” data set: (a) standard isosurfaces; (b) standard isosurfaces with material color; (c) material-enriched isosurfaces using color of material v_{near}; (d) material-enriched isosurfaces using color blending; (e) direct volume rendering.

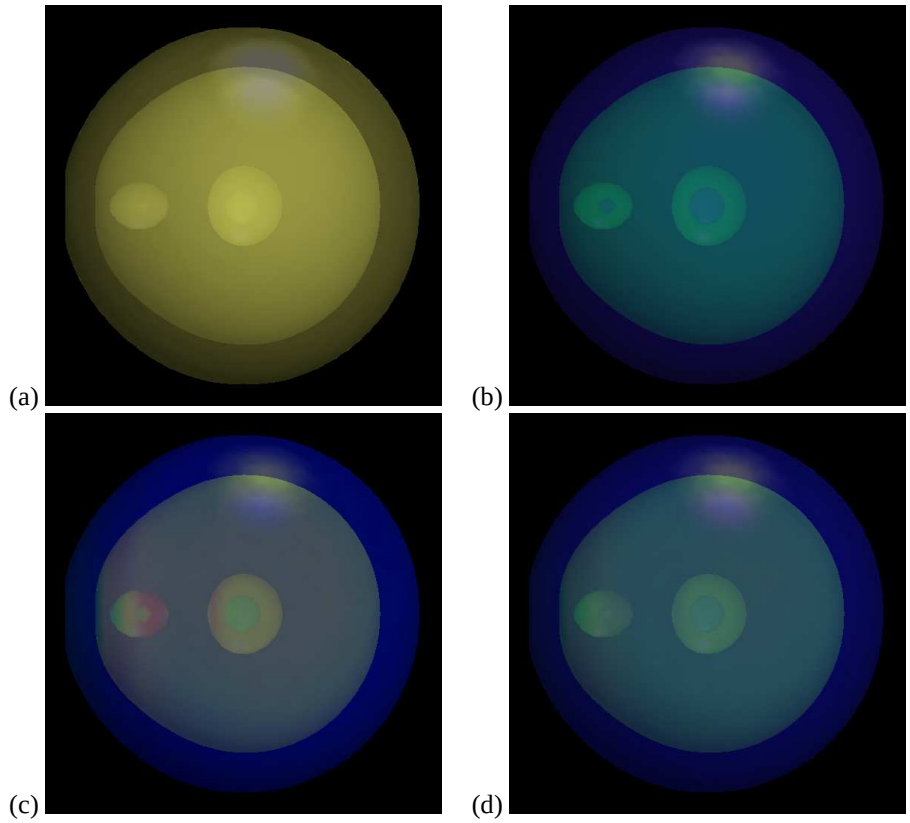

Figure 4: Multiple transparent isosurface rendering and direct volume rendering applied to "nucleon" data set: (a) standard isosurfaces; (b) standard isosurfaces with material color; (c) material-enriched isosurfaces using color of material v_{near}; (d) material-enriched isosurfaces using color blending.

Still, there is room for improvement. A run-length encoding would allow large empty spaces to be skipped. After isosurface extraction, we would determine for each cell how far the next non-empty cell is. Then, during rendering, instead of stepping through the cells one-by-one, we would use the stored information to skip all empty cells. However, the run-length encoding would have to be done six times, storing how many cells to be skipped for each of the possible six drawing orders. OpenGL display lists could also work, but constructing six of them would be necessary to cover all possibilities. Alternatively, one could use a hierarchical volumetric data organization such as an octree data structure.

6 Conclusions and Future Work

We have presented an enhancement of isosurface rendering based on coloring isosurfaces with respect to material information at a short distance from the surface in surface normal direction. Already a single isosurface visualization can benefit from the enhancement. A single well-chosen material-enriched isosurface can provide a fairly good insight into the nature of the underlying scalar field. When using multiple transparent material-enriched isosurfaces, a volumetric impression of volume data can be achieved similar to a visualization using direct volume rendering.

Multiple transparent material-enriched isosurfaces diminish the drawback of standard isosurfaces, which do not represent any information of the volume data apart from a few selected material boundaries. In particular, standard isosurfaces do not capture any gradient information of a scalar field. Adding more and more isosurfaces would eventually lead to a banding effect. By visualizing material information close to the isosurfaces, material-enriched isosurfaces capture both material boundaries and gradient information, leading to a more complete visual depiction of the overall data set.

One idea to expand upon would be to weight the blending of the color of the "iso-material" and the color of the near material. We plan on basing the weight coefficients on the opacity values of the two materials obtained from the transfer function. Our approach should easily support such a change, which may produce an even better approximation to direct volume rendering

with arbitrary transfer functions.

While diminishing drawbacks of standard isosurfaces rendering, multiple transparent material-enriched isosurfaces still benefit from the advantages of isosurface rendering: Rendering is still fast (due to our fast sorting method), lighting is easy, fast and of high quality, and material boundaries are defined explicitely, if required, e. g., for quantitative analyses.

Acknowledgments

This work was supported by the National Science Foundation under contract ACI 9624034 (CAREER Award), through the Large Scientific and Software Data Set Visualization (LSSDSV) program under contract ACI 9982251, through the National Partnership for Advanced Computational Infrastructure (NPACI) and a large Information Technology Research (ITR) grant; the National Institutes of Health under contract P20 MH60975-06A2, funded by the National Institute of Mental Health and the National Science Foundation; and the U.S. Bureau of Reclamation. We thank the members of the Visualization and Computer Graphics Research Group at the Institute for Data Analysis and Visualization (IDAV) at the University of California, Davis.

References

[BL03] David Banks and Stephen Linton. Counting cases in marching cubes: Toward a generic algorithm for producing substitopes. In *Proceedings of IEEE Conference on Visualization 1998*. IEEE Computer Society Press, 2003.

[CCF94] Brian Cabral, Nancy Cam, and Jim Foran. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In *Proceedings of the 1994 symposium on Volume visualization*, pages 91–98. ACM Press, 1994.

[Ger02] Thomas Gerstner. Multiresolution Extraction and Rendering of Transparent Isosurfaces. *Computers & Graphics*, 26(2):219–228, 2002.

[GH95] André Guéziec and Robert Hummel. Exploiting triangulated surface extraction using tetrahedral decomposition. *IEEE Transaction on Visualization and Computer Graphics*, 1(4):328–342, 1995.

[Ham91] Bernd Hamann. *Visualization and Modeling Contours of Trivariate Functions*. PhD thesis, Arizona State Univeristy, Tempe, Arizona, 1991.

[LC87] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH 1987*, pages 163–169. ACM Press, 1987.

[Lev88] Marc Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.

[LL94] Philippe Lacroute and Marc Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques - SIGGRAPH 1994*, pages 451–458. ACM Press, 1994.

[Nie03] Gregory Nielson. MC*: Star functions for marching cubes. In *Proceedings of IEEE Conference on Visualization 2003*. IEEE Computer Society Press, 2003.

[Wes90] Lee Westover. Footprint evaluation for volume rendering. In Forest Baskett, editor, *Proceedings of the 17th annual conference on Computer graphics and interactive techniques - SIGGRAPH 1990*, pages 367–376. ACM Press, 1990.

[WTTL96] Ming Wan, Long Tang, Zesheng Tang, and Xinyou Li. Pc-based quick algorithm for rendering semi-transparent multi-isosurfaces of volumetric data. In *Proceedings of the 1996 Conference on Computer Graphics International*, page 54. IEEE Computer Society, 1996.