in/
presented at: "Visual Data Exploration and Analysis," San Jose, CA, Feb. 1994; eds.: R.J. Moorhead, D.E. Silver, S.P. Uselton.

Oceanographic Visualization Interactive Research Tool (OVIRT)

Robert J. Moorhead II, Bernd Hamann, Cass Everitt, Steve Jones, Joel McAllister, and Jonathan Barlow

NSF Engineering Research Center for Computational Field Simulations
P.O. Box 6176
Mississippi State University, Miss. State, MS 39762

Abstract

The Oceanographic Visualization Interactive Research Tool (OVIRT) was developed to explore the utility of scalar field volume rendering in visualizing environmental ocean data and to extend some of the classical 2D oceanographic displays into a 3D visualization environment. It has five major visualization tools: cutting planes, minicubes, isosurfaces, sonic-surfaces, and direct volume rendering. The cutting planes tool provides three orthogonal cutting planes which can be interactively moved through the volume. The minicubes routine renders small cubes whose faces are shaded according to function value. The isosurface tool is conceptually similar to the well-known marching cubes technique. The sonic surfaces are an extension of the 2D surfaces which have been classically used to display acoustic propagation paths and inflection lines within the ocean. The surfaces delineate the extent and axis of the shallow and deep sound channels. The direct volume rendering (DVR) techniques give a global view of the data. Other features include the ability to overlay the shoreline and inlay the bathymetry. There are multiple colormaps, an automatic histogramming feature, a macro and scripting capability, a picking function, and the ability to display animations of DVR imagery. There is a network feature to allow computationally expensive functions to be executed on remote machines.

Introduction

To locate oceanographic features such as the geographic locations of ocean fronts and eddies, it is useful to analyze three-dimensional fields of sound velocity in the ocean. Traditional methodology involves graphical depictions of slices of the ocean at varying depths or vertical contours of sound speed or temperature values across a track of interest. The limitation of those techniques is that the analyst only views a limited portion of the operating area at once. To achieve a total view requires assimilating multiple executions of the display routine and mentally composing the results. Because of that limitation, a tool was developed that would volume render oceanographic data sets. The result of that effort is the Oceanographic Visualization Interactive Research Tool (OVIRT). The principle objective of OVIRT is to visualize the 3D ocean sound speed field as a volume. By doing so, one can determine the location of oceanographic features below the surface.

OVIRT uses a MOTIF graphical user interface. The rendering is done in GL, using mixed-mode programming. It has five major visualization tools: cutting planes, minicubes, isosurfaces, sonic-surfaces, and direct volume rendering. It is based on a uniform rectilinear grid in the x and y (latitude and longitude) directions with a non-uniform, but horizontally invariant, spacing in the z or depth dimension, i.e., the sample depths are not a function of latitude and longitude or a function the height of the water column.

Techniques

The cutting planes technique [10] allows the user to move each of the three orthogonal cutting planes through the volume by altering the value of an independent variable. The value may be incremented, decremented, or selected randomly. The incrementation/decrementation allows the user to step through the volume; the random selection allows rapid selection of a particular value, which is useful when the amount of data is sufficiently large to cause uncomfortably long rendering times if each slice is rendered sequentially.

Minicubes [10] allows the user to take a volume of data and extract small cubes in the volume which are uniformly spaced through the volume. Four or more functional values are then mapped onto the face of the cubes. Note that different scalar functions, e.g., temperature, pressure, salinity, could be rendered onto different faces. Computationally, the minicube extraction technique falls between cutting planes and direct volume rendering (i.e., ray casting). It actually lets you perceive the whole volume of data. You can see the whole volume at once or you can zoom into

0-8194-1473-5/94/$6.00

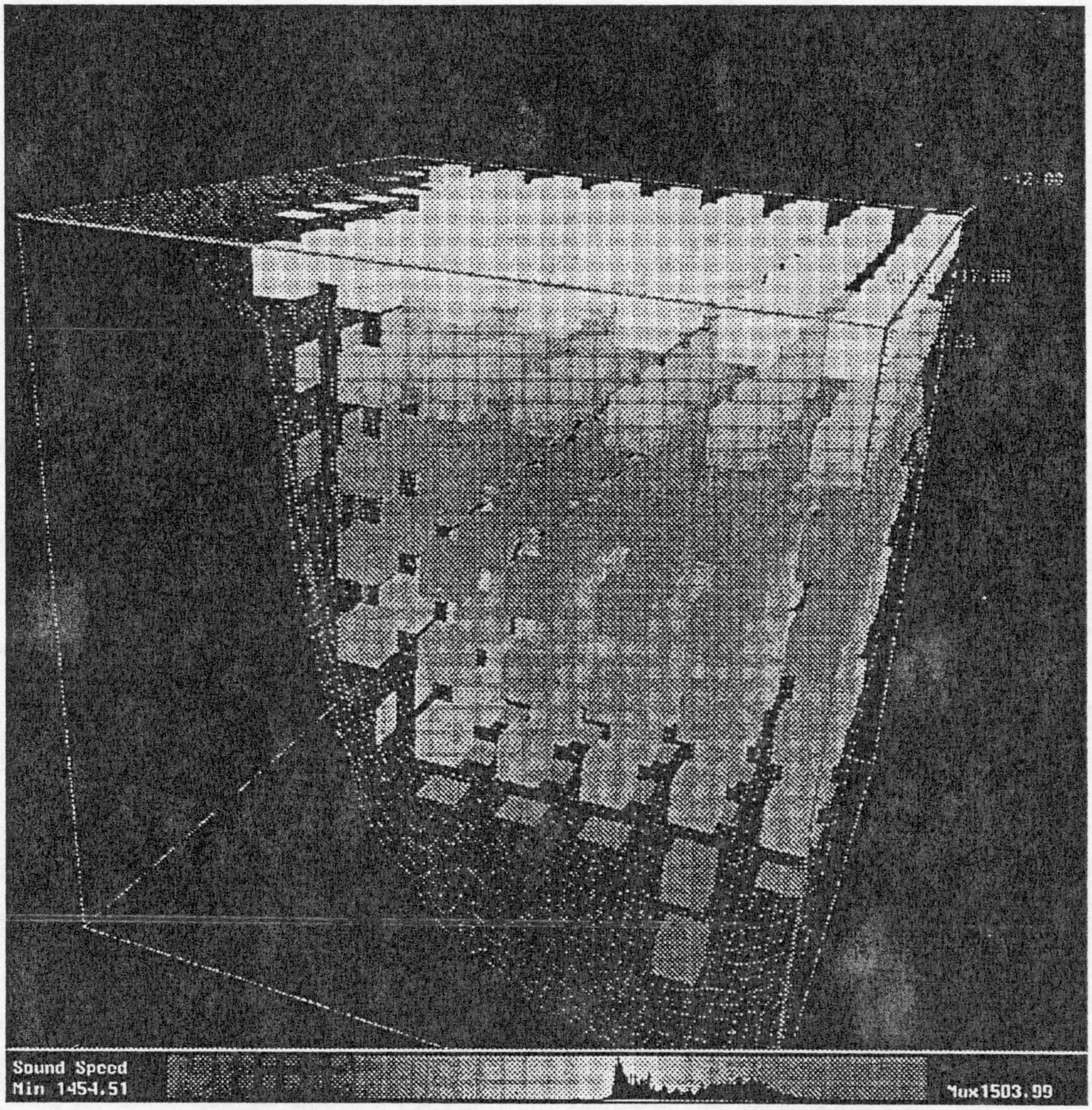

Figure 1. Minicubes technique with sound speed data visualized as 1000 equal-size cubes

the volume and look at a subvolume. "Cubes" are actually hexahedra with orthogonal edges and varying size faces. In Figure 1, a 1000 cubes (10 in each direction) have been selected, but due to the bathymetric clipping within the rectilinear volume, some of, or part of some of, the cubes have been removed from the view. The data has been re-sampled using tri-linear interpolation so that interpolated sample points form the vertices of the cubes. The bathymetry clipping has eliminated all of the bottom row in this view. You can see the bathymetry rendered as a wireframe. In Figure 2, once again a 1000 cubes have been selected. In this case the original non-uniform spacing in z is used to define the vertices of the cubes, creating cubes which increase in height as the depth increases, but which are more precise representations of the data input to the visualization toolkit, since it uses the original grid points. The density of the sampling in the datasets decreases with depth. With the tool, the user can zoom in, move around, and examine values in the center of the volume.

In an attempt to extend the visualization techniques with which the analyst is familiar, the 2D curves of sound speed versus depth, i.e., sound speed profiles, were extended to create a 3D visualization technique in which all the sound speed profiles for a particular latitude or longitude could be simultaneously rendered. This allowed the analyst to

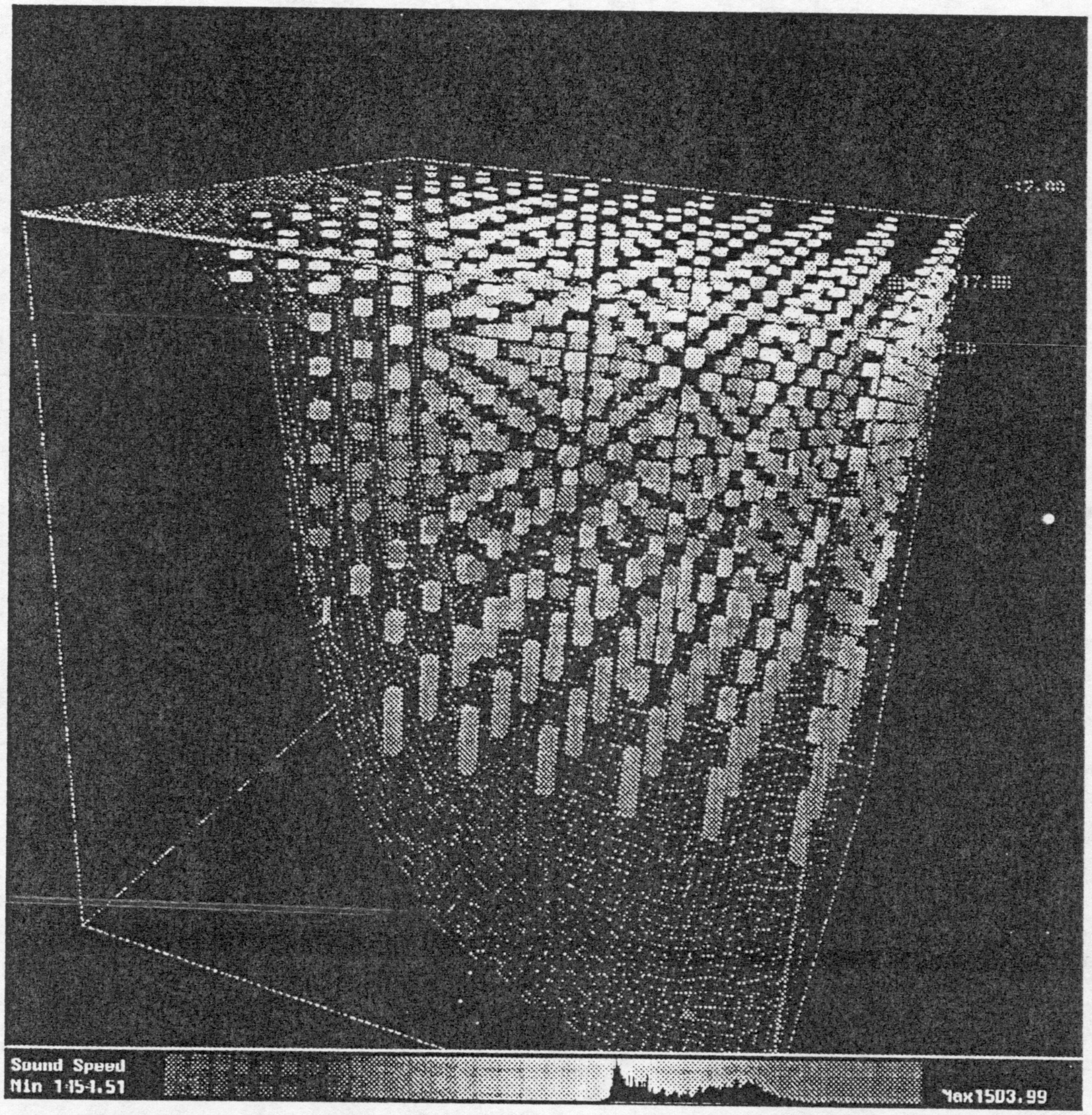

Figure 2. 1000 unequal-size minicubes using original sampled data as vertices

move from something familiar to something novel that gives more information in one view. We call the technique ***marching wiggles***, because as one sequences through various latitude or longitude values rendering the sonic profiles at each step, the curves appear to wiggle as they march along. Traditionally the analyst could look at the sonic profiles for either a longitudinal or a latitudinal slice, but he could not see the sonic profiles or sonic surfaces (deep sound channel, sonic layer depth, critical depth, etc.) inlaid in the water volume. The bathymetry clips the wiggles at the appropriate depth. The sound speed is rendered redundantly, in that both the horizontal distance from one face of the cube and the color of the curve indicate the sound speed. Although this tool was originally developed to allow the developers to verify the construction of the functional (sonic) surfaces, it has turned out to be very useful to the operational oceanographers who have used the toolkit because it does map to something with which they are very familiar.

There are certain surfaces that are of particular interest to an ocean acoustician that can be extracted from the collection of sound speed profiles. In general, they bound or are the axis of a sound channel, where a sound channel is a region in the water column where sound speed first decreases with depth to a local minimum value, and then increases. In order, from the ocean surface down, the surfaces of greatest interest are:

- shallow sound channel (SSC) axis,
- sonic layer depth (SLD),
- deep sound channel (DSC) axis, and
- critical depth (CD).

The SLD is defined as the depth at which maximum sound speed occurs in the layer above the thermocline, i.e., the upper levels of the water column. Thus, the sound speed profile at each latitude/longitude sample point is traversed from the top down to find the first relative maximum. In general, the next relative maximum is the CD, i.e., the sound speed is the same at the SLD and the CD at each x,y location. The SLD and the CD are each relative maxima and the deep sound channel axis is a relative minimum. The idea is as follows: If one puts an acoustic source somewhere below the SLD and above the CD, the sound rays will bounce between the two surfaces, and the sound will propagate more–or–less horizontally within the water volume. If the acoustic source is above the SLD, the sound rays will bend upward and usually penetrate the water surface. If the acoustic source is below the CD, the sound rays will bend downward [12]. In general, the sound speed increases as the depth increases, so the maximum sound speed would be at the bottom of the water mass. Figure 3 shows the DSC axis and the CD. Once again, the wireframe is the bathymetry of this section of the ocean. The wireframe offers two advantages: increased interactivity and the ability to see through the bathymetry, which is useful when there are seamounts. Note that SLD is a surface tessellation in which original grid points constitute the vertices, whereas the CD is constructed from depth–interpolated points at which the sound speed equals the sound speed on the SLD.

An isosurface routine is also included [6,9]. An isosurface is a surface constructed from sample points with the same scalar value. It has been found that shading isosurfaces is confusing to the analyst, since the color variations are due to a lighting model, not a mapping of the functional values. A mapping of the functional values into a colormap makes no sense, since all the functional values on the surface are the same. The shading helps the user perceive the depth or shape of the 3D object. Unfortunately, it is the only surface rendering technique in the toolkit that uses color for shading and not as a means for indicating functional values. The analyst can choose to render the isosurfaces flat shaded, transparent–flat shaded, Gouraud shaded, or transparent–Gouraud shaded. Flat shading uses the same color across each polygonal primitive, Gouraud shading linearly interpolates the color between vertices which makes the object look much smoother. The advantage of flat shading are the visibility of the vertices and faster rendering on most machines. The advantage of Gouraud shading is photorealistism, i.e., it "looks" better.

Finally, two direct volume rendering techniques were included, based on the work of Levoy [5] and Sabella [11], respectively. The idea behind these methods is that light is attenuated and refracted as it passes through a medium. If the light completely penetrates a medium, it means that it is transparent; if it does not penetrate, the medium is opaque. Ray–casting provides a better 3D impression, and when rotating the volume, it looks strikingly realistic. It allows the analyst to see "features" rather than isosurfaces, due to the fact that the entire volume is considered. Ray–casting is based on the idea that data lying deeper in the volume has more influence on the color than ones lying closer to the viewer. This is accomplished by using an exponentially decreasing weight function to accumulate the data along each ray. For more details, see [5,11] It was discovered that this simple opacity mapping, i.e., a monotonic mapping of sound speed (density, temperature) to opacity, is too simplistic. Such a visualization is presented in Figure 4, in which the sonic properties are admittedly not visible. Even on a high–resolution monitor utilizing all 3 color channels to convey information (using the Levoy technique), the sonic surfaces are only arguably detectable. More robust opacity mapping functions are necessary [3,4]. Also, rather robust data classifiers are needed. It does not suffice to simply cast rays through the whole water column at arbitrary viewing angles. If the sonic surfaces are to be adequately visualized, they must be "classified" in some way [3]. Good examples of successful volume rendering for environment data can be found in [7,8].

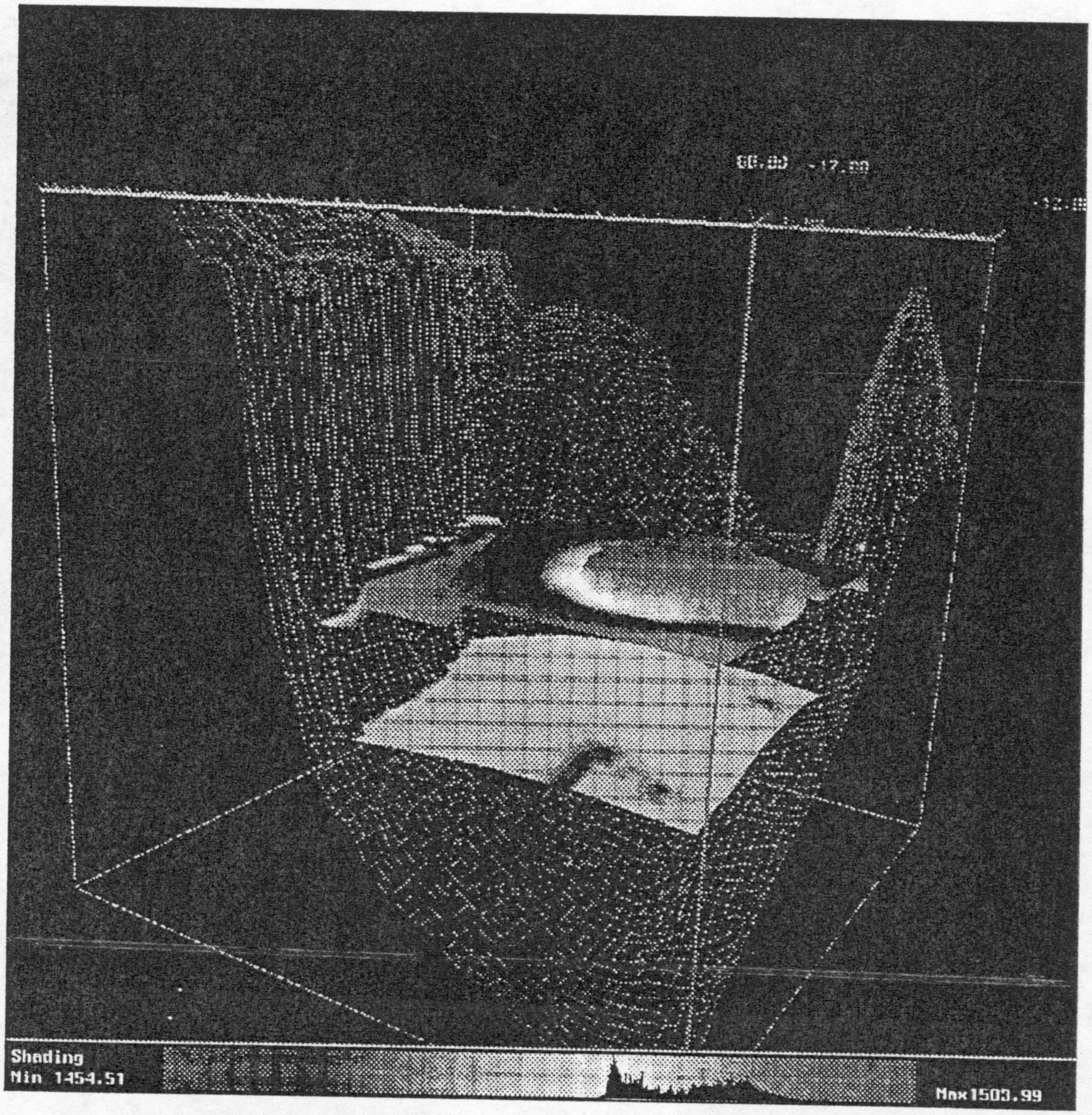

Figure 3. The Deep Sound Channel Axis (top) and Critical Depth (bottom) inlaid in the bathymetry

Qualitative and Functional Aspects

The toolkit includes 6 different colormaps (continuous spectral, banded spectral, red, green, blue, and gray); adding additional ones is simple. There is a macro capability which allows the user to create and execute macro commands, i.e., groups of commands. The toolkit allows the analyst to run applications remote, yet to visualize on a local workstation. Computationally complex modules have been ported to run remotely on the Sun SPARCStation 10, the Cray YMP, and various SGI machines. A feature which most users have found to be very useful is a histogram of the data overlaid on the color map. This allows the user to see the distribution of the data, since all the data points are not necessarily rendered at one time.

The toolkit reads data located on parallel planes in a fully populated rectilinear volume, i.e., independent of whether a sample point is within a landmass or within the water, there is a scalar value at each grid point. Thus, all the data is stored internally in a 3D array. A different database that has the bathymetry and/or coastline data is used to clip out the invalid values. Unfortunately, zero-level bathymetry and coastline data are not always the same because the

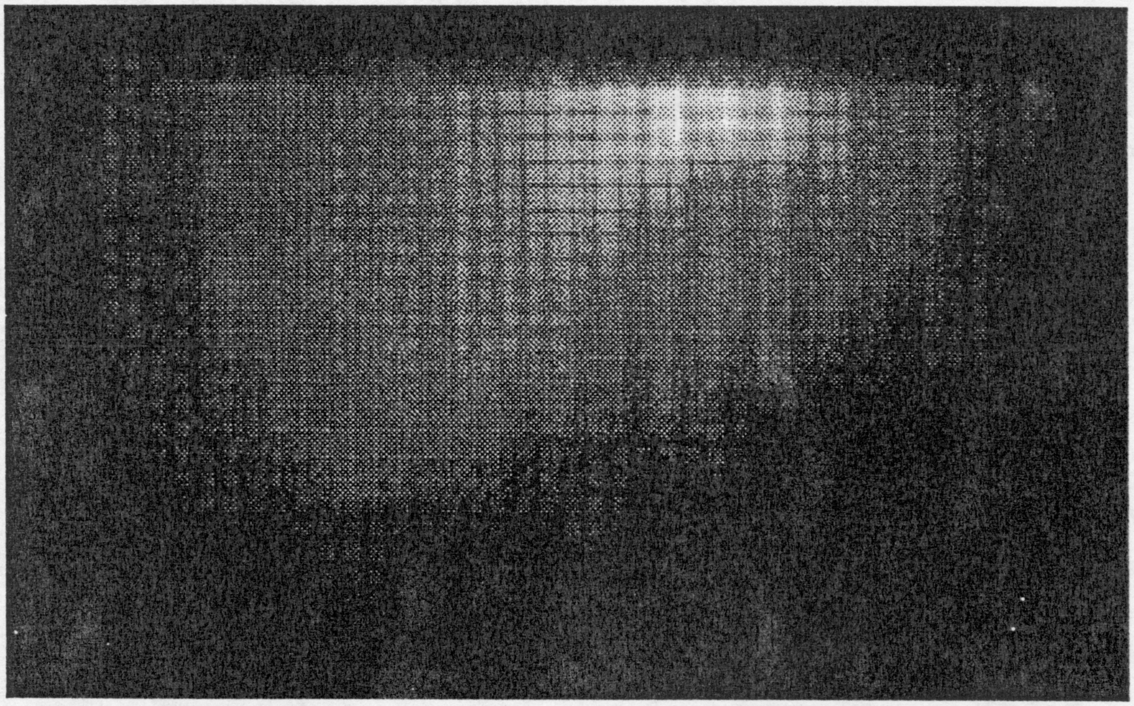

Figure 4. Direct Volume Rendering of the water volume using an exponentially decreasing weight function to accumulate the data along each ray

bathymetry has been measured and the coastline is defined by political entities Both bathymetry and coastline data are displayed in OVIRT. Other features include

- scripting and journal capabilities,
- interactive query within the viewport of both location and value via picking,
- multiple rendering modes for bathymetry, cutting planes, and sonic surfaces, and
- mouse-driven rotations, scaling, and translations.

Conclusions and Future Work

A new oceanographic analytical toolkit has been developed. It is simple enough for an operational oceanographer to use it in a tactical environment, yet powerful enough for a researcher to analyze and explore model results or experimental data. The toolkit is reasonably responsive and very functional.

As for further work, the sound channels need to be emphasized more by making the transfer functions depend on the gradient rather than the functional values. Adding acoustic ray-tracing[12] within the sound channels is envisioned. OVIRT would then allow the user to try various "what if" scenarios. What if the acoustic source is here? In what direction and how far do the sound rays propagate? Presently, OVIRT only visualizes scalar data; vector data visualization techniques need to be added, i.e., visualization methods for vector data, such as velocity.

References

1 Brodlie, K.W. et al (Editors), Scientific Visualization Techniques and Applications, Springer-Verlag, 1992.

2 Cline, H. E., Lorensen, W. E., Ludke, W., Crawford, C. R., and Teeter, B. C., "Two Algorithms for Three-Dimensional Reconstruction of Tomograms," *Medical Physics*, Vol. 15, No. 3, May-June, 1988, pp. 320-327.

3 Elvins, T., "A Survey of Algorithms for Volume Visualization," *Computer Graphics*, Volume 26, Number 3, August 1992, pp. 194-201.

4 Everitt, C., "Using Classification and Feature Detection in Splatting of Scientific Datasets," National Conference on Undergraduate Research, Kalamazoo, MI, April 14–16, 1994 (submitted).

5 Levoy, M., "Display of Surfaces from Volume Data", *IEEE Computer Graphics and Applications,* Volume 8, Number 3, May 1988, pp. 29–37.

6 Lorensen, W. E., and Cline, H. E., "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", *Computer Graphics,* Volume 21, Number 4, July 1987, pp. 163–169.

7 Max, N., Becker, B., and Crawfis, R., "Flow Volumes for Interactive Vector Field Visualization," *IEEE Visualization '93 Proceedings*, Oct. 1993, San Jose, CA, pp. 19–24.

8 Max, N., Crawfis, R., and Williams, D., "Visualization for Climate Modeling," *IEEE Computer Graphics & Applications,"* Vol. 13, No. 4, July 1993, pp. 34–40.

9 Nielson, G. M., and Hamann, B., "The Asymptotic Decider: Resolving the Ambiguity in Marching cubes," *IEEE Visualization '91 Proc.*, San Diego, CA, Oct. 1991, pp. 83–91.

10 Nielson, G. M., Foley, T. A., Hamann, B., and Lane, D. A., "Visualizing and Modeling Scattered Multivariate Data," *IEEE Computer Graphics & Applications*, Vol. 11, No. 3, 1991, pp. 47–55.

11 Sabella, P., "A Rendering Algorithm for Visualizing 3D Scalar Fields," *Computer Graphics,* Vol. 22, No. 4, August 1988, pp. 51–58.

12 Ziomek, L.J., and Polnicky, F.W., "The RRA Algorithm: Recursive Ray Acoustics for Three-Dimensional Speeds of Sound," *IEEE Journal of Oceanic Engineering,* Vol.18, No.1, January 1993, pp.25–30.