

Visualization of 3D Scientific Datasets Based on Interactive Clipping

Gaurav Khanduja
Department of Computer Science
Louisiana State University
Baton Rouge, Louisiana 70803,
USA
khanduja@bit.csc.lsu.edu

Bijaya B Karki
Department of Computer Science
Louisiana State University
Baton Rouge, Louisiana 70803,
USA
karki@bit.csc.lsu.edu

ABSTRACT

We have exploited clipping in combination with multiresolution rendering to visualize large-scale three-dimensional (3D) scientific datasets. Our interactive clipping approach involves the dynamic manipulation of a clip plane to expose any cross-section of a given volume data and the subsequent adjustment of the clipped surface to the best view position. The data are rendered as stacks of 2D textures at high-resolution (HR, the original resolution of input data/image) and low-resolution (LR, sampling at reduced-resolution). While clipping operations take place in LR to achieve an interactive frame rate, the best-view position also supports HR. Our approach thus enables a real-time exploration of any interior region of 3D volumetric data at a desired resolution. Its successful demonstration has been carried out for two scientific datasets, which are the simulated electronic density distributions in a crystal and confocal microscopy images of tissues in a plan stem.

Keywords

Clipping, volume rendering, scientific visualization

1. INTRODUCTION

High-end scientific computation and three-dimensional imaging systems are producing ever-increasing amounts of volume data. Visualization of such datasets still poses tremendous challenges primarily because it involves mapping from 3D space to 2D display. A number of visualization techniques exist [sch97]. One category includes surface rendering techniques, which convert structures in 3D data into surface representations, e.g., isosurfaces. The other category deals with direct volume rendering techniques, which involve the generation of a RGBA volume from the data, reconstruction of a continuous function, and projecting it onto the 2D viewing plane. No interior information is thrown away in volume rendering but it has difficult interpretation of the cloudy interiors and slow rendering.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSCG SHORT papers ISBN 80-903100-9-5
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.
Copyright UNION Agency – Science Press

The other important category of 3D visualization techniques is related to clipping, which allows us to uncover important, otherwise hidden details of a dataset to the extent which is often not feasible with other techniques. The interactive clipping approach we have adopted here is not an entirely new idea. Rather it is a practical implementation which shows how even a simple form of clipping (such as a planar clipping) can be exploited in an effective way to provide a real-time multi-resolution visualization of large-scale datasets. The clip plane once it is generated is manipulated in real time to expose an arbitrary cross-section through the volume data. The clipped surface is then adjusted to attain the best view position in which the surface is visible to its high exposure and rendered at high-resolution. A portable and flexible interactive 3D visualization system is being developed based on the proposed approach. It is expected to guarantee that one is able to view every possible cross-section of the volume data at desired resolution. Successful applications are demonstrated for two important scientific datasets. They represent the electronic charge density distributions in a solid material taken from large-scale simulations and the tissue images of a plant stem generated by confocal microscopy.

2. PREVIOUS WORK

Clipping was previously performed using a variety of clip geometry, from simple plane, to multiple planes, to complex polygons. For example, Van

Gelder and Kim used clip planes [Van96]. Weiskopf et al. have proposed techniques for volume clipping with complex geometries, which are based on the depth structure and voxelization of the clip geometry [Weisk02] and also involve subsequent shading of the clipping surface [Weisk03]. Some studies have exploited isosurface clipping [Ferg92]. Often clipping is combined with texture-based rendering of the data to exploit the advanced fragment operations supported by graphics hardware. Also hardware-based textures help improve interactivity, compared to point-based rendering [Wilson02]. However, for very large datasets now routinely produced by simulations and experiments, the amount of available texture memory and the fill rate of the graphics system can still pose limitations. To overcome these, multi-resolution rendering techniques such as combining texture mapping and point-based rendering [Wilson02] or using lower sampling rates for areas of low interest [LaMar99, Weiler00] were previously proposed. However, no clipping has yet been implemented in combination with any multi-resolution rendering option.

3. CLIPPING-BASED VISUALIZATION

We have explored how clipping can be exploited in an efficient-effective manner to visualize large-scale 3D scientific datasets by combining it with dual-resolution texture-based volume rendering. The overall goal is to support different specific services such as rapid navigation through the data, emphasis on data features and realistic rendering without sacrificing interactivity, which are crucial in real-time exploration of massive datasets. Here, interactive clipping is adopted to uncover every interior region of a given volumetric data so that one can visualize the clipped surface at its maximum exposure and at the desired level of details.

3.1 Interactive clipping

Our interactive clipping approach involves the generation and subsequent adjustment of clip geometry, which is simply a plane. The clip plane is represented by the equation $Ax + By + Cz + D = 0$ in a 3D space. OpenGL [Woo99] is used to define the clip plane and perform all transformations the plane undergoes subsequently. The clip plane like any other geometry is transformed according to the current viewing matrix. All points with eye coordinates (x_e, y_e, z_e, w_e) that satisfy

$$(A \ B \ C \ D)M^1(x_e, y_e, z_e, w_e)^T >= 0$$

lie in the half-space defined by the plane, where M is the current model-view matrix. All points not in this half-space are clipped away.

The clip plane is initiated by choosing any two points, (x_1, y_1) and (x_2, y_2) , on the display screen, which is initially set parallel to the x - y plane. The initial plane is assumed to be parallel to the z -axis

passing through the origin. The required coefficients are thus given by

$$A = y_2 - y_1; B = -(x_2 - x_1); C = 0; D = 0$$

Once the clip plane is defined, its position and orientation can be changed in real time in order to cut away the given volume at any orientation and any position. Varying the coefficients A , B and C enables an arbitrary rotation of the clip plane in a 3D space. For translation, the coefficient D is varied; the plane moves away from the origin in either direction along the normal to the plane passing through the origin.

3.2 Best-view mode

In the process of interactively changing the position and orientation of the clip plane, it is likely that effective visible area of the clipped surface becomes smaller and eventually disappears. In other words, the user cannot see the surface clearly because the plane tends to be parallel to the viewing direction. In such case, it is important to adjust the position/orientation of the clipped geometry so that the surface is again visible at larger exposure. Here, we want to introduce the idea of the *best-view* mode by which we mean the following two properties associated with it. First, the clipped surface is oriented nearly perpendicular to the viewing direction so that the surface is portrayed at its nearly maximum exposure. Second, the cutaway surface is rendered at high-resolution so that the details of data are displayed. The second feature can further be extended to support multi-resolution and additional display options such as isolines or glyphs or streamlines depending on the properties of the data that need to be understood.

To guarantee an interactive frame rate, particularly, when the datasets become very large, it is desirable to support an automatic option to attain the *best-view* position whenever it is needed. This requires tracking all transformations the clip plane has gone through and using this information to change the current view of the clipped object accordingly. For instance, a simple approach, say for our initial setting of the clip plane, involves the following two rotations: The first rotation is about the z -axis since the initial clip plane is set parallel to the z -axis and is rotated about this axis. The second rotation is about the vector, which is defined by the intersection of the clip and x - y planes. As a result, the clipped surface becomes nearly perpendicular to the z -axis, which is also the current viewing direction. After reaching the *best-view* position, the clip plane can further be manipulated by bringing back the object to the low-resolution mode.

3.3 Rendering

A given 3D volume dataset is rendered as stack of 2D slices in the form of 2D textures [Cab94]. Rendering is performed at two levels of resolution, n

the form of 2D textures. The high-resolution (HR) mode corresponds to the original fine grid of the input data (or equivalently, the original resolution of the input images). On the other hand, the low-resolution (LR) mode uses the re-sampled data at a lower resolution than the original resolution. Rendering speed (number of frames per second) is shown to differ substantially between the HR and LR rendering modes. HR is not fast enough to yield an interactive frame rate even for the data sizes we have considered here. However, it carries the complete information contained in the data. On the other hand, LR gives a much better frame rate but it suffers from the loss of information. The observed trade-off between rendering speed and resolution is carefully balanced in our visualization approach. In particular, the planar clipping is supported in the LR

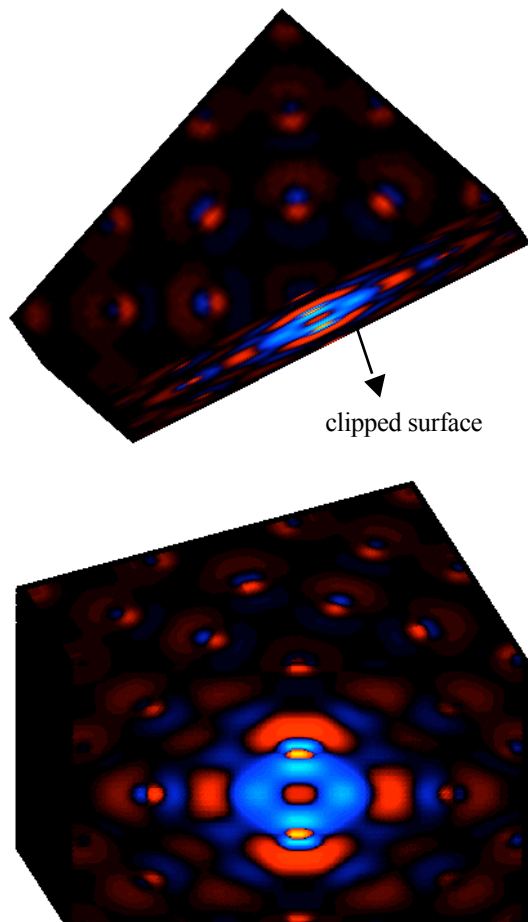


Figure 1: Charge density visualization: Clipping in the LR mode (top) and best-view rendering in the HR (bottom) modes. Electrons are depleted from blue regions whereas electrons are deposited in red regions due to a vacancy defect located at the center of the system.

mode whereas the *best-view* option is also supported in the HR mode. This means that one can navigate through the volume data in the LR mode without hampering interactivity and switch to the HR mode to retain the details of data.

4. APPLICATIONS

4.1. Charge density data

Here, we visualize the 3D electronic density data produced by massive quantum mechanical simulations. The simulated MgO system consists of 216 atoms and 864 valence electrons, which are distributed throughout the system. The electron density data on a regular Cartesian 512^3 grid is considered. Such simulations are expected to produce even larger datasets, which need to be visualized to gain insight into the electronic structures in the presence of static (such as defects) and dynamic (such as diffusion, deformation) processes in a material system. The given data is mapped to 512 textures; each texture has 512×512 pixels. Rendering them as a stack results in the HR mode. A stack of 128 textures at 128×128 resolution is used for the LR mode. Figure 1 shows clipped surface in the HR and LR modes. The performance measurements are carried out on Dell latitude of 2.40 GHz processor, 1.00 GB RAM for the window display size of 512^2 . The calculated frame rates for clipping in the LR and HR modes are, respectively, 35 and 4. This difference in the clipping speed is substantial and should be important for larger datasets.

4.2. Confocal data

Three-dimensional imaging systems often produce large collections of 2D images of the interior of a 3D structure at different depths. The present application considers the confocal data containing 200 images, each with 512×512 pixels, of tissues of a plant stem. The visualization process starts with reconstruction of a 3D structure to which these images originally belong by rendering the given set of 200 tissue images as a stack of 2D textures. HR, which retains the original resolution of the input images, represents the one-to-one mapping between texels and pixels. LR corresponds to the stack of 50 textures at 128×128 resolution. Figure 2 shows clipped surface in the HR and LR modes. For 512^2 window display, the calculated frame rates for clipping in the LR and HR modes are, respectively, 28 and 0.6.

5. CONCLUSIONS

We have explored how clipping can be exploited in an interactive manner to visualize massive three-dimensional datasets (for calculated charge density and confocal tissue images) produced by large-scale scientific simulations and 3D imaging systems. In essence, the proposed approach involves the dynamic manipulation of the clip plane to expose any cross-

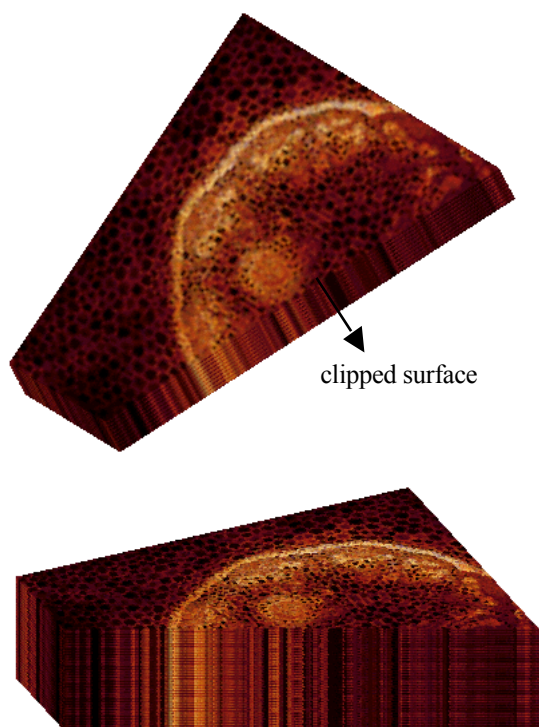


Figure 2: Confocal data visualization: Clipping in LR (top) and best-view rendering in HR (bottom). Twenty-five images (each with 512 x 512 pixels) were replicated to generate 200 images used in HR.

section of the volume data and subsequent adjustment of the clipped surface to the best view position. While all clipping operations are supported in the low-resolution (LR) mode to achieve interactivity, the best-view position is also supported in the high-resolution (HR) mode (i.e., the original resolution of the input data/image). It is expected that the proposed interactive clipping combined with dual-resolution rendering (which can essentially be extended to multi-resolution and multi-scale rendering) will be a powerful approach to interactive visualization of ever larger sets of three-dimensional scientific data.

6. ACKNOWLEDGMENTS

Our thanks go to J. Lynn for providing the confocal data, and S.S Iyengar and R. Kannan for useful discussion. This work is supported by NSF Career (EAR 0347204) and ITR (ATM 0426601) grants.

7. REFERENCES

- [Cab94] B. Cabral, N. Cam and J. Foran. Accelerated volume rendering and tomographic reconstruction using texture-mapping hardware. In *1994 Symposium on Volume Visualization*, pages 91-98, 1994.
- [Forg92] Forguson. Visual Kinematics, Inc. (Mountain View, CA USA), FOCUS User Manual, Release 1.2., 1992.
- [LaMar99] E. LaMar, B. Hamann, and K. Joy. Multiresolution techniques for interactive texture-based volume visualization. In *Proceedings Visualization 99*, pages 355-361. ACM Press, New York, 1999.
- [Sch97] W. Schroeder, K. Martin and B. Lorensen. *Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*, Prentice Hall, 1997.
- [Weiler00] M. Weiler, R. Westermann, C. Hansen, K. Zimmerman, and T. Ertl. Level-of-detail volume rendering via 3D textures. In *Proceedings of IEEE Volume Visualization 2000*, 2000.
- [Weisk02] D. Weiskopf, K. Engel and T. Ertl. Volume clipping via per-fragment operations in texture-based volume visualization, *IEEE Visualization 2002 Proceedings*, pages 93-100, ACM Press, 2002.
- [Weisk03] D. Weiskopf, K. Engel, and T. Ertl. Interactive clipping techniques for texture-based volume visualization and volume shading. *IEEE Trans. on Visualization and Computer Graphics*, 2003.
- [Wilson02] B. Wilson, K. Ma and P. S. McCormick. A hardware-assisted hybrid rendering technique for interactive volume visualization. *Proc. of Volume Visualization and Graphics Symposium 2002*, 2002.
- [Woo99] M. Woo, J. Neider, T. Davis and D. Shreiner. *OpenGL Programming Guide: The Official Guide to Learning OpenGL*, Addison-Wesley, 1999.
- [Van96] A. Van Gelder and K. Kim. Direct volume rendering with shading via three-dimensional textures. In *1996 Symposium on Volume Visualization*, pages 23-30, 1996.