Four-Dimensional Views of 3D Scalar Fields

Andrew J. Hanson

CN/AS Division
CERN
CH-1211, Geneva 23, Switzerland
and
Department of Computer Science
Indiana University
Bloomington, IN 47405

Pheng A. Heng

Department of Computer Science
Indiana University
Bloomington, IN 47405

Abstract

Scalar functions of three variables, $w = f(x, y, z)$, are common in many types of scientific and medical applications. Such 3D scalar fields can be understood as elevation maps in four dimensions, with three independent variables (x, y, z) and a fourth, dependent, variable w that corresponds to the elevations. We show how techniques developed originally for the display of 3-manifolds in 4D Euclidean space can be adapted to visualize 3D scalar fields in a variety of ways.

1 Introduction

We examine the problem of visualizing scalar functions of three variables, or 3D scalar fields. We adopt the point of view that, just as scalar functions of one and two variables can be viewed as elevation maps in 2D and 3D Cartesian coordinate systems, scalar functions of three variables are elevation maps in a 4D coordinate system, where the value of the function is identified with the fourth dimension.

That is, if we can visualize $y = f(x)$ as a curve in a 2D Cartesian plane, and $z = f(x, y)$ as a 3D surface viewable from oblique angles in 3D space, then the function $w = f(x, y, z)$ should be representable as a 4D volume (technically, a 3-manifold embedded in 4D Euclidean space) viewable from oblique angles in 4D space.

Previous Work. Surfaces of the form $z = f(x, y)$ in 3D space may be represented graphically in a variety of ways, including contour plots in a plane, elevation-keyed pseudocolor plots in a plane, oblique views of rectangular grids projected onto the surface, oblique views with elevation-keyed surface pseudocolors, and shaded surface rendering. For volume data of the form $w = f(x, y, z)$, extensive attention has been paid to the 3D analogs of 2D contour plots and pseudocolor maps: see, for example, various methods for constructing isosurfaces [11, 1], viewing color-coded density data [14, 12, 6], making easily visible pseudocolor maps [13], and exploiting transport theory [10]. However, except for some exploratory efforts (see, e.g., [4, 9]), the analogs of elevation maps for 3D scalar fields have not been pursued to their logical completion.

The purpose of this paper is to extend methods developed for the rendering of mathematical 3-manifolds embedded in 4D Euclidean space [3, 15, 7, 8] to create a more complete family of volumetric analogs of the standard elevation map techniques used for surface representation.

Approach. We suggest that new insights into the properties of 3D scalar fields can be achieved by utilizing the information redundancy present in 4D-rotated pseudocolor volume grids, and by rendering the corresponding 3-manifolds using 4D lighting and shading rules. In particular, the following features of the 4D treatment of 3D scalar fields do not seem to have been fully exploited previously:

- **Grid Planes.** Grid lines in 3D plots correspond to the lines (x = constant, y = constant) that divide the surface $z = f(x, y)$ into square patches. The 3-manifold $w = f(x, y, z)$ is analogously divisible into cubic patches by the grid planes, (x = const, y = const, z = const).

  Just as *tilted* grid lines are useful in standard 3D grid plots to give depth cues in 3D, the distortions of the grid planes resulting from 4D rotation of the viewpoint provide strong geometric cues.

0-8186-2897-9/92 $03.00 © 1992 IEEE

- **Hidden Volume Removal.** Clarity is enhanced in 3D by removing hidden surfaces, so that surfaces nearer to the viewpoint occlude those farther away. This is equivalent to making the surface stretched between grid lines opaque and carrying out z-buffered rendering. In 4D, the grid planes enclose *volumes* rather than *areas*. Thus, the equivalent procedure in 4D is to use 4D depth buffering (w-buffering) relative to a 4D viewpoint to carry out hidden volume removal. This will cause various pieces of volumes to disappear from the voxel array after 4D rotation because they become occluded by nearer neighbors.

- **Shading Using 4D Light.** There is intrinsic information present in a 3D scalar field viewed as a 4D elevation that is not revealed by projecting level sets to 3D and using 3D lighting. By applying 4D lighting, shading, and specularity algorithms in the manner of [7, 8] while projecting to the 3D voxel array, we can produce a uniquely informative set of shading patterns. In particular, portions of the hypersurface that face in similar directions in 4D can be clearly perceived, so information about derivatives can be visualized more readily.

In the remainder of the paper, we first give graphical examples of a number of standard methods for handling 2D scalar fields that we will refer to when constructing analogies for 3D scalar fields. We then describe 4D analogs of 3D methods for viewing elevation grids, outline 4D shading methods, and show some examples.

2 Three-Dimensional Shape Plots

Functions of the form $z = f(x, y)$ can be displayed using standard software packages such as NCSA Image, Mathematica, gnuplot, etc., to show the shape of the function. Here, by way of introduction to the 4D methods, we show a family of standard 3D plotting methods generated by the PAW package [2].

Let us take as examples the equations

$$z = \sqrt{\max(0, 1 - x^2 - y^2)} \qquad (1)$$

for a hemisphere plotted in the range $-1 < x < 1$ and $-1 < y < 1$, and

$$z = \exp(-x^2 - y^2) + \frac{1}{2}\exp(-(x+3)^2 - (y+3)^2) \quad (2)$$

for a double Gaussian bump plotted in the range $-5 < x < 2$ and $-5 < y < 2$.

Then, as shown in Figures 1, 2, and 5, we can plot the hemisphere (Equation (1)) in the following ways:

- **Contours and Icons.** In Figure 1(a), we show the "straight-down" view of the hemisphere contour plot. The only cues we have for the depth are the labels or styles of the contour lines. A related method, shown in 1(b), plots icons (e.g., squares) whose size is keyed to the z-value at equally spaced (x, y) grid points.

- **Surface Grids.** In Figures 1(c) and (d), we show oblique views of a block or "lego" representation and of a grid-line plot with hidden surfaces removed. These approaches provide additional clarity to the viewer because only the nearest block tops or patches of the tilted checkerboard pattern are visible; areas lying on the grid behind a foreground patch are hidden from view. In 1(d), the distorted projections of individual outlines that are perfect squares in the straight-down view (see 5(c)) also give additional depth information.

- **Pseudocolor Contours.** Representing the contours as color areas instead of grid lines, as in Figure 5(a), makes it easier to distinguish limited ranges of elevation if the color map is carefully chosen. No geometric cues are present.

- **Oblique Pseudocolor Contour.** In Figure 5(b), we combine the information in the pseudocolor contour plot with the geometric cues of an oblique view of the colored surface without a grid. This provides very useful *redundant* cues for the function values, since both the color and the 3D aspect of the image contain similar information about z.

- **Pseudocolor Straight-Down Grid.** In Figure 5(c), we superimpose a straight-down view of the grid outline, seen obliquely in Figures 1(d) and 5(d), on the pseudocolor contours. Again, no geometric cues are present.

- **Pseudocolor Oblique Grid.** However, if we make an oblique view, as in Figure 5(d), we get additional information, now *triply* redundant: the color, the hidden surface effect, and the distortion of the grid lines. This image is the richest we can construct because it exhibits the shape information to the viewer in multiple modes with little ambiguity.

- **Shaded Specular Surface.** In Figure 2, we show how the shaded hemisphere would appear from an assortment of viewpoints. As we move from the straight-down viewpoint Figure 2(a),

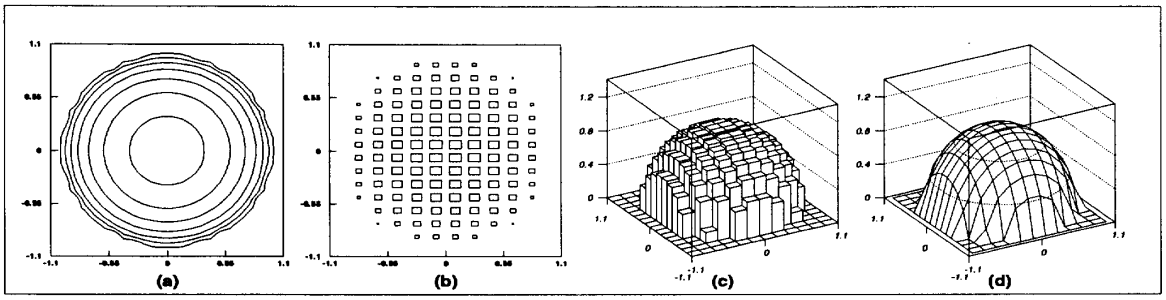

Figure 1: Line-drawing approaches to displaying the 3D structure of a hemisphere, a simple example of a 2D scalar field given by Equation (1). (a) Straight-down contour map. (b) Straight-down variable-size icon map. (c) Oblique block or "lego" map. (d) Oblique grid line elevation map. (PAW images [2].)

which has no occlusion, to a very oblique viewpoint Figure 2(d), we see the hemisphere "rising out of the background," in a manner of speaking. The most oblique view has the most occlusion, and also shows the profile of the highest elevation points dramatically. The regions with high specular reflection give additional cues about surface orientation.

Next, we show in Figure 6 a similar sequence of displays for the double Gaussian bump, Equation (2).

- **Grid.** As shown in Figure 6(a), we can get a very good image of the 3D structure using just the oblique perspective of the bare grid with hidden surfaces removed.
- **Pseudocolor.** Adding pseudocolor to a perspective view with no grid structure as in Figure 6(b) gives dual cues — the oblique view of the elevation with hidden surfaces removed, and the redundant color assignment of each elevation contour.
- **Pseudocolor Grid.** The distortion of the grid squares in the oblique projection adds a third cue, as shown in Figure 6(c).
- **Shaded Specular Surface.** In Figure 6(d), we assume a reflective surface, introduce 3D lighting, and use standard shading techniques [5] to create a shaded scene rendering. This image has a very natural aspect, and contains information about the shape of the surface that is compelling for most viewers. This shaded view contains information about the surface orientation that is more explicit than in the other methods.

3 Four-Dimensional Shape Plots

Figures 1(a), 1(b), 2(a), 5(a), and 5(c) are the simplest examples of 2D scalar field representations. The data plots were generated by looking "straight down" at the data and visualizing the values of the elevations in various ways with this constraint. Analogs of each of these depictions can be constructed for 3D scalar fields as well. For example, contour lines for 3D scalar fields are basically nested shells produced by an isosurface construction algorithm [11, 1]. If we add pseudocolor to indicate where the level sets are located within the volume, we can use the "tiny cubes" algorithm of Nielson et al. [13] to make the interior voxels of the cube visible.

3.1 Pseudocolor 4D Hemisphere

We can easily define a 4D analog of the hemisphere we used for illustration in Figures 1, 2, and 5 as

$$w = \sqrt{\max(0,\, 1 - x^2 - y^2 - z^2)}\,. \qquad (3)$$

The 4D analog of the 3D method used to produce Figure 5(c) gives the cubic representation of the "hemispherical" 3D scalar field shown in the upper left of Figure 7. Here the pseudocolored cubic latticework is an undistorted "3D checkerboard," just as the straight-down view of the square checkerboard grid is undistorted in Figure 5(c). We may of course choose to alter the 3D viewpoint, but this does not alter the actual lattice geometry.

3.2 4D "Transparent" Volume Grids

If we now rotate in 4D before projecting to 3D, we introduce a distortion in the apparent form of the cubic lattice that persists regardless of how we change the 3D viewpoint. The image in Figure 3 shows this effect *without* attempting to distinguish whether one

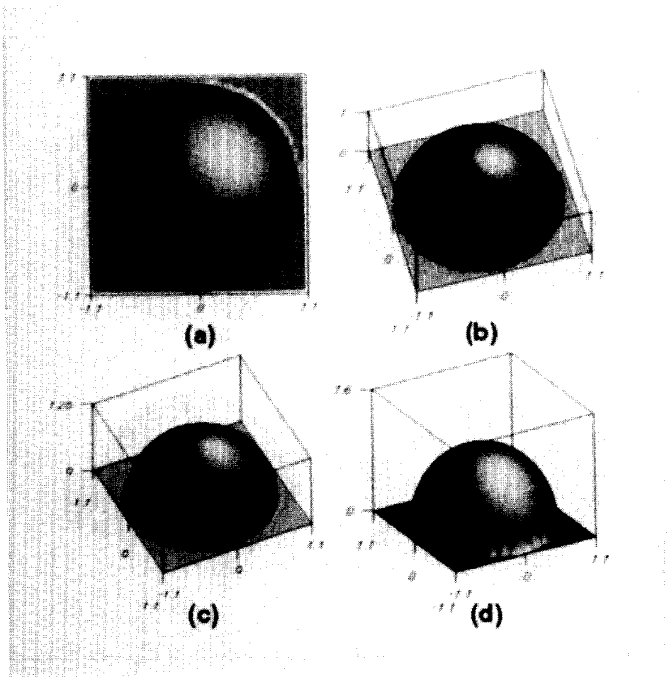

Figure 2: Four 3D viewpoints of the hemispherical 2D scalar field given by Equation (1). (a) is straight down, and has no occlusion. (b) and (c) show the hemisphere as it rises from the background as we rotate the 3D viewpoint; we begin to see some occlusion. Finally, in (d) we see a very oblique view with extensive occlusion and a profile explicitly showing the elevation of the hemisphere above the background. (PAW images [2].)

lattice cube hides another in 4D. Figure 3 treats the small 4D cubes making up the grid volume as transparent in 4D, but *opaque* once projected to 3D; this makes the resulting image easier to interpret as a 3D structure. This image is the analog of Figure 1(d) if we displayed the surface using small tilted quadrilaterals with open space around the edges and did not remove back-facing surfaces.

3.3 4D Pseudocolor Grid

By assigning a pseudocolor map to the values in the voxel array, we can generate a volume image with scalar values coded by color. Rotating the grid in 4D causes these colored regions to shift in such a way that a redundant value cue is produced, as we show in Figure 7. The pseudocolor attached to any particular volume element remains fixed to a particular element, just as it did when we rotated in 3D to make the transition from Figure 5(c) to Figure 5(d). The lower right image in Figure 7 is similar in spirit to Figures 5(d) and 6(c) due to the three-fold redundancy of the information display.

3.4 Hidden Volume Removal in 4D

The grid can now be rendered using 4D depth buffering to eliminate confusion between overlapping volumes, just as we used hidden surface elimination in 3D to make the image of Figure 1(d). A typical result from a 4D viewpoint is shown in Figure 8. Now we see a very strange phenomenon from the point of view of an observer accustomed to 3D mesh plots: instead of having tilted rectangles hidden by other tilted rectangles in the foreground, we have distorted cubes that are truncated by other opaque distorted cubes that sit in front of them in 4D.

4 4D Illumination and Shading

Our final method is the 4D analog of the 3D shading method used in Figures 2 and 6(d). We use lighting and shading in four dimensions to reveal geometric characteristics of the 3D scalar field's elevation map.

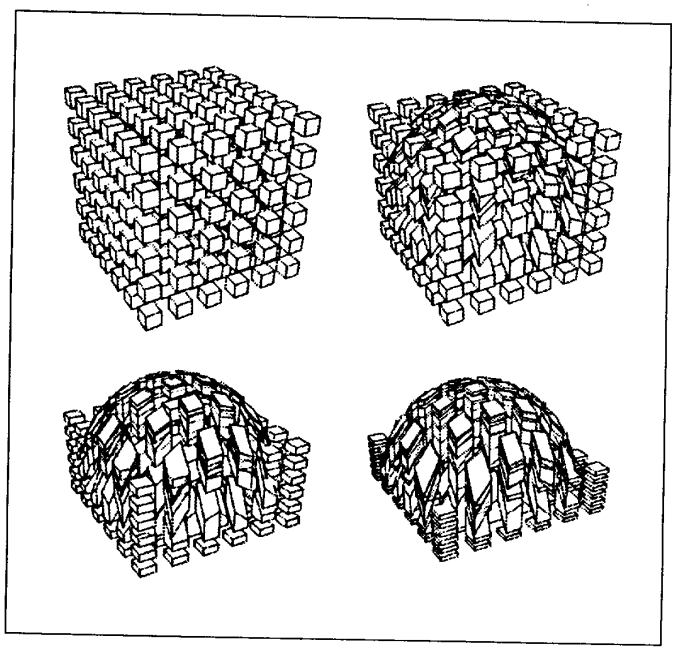

Figure 3: An image of a 4D hemisphere represented as a bare grid without removing hidden 4D volumes, but treating solid cubes in the 3D projection as opaque. We show four stages in the 4D rotation.

In particular, this method provides a quick insight into regions of the elevation map that have similar 4D normals. The details are given in Hanson and Heng [7, 8]; we provide a brief overview here for completeness.

Four-dimensional shading models with specularity can be introduced by computing the normals at each vertex, either by taking the gradient of the equation defining the 3-manifold, taking the cross-product of the three tangents to a grid point on the 3-manifold in 4D, or averaging the normals to the volumes surrounding the point. For parametric volumes, this computation involves computing the three tangent four-vectors $(\vec{P}, \vec{Q}, \vec{R})$ emerging from the grid point in "right-handed" order. The four-vector that is perpendicular to this family of tangents has components that are the cofactors of the first column of the following determinant, which is the generalization of the cross-product to 4D:

$$\mathrm{Det}\begin{vmatrix} \hat{x} & P_1 & Q_1 & R_1 \\ \hat{y} & P_2 & Q_2 & R_2 \\ \hat{z} & P_3 & Q_3 & R_3 \\ \hat{w} & P_4 & Q_4 & R_4 \end{vmatrix} . \qquad (4)$$

To compute the normal at the grid point, we normalize the four-vector $\vec{N}$ given by the coefficients of $(\hat{x}, \hat{y}, \hat{z}, \hat{w})$ in Equation (4) and call that $\hat{N}$. If all we know are the tangents to the volumes surrounding the vertex, we compute all the volume normals using a determinant of this same form, normalize the resulting normals, average them, and normalize again to get the grid-point normal $\hat{N}$.

However one finds the normal at each 4D vertex point, the result is then used in the standard shading equations (see [5] for 3D and [7] for additional 4D discussion). The smooth shaded term with normalized lighting vector $\hat{L}$,

$$I_G = I_0 \, \hat{L} \cdot \hat{N} \, , \qquad (5)$$

is added to the specular term,

$$I_P = I_1 \left| \hat{B} \cdot \hat{N} \right|^k , \qquad (6)$$

where $\hat{B}$ is the normalized average of the direction to the camera and the direction to the light source. The

intensity in either term is generally set to zero if the corresponding dot product is negative.

One then projects volume elements from the tessellated 3-manifold in 4D space to a 3D view volume of voxels, interpolates the normals from voxel to voxel, applies the selected shading algorithm, and renders the resulting intensity into the voxel array. Such view volume images are 4D analogs of the 3D view plane images in Figure 2. The voxel data must itself be volume rendered to produce an ordinary image. In Figure 4, we apply this approach to render the hemisphere-like 3-manifold given by Equation (3).

5 Example Applications

To illustrate the application of some of our methods to real data, we first show in Figure 9 a comparison between two ways of rendering a set of astronomical binary star density data. On the left is a standard volume rendering produced by the AVS "tracer" module [16]. On the right, we show a shaded rendering of the same data using our 4D lighting technique. Note that our actual rendering is a volume image, which must itself be volume rendered to produce this 2D image; a rotating animation or a stereographic image would show the internal structure of our depiction more clearly. Previously invisible features, basically corresponding to regions with similar 4D gradients, are revealed by our approach.

Next we repeat the same comparison for a data set describing the electron density of the hydrogen molecule. The left image in Figure 10 shows a standard AVS volume rendering. On the right, we show for comparison a 4D shaded rendering of the density elevation plot. Again, we see suggestive additional structure in our 4D rendering. We have proposed many other related 4D visualization methods as well, but cannot present further examples of them all here; the fundamental feature of our other methods is the systematic exploitation of redundant geometric cues in four dimensions, as exemplified by Figures 7 and 8.

6 Remarks: 3D Fields as 4D Images

Since one of the main techniques introduced in this paper was the creation of a volume image from 3D scalar field elevation maps using 4D lighting and shading methods, one might imagine reversing the process. That is, one might consider the 3D scalar field data itself to be the result of a 4D imaging process. Given some reasonable assumptions about what 4D lighting parameters produced the data, one could in principle reconstruct some information about what sort of 4D object could have produced the observed image. Unfortunately, just what constitutes a "reasonable" assumption is extremely ambiguous, so that we have not been able to systematically deduce relevant interpretations of 3D scalar fields as images of 4D objects (unless, of course, the fields happen to be derived from a 4D imaging process). This nevertheless remains an intriguing additional approach related to the ones presented in this paper, and one which might be of use in some special circumstances.

Acknowledgments

We gratefully acknowledge the support of the staff and the use of the facilities at CICA, the Indiana University Center for Innovative Computer Applications. We are indebted to Brian Kaplan and Peter Shirley for suggestive remarks and for access to the volume data used in the application figures. A.J.H. thanks René Brun and the CN/AS group at CERN for their patience in revealing the workings of the PAW system.

References

[1] H.H. Baker, "Building surfaces of evolution: the weaving wall," *International Journal of Computer Vision*, Vol. 3, No. 1, pp. 51–71, 1989.

[2] R. Brun, O. Couet, C. Vandoni, and P. Zanarini, "PAW – Physics Analysis Workstation, The Complete Reference," Version 1.07, CERN, Geneva, Switzerland, October 1989.

[3] S.A. Carey, R.P. Burton, and D.M. Campbell, "Shades of a Higher Dimension," *Computer Graphics World*, pp. 93–94, October 1987.

[4] S. Feiner and C. Beshers, "Visualizing n-Dimensional Virtual Worlds with n-Vision," *Computer Graphics*, Vol. 24, No. 2, pp. 37–38, March 1990.

[5] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes, *Computer Graphics: Principles and Practice*, Second Edition, Addison Wesley, Reading, MA, 1990.

[6] H. Fuchs, M. Levoy, and S.M. Pizer, "Interactive Visualization of 3D Medical Data," *IEEE Computer*, Vol. 22, No. 8, pp. 46–51, August 1989.

[7] A.J. Hanson and P.A. Heng, "Visualizing the Fourth Dimension Using Geometry and Light," in Proceedings of *Visualization 91*, San Diego, CA, Oct. 22–25, pp. 321–328, IEEE Computer Society Press, Los Alamitos, CA, 1991.

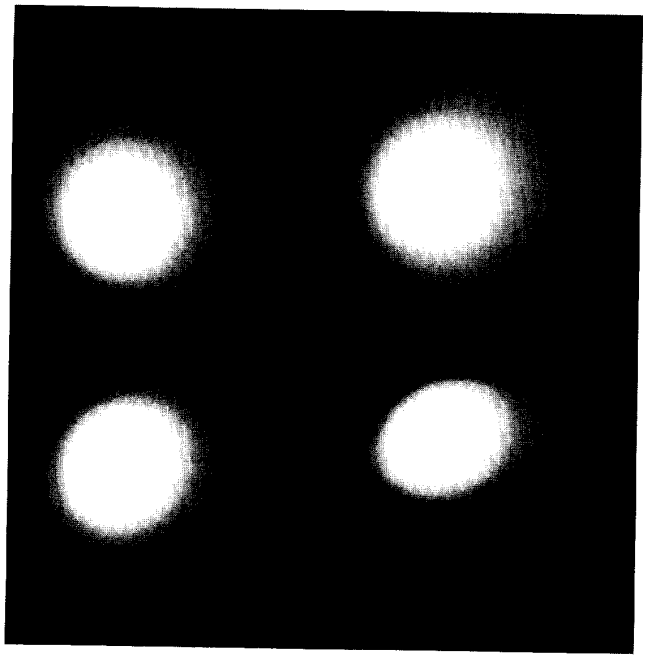

Figure 4: 4D lighting and shading applied to the hemispherical 3D scalar field of Equation (3). The four different 4D viewing angles shown here exhibit the 4D analog of the 3D view-dependent shading effects in Figure 2.

[8] A.J. Hanson and P.A. Heng, "Illuminating the Fourth Dimension," *IEEE Comp. Graphics and Appl., in press*, July, 1992.

[9] C. Hoffmann and J. Zhou, "Displaying Manifolds in Four-Dimensional Space," in Purdue University Computer Science Department Report: *Computing About Physical Objects*, Purdue University, Lafayette, IN, 1991.

[10] W. Krueger, "The Application of Transport Theory to Visualization of 3-D Scalar Data Fields," *Computers in Physics*, pp. 397–406, July-August 1991.

[11] W.E. Lorensen and H.E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics*, Vol. 21, No. 4, pp. 163–169. Proceedings of SIGGRAPH 87, July 1987.

[12] G.M. Nielson, T.A. Foley, B. Hamann, and D. Lane, "Visualizing and Modeling Scattered Multivariate Data," *IEEE Comp. Graphics and Appl.*, Vol. 11, No. 3, pp. 47–55, May 1991.

[13] G.M. Nielson and N. Hamann, "Techniques for the Interactive Visualization of Volumetric Data," in Proceedings of *Visualization 90*, San Francisco, CA, Oct. 23–26, pp. 45–50, IEEE Computer Society Press, Los Alamitos, CA, 1990.

[14] P. Shirley and A. Tuchman, "A Polygonal Approximation to Direct Scalar Volume Rendering," *Computer Graphics*, Vol. 24, No. 5, pp. 63–70, November 1990.

[15] K.V. Steiner and R.P. Burton, "Hidden Volumes: The 4th Dimension," *Computer Graphics World*, pp. 71–74, February 1987.

[16] C. Upson, T. Faulhaber, D. Kamins, D. Laidlaw, D. Schlegel, J. Vroom, and A. van Dam, "The Application Visualization System : A Computational Environment for Scientific Visualization," *IEEE Comp. Graphics and Appl.*, Vol. 9, No. 4, pp. 30–42, July 1989.

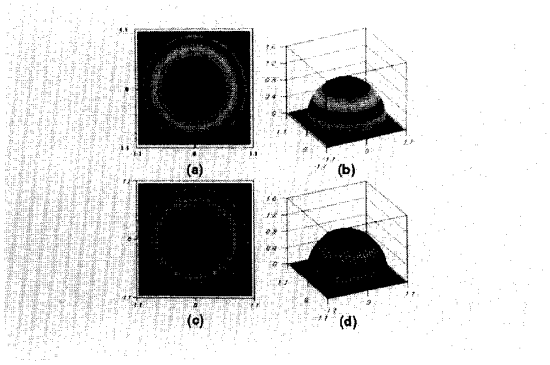

Figure 5: 2-D scalar field of equation (1): (a) straight-down pseudocolor map; (b) oblique view; (c) with grid lines; (d) oblique view. (PAW images (2).)

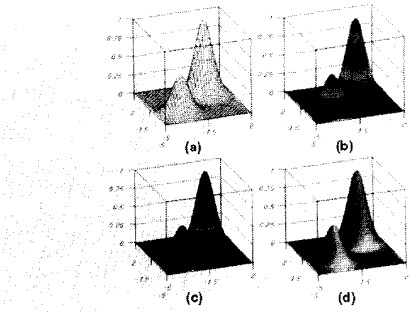

Figure 6: Oblique views of 2-D scalar field of equation (2): (a) grid lines alone; (b) pseudocolor alone; (c) both; (d) shaded 3-D rendering. (PAW images (2).)

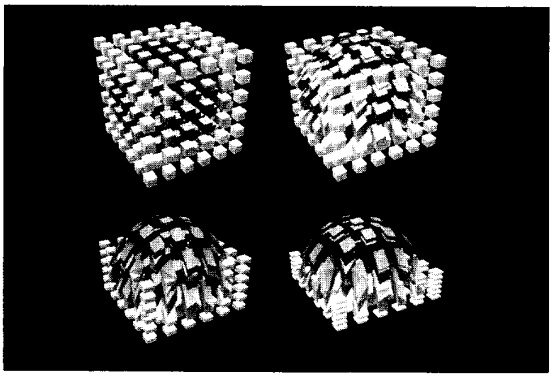

Figure 7: 4-D rotations of pseudocolored 4-D hemisphere. The color of each small cube remains fixed.

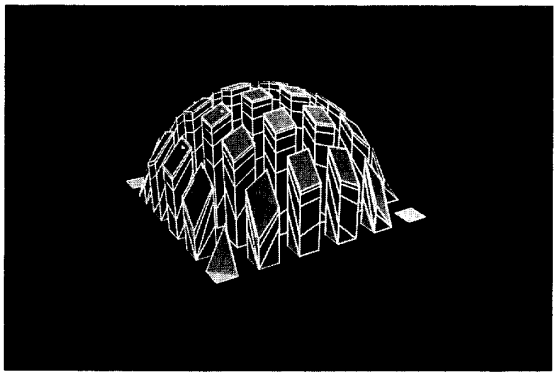

Figure 8: Oblique view of 4-D hemisphere with pseudocolor grid and hidden volume elimination.

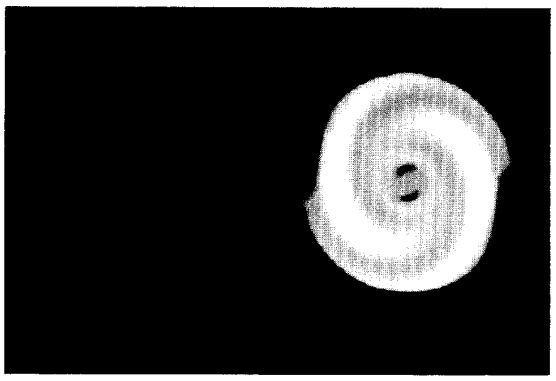

Figure 9: (Left) volume rendering of binary star density data using the AVS tracer* module. (Right) our 4-D shaded rendering of the data.

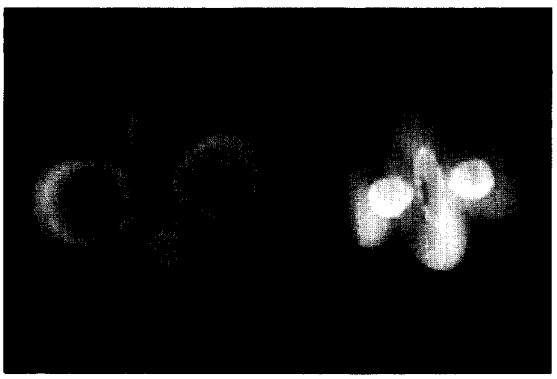

Figure 10: (Left) AVS volume rendering of the electron density field of a hydrogen molecule. (Right) our 4-D shaded rendering of the same data.

(See color plates, p. CP-11.)