

Volume Rendering with Animation of Gulf Stream Currents

David Ehrens
Computer and Information Systems
Univ. of Massachusetts, Dartmouth

Ramprasad Balasubramanian
Computer and Information Systems
Univ. of Massachusetts, Dartmouth

Amit Tandon
Physics Department
Univ. of Massachusetts, Dartmouth

Jürgen Schulze
Department of Computer Science
Brown University

Abstract

This paper describes the novel use of volume rendering for the visualization of MICOM oceanographic data and the tracking of mesoscale eddies. We present methods for generating 3D volumetric data and describe how to use VIRVO to build an animation for MICOM Gulf Stream motion data. This put the data in a format that easily gives oceanographers the ability to visualize mesoscale eddies from any angle, including an undersea aspect, over any duration, and by combinations of spatial dimensions. One advantage of this technique is that only minimal processing of the oceanographic data is required. High or low density objects, corresponding to heat-laden currents, can be enhanced or masked by modifying a dynamic transfer function, also providing the ability to visualize cooler return waters. While the data format is relatively simple and the software is easily installed on a personal computer, the same animations can be viewed in an immersive virtual reality environment.

1. Introduction

Understanding the ocean through computer models is important and numerous models exist. Several, including MICOM, the Miami Isopycnic Coordination Ocean Model, and HYCOM, the HYbrid Coordinate Ocean Model, use an isopycnic (equal density) model that has been successful in modeling currents. Other isopycnic models include the Hallberg Isopycnal Model (HIM), the Navy Layered Ocean Model (NLOM), and the Parallel Oregon State University Model (POSUM).

The DYNAMO Project compared three ocean models and only isopycnic models predicted the Azores Current and a correct trajectory for the North Atlantic Current [1]. Oceanographers study the adiabatic properties of the sea, looking at variables such as heat, velocity, density and salinity. Visualizing synthetic variables, such as *heat index* [2], a function of velocity and temperature, can be difficult. The

velocity component itself is a function of Eastward and Poleward current movement, so visualization can be quite useful.

In this paper we describe how we used VIRVO [3], a volume rendering tool, to visualize Gulf Stream MICOM oceanographic data, mesoscale coherent structures like jets and eddies, by converting it into volume rendering format. Our paper is organized as follows: we describe prior work in 3D oceanographic visualization, we introduce methods we use to prepare 3D views, and show the results and options available for users to view the resulting datasets. Finally, we discuss possible enhancements to these techniques for extremely large datasets and for performing diagnostics on the data.

2. Previous Work

The earliest visualization tools were developed at the (National Center for Atmospheric Research (NCAR) in the 1960s. NCAR's Man-Computer Interactive Data Access System (McIDAS) and the AFOS (Automation of Field Operations) system from the National Weather Service date back to the 1970s. During the 1980s systems like PROFS concentrated on developing highly interactive applications.

Despite these advances, geoscience requirements for 3D visualization have always been challenging. Data collected hourly and over vast areas can easily overwhelm databases and conventional modeling and visualization tools. Both the atmosphere and ocean are relatively thin when compared with the earth's vast surface, and stretching of the z-axis is required to visualize geophysical phenomena. Viewing multiple attributes of geospatial data simultaneously requires associating each with either a color or transparency so that each may be viewed separately. Representing flow must often employ 2D techniques such as arrow plots, streamlines, advected particles, wire frame overlays, digital elevation models, color-mapping techniques, and quiver plots, but they may not easily adapt to 3D. In addition at-

ospheric layers and time steps must be added to models [4].

2.1. 3D Modeling Techniques

Recent computational projects involving global ocean models are numerous and span most of the ocean models. Although most ocean models are 3-dimensional, many visualization projects model current activity only in two dimensions. A number of researchers have developed 3D visualization tools, but we are particularly interested in 3D volumetric approaches which model the ocean as a solid with variations in density and temperature.

While the following is not intended to be a comprehensive survey, some of the efforts in 3D volume rendering visualization of oceanographic include cloud water density mapping of LAMPS data [5], template matching and time tracking applied to NOAA/Levitus ocean data [6], color mapping of volumetric data [7], POP and MICOM data [8], VRML for volume visualization [9], OVIRT [10], motion estimation in volumetric data [11], and ParVox [12]. Volume rendering applications in other domains, such as visualization of astrophysics [13] or atmospheric data have been explored. Most, however, require either servers, mainframes, supercomputers, special-purpose workstations, immersive virtual reality environments, or custom application software. None met our needs for an applications programming interface with an open architecture that runs on a desktop compactly, provides real-time rendering, time series, animation, and has immersive virtual reality and parallel processing options.

3. Motivation

Isopycnic ocean models represent the z -dimension by density, not layer thickness. Instead of predicting water density changes at certain depths, isopycnic models predict the depth at which densities change. These densities are also key to a number of predictive capabilities.

Mesoscale eddies and jets are believed to be important in poleward heat transport mechanisms. While numerical models quantify this by looking at heat flux components across sections of constant latitude, they do not track the lifetime of these eddies and jets. By tracking the eddies themselves and knowing isopycnic layer densities, velocity, and temperature, rather than estimating heat content in oceanic zones, better estimates of poleward heat transport can be directly measured.

Since our long-term goal is to study the the role of coherent 3D structures in poleward current movement, we need an appropriate 3D visualization tool that permits us to perform diagnostics while we analyze data. For instance, while individual eddies can be tracked, we would like to know

how accurate our eddy-recognition algorithms are. Combining diagnostics with visualization offers oceanographers the ability to gain insights into oceanic current behavior and also to eliminate false positive algorithmic identifications of eddies. Being able to view large datasets in an immersive virtual reality environment can enhance these advantages by providing the user with a larger field of view, better 3D perception, and interactive setting of clipping planes.

In evaluating VIRVO, we were looking for an API with an open architecture that is small, fast, runs on a desktop, does real-time rendering, supports time series, animation, can be used with immersive virtual reality, and can potentially process large data in parallel on commodity hardware.

4. Methodology

The MICOM data product we used is from ECMWF (European Centre for Medium-Range Weather Forecasts) data. Available variables are southward barotropic velocity, eastward barotropic velocity, barotropic pressure, mixed layer Montgomery potential, mass weighted average mixed layer sigma theta, mass weighted average southward velocity, mass weighted average eastward velocity, layer thickness in pressure units, mass weighted average layer temperature, and mass weighted average layer salinity.

MICOM ECMWF products also can provide wind stress vector, wind velocity, surface radiation, humidity, air temperature, and precipitation data. Heat flux is calculated from surface radiation, air temperature, specific humidity, wind speed and sea surface temperature. Fresh water flux is calculated from evaporation estimates, specific humidity and sea surface temperature.

One feature of the MICOM model is a top, *mixed* layer. This layer interacts with not only the other layers but the wind and solar system, hence the name. Oceanographers who use isopycnic models are interested in properties of these layers. We are particularly interested in thermal transport characteristics and the function of eddies and jets. Building a 3-dimensional representation of an area of interest, then having a visual framework to check our assumptions is invaluable.

4.1. Algorithm

MICOM's geographic area is North and Equatorial Atlantic (28 deg S to 70 deg N), and includes the Mediterranean Sea. Horizontal resolution of the model is 1/12th deg (roughly 6 km) and vertical resolution is 20 layers. We chose a subset of data 0 to 60 degrees Latitude and -95 to 16 degrees Longitude (see Fig. 1) and selected all 20 layers. Resolution of the data was 395 x 376 x 20. NCAR provided data for four consecutive years, 1979-1982. Readings are provided every three days, with a year consisting of approximately 120 individual datasets.

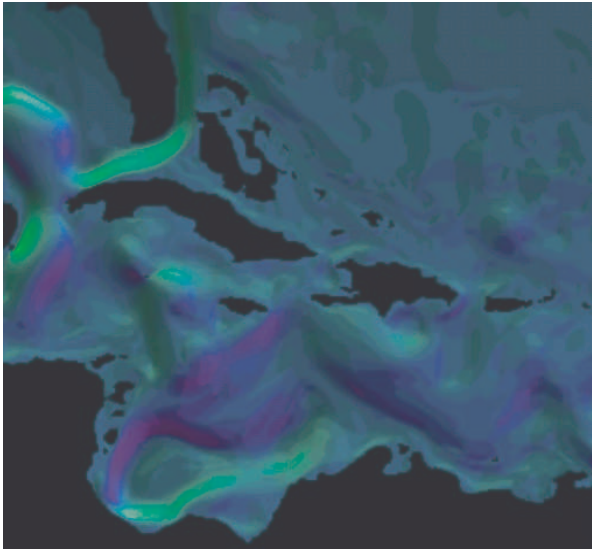


Figure 1: Eddy-rich region of interest (in 2D VIRVO view of Caribbean)

A year's data then contains roughly 360 million discrete data points, each with its own temperature, density, and two velocities. We considered both storage requirements and ways of representing this data in volume rendering software.

For 1979-1982, we received 488 files from NCAR, each one 7 megabytes and containing a single day's data. Data arrived in a proprietary Fortran format which we decoded into a binary format that could be read by either Matlab or another program we wrote to create VIRVO XVF files.

We are interested in the heat transport of currents. We define *heat index* [2] as

$$heat_index(i, j, l) = \frac{v(i, j, l) * u(i, j, l) * dp(i, j, l)}{P(i, l)}$$

where v , u , and dp are defined above, i and j are longitudinal and latitudinal indices, and l refers to the isopycnic layer. As we process MICOM layers, starting from layer 1 and going down in depth, we must convert density to depth, maintain a map of the current depth at a given Cartesian coordinate, and do this for each layer. To convert pressure to distance, we apply a gravitational constant:

$$km = \frac{dp(i, j)}{980.6/100000}$$

Fig. 2 illustrates the relationship between MICOM layers and VIRVO slices. As we process each layer, we map the current thickness of the layer to a number of slices. In our experiments, we compared the final depth at layer 20 to bathymetric data and verified this mapping.

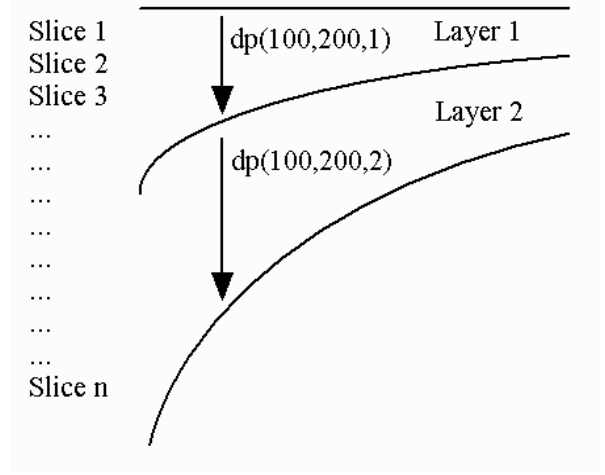


Figure 2: Mapping Layers to Slices

4.2. Scalar and RGBA

In VIRVO it is possible to store voxels as scalar values or as RGB or RGBA values. VIRVO uses regular (Cartesian) grids, as opposed to irregular or structured grids, which improves efficiency and speed and also simplifies programming. It was trivial to store the heat index as a scalar and the resulting image was visually identical to previous 2D Matlab images we had created. Stored as scalars, a single day's image was 3.6 megabytes.

In another experiment, we encoded MICOM data directly in RGBA attributes: R = dp, G = u, B = v, and A = temp. Each voxel requires 8 bytes. Compressed, each layer is 660 Kbytes, and 64 slices for a single day is over 40 megabytes. An animation of 5 gigabytes was therefore not practical but, by mapping MICOM's mixed layer to one slice, we were able to use VIRVO as a 2D visualization tool. 2D visualizations in RGBA run well on workstations and the animation is simple and spectacular.

Scalar datasets likewise were too large for most desktop systems. We could not load entire-year animations at full resolution and, even by reducing slices or downsampling the resolution (from 376x295 to 256x192) or animating half the number of frames, some compromises were necessary.

Since VView and VShell were not specifically designed for oceanographic research, we overlaid a transparent map of the coastlines in the first slice. On scalar views, it is easy to add Longitude and Latitude marks and grids as well as date and time stamps to identify the samples. In RGBA views, bathymetry can be added as one of the modalities.

With VView and VShell we were able to easily manipulate views of ocean sections. With volume rendering, it is possible to obtain a 3D view of an object from any angle, including, in our case, the ocean floor. Researchers are able to view data on commodity PCs with no special software

installations. While desktop viewing is probably sufficient for most researchers, we were also able to view exactly the same datasets in an immersive virtual reality environment (see Fig. 3).

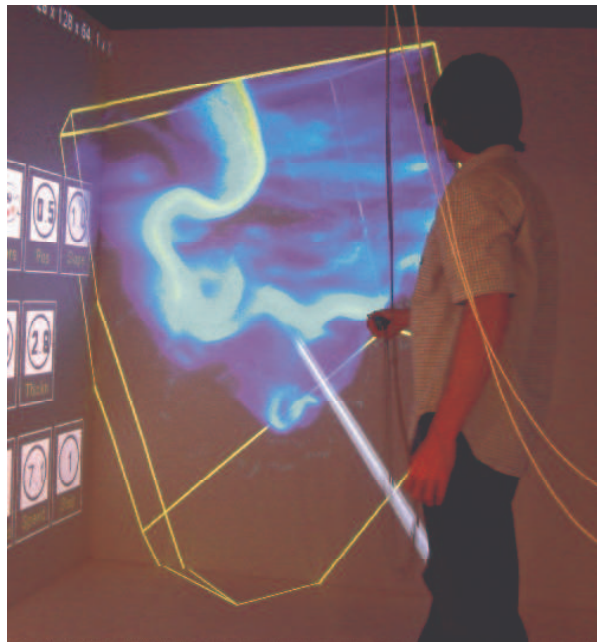


Figure 3: MODIS data viewed in the CAVE at Brown University

4.3. VIRVO

VIRVO (Virtual Reality Volume Rendering) is a set of volume rendering programs and an object-oriented API originally developed at the High Performance Computing Center at the University of Stuttgart in Germany [3]. It primarily uses OpenGL hardware primitives for direct rendering. However, the VIRVO architecture is open and rendering can be implemented by numerous means. Among other things, VIRVO can bypass hardware rendering limitations by using software rendering based on the Shear-Warp algorithm. This algorithm provides high performance and its rendering speed is not limited by hardware pixel fill-rate. MPI with commodity personal computers can be used to construct a parallel imaging system [14]. This technique has already been used for visualization of massive (80 gigavoxel!) astrophysics data [13]. It seemed equally promising for oceanographic research.

Either scalar or RGBA (red, green, blue, alpha) voxels comprise an image. Depending on the scheme used, an image consists of pixels made up of as few as a single byte or as many as 16 bytes. Scalar values are mapped to colors via lookup tables and defined by transfer functions, which can mask various intensities or define opacity. VIRVO images

consist of slices, layers of 2D views. Because of this design, VIRVO was extremely attractive because of the ease with which 3-dimensional representations of oceanographic data could be built. Since the MICOM model consists of 20 layers, VIRVO could easily permit us to map the layers to an equal or greater number of slices.

4.4. VIRVO Features

VIRVO features include transfer functions, transparency and masking, 3D rotation, interpolation, time sequences, and animation. It is possible to choose any 3D section of the ocean and view it from any conceivable angle, including the ocean floor. VIRVO has been extensively used in the medical [15], astrophysics [13], and materials sciences [16], but until recently not for the geosciences. We used VIRVO's built-in visualization tools, VView and VShell.

4.5. VShell

VShell is a volumetric editor which permits custom transfer functions and views to be created (see Fig. 4).

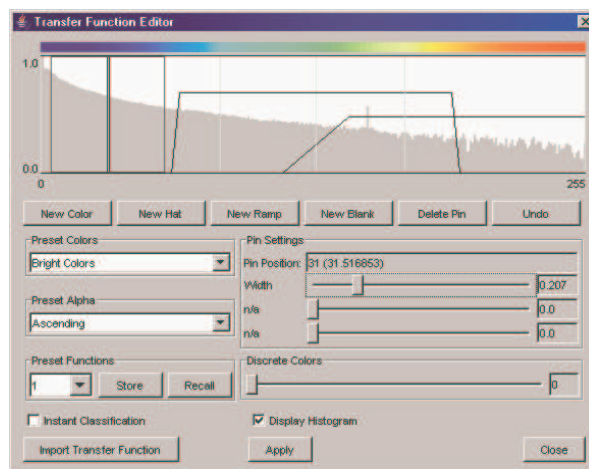


Figure 4: VShell Transfer Function

Transfer functions control how the voxels are displayed. To generalize, alpha pins (ramps) control intensity, trapezoidal functions (hats) influence contrast, and blanks determine opacity. Distances between pins create color gradients. By manipulating combinations of these transfer functions, users can focus on structures of interest or can suppress non-interesting phenomena. For example, in examining eddies, transfer functions permit us to emphasize either the warm Gulf Stream waters or the cooler returns.

VShell also provides a slice viewer, which permits the user to "dive" down to any isopycnic layer (see Fig. 5).

Slices of any dimensions can be examined (see Fig. 6):

Users can also use the transfer functions to build custom color maps.

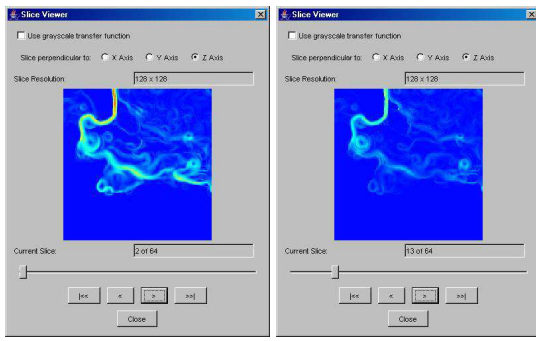


Figure 5: Isopycnic Layers 2 and 13

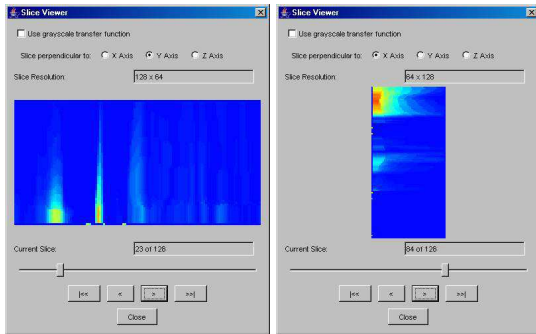


Figure 6: XZ Slice and YZ Slices

4.6. VView

VView is a volumetric viewer which permits the user to load a visualization, perform "canned" transfer functions (threshold minimization) and to animate a sequence of frames (see Fig. 7).

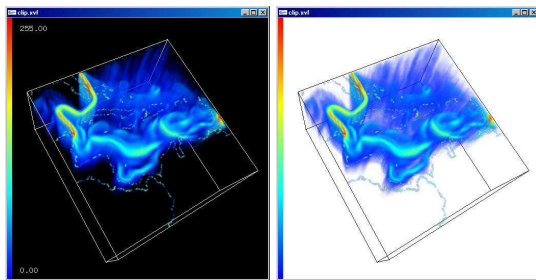


Figure 7: Using VView

After applying transfer functions, structures of interest can be masked or enhanced. In (see Fig. 8) on the left we have enhanced the warm Gulf Stream and on the right we mask the Gulf Stream and display instead the cooler return waters.

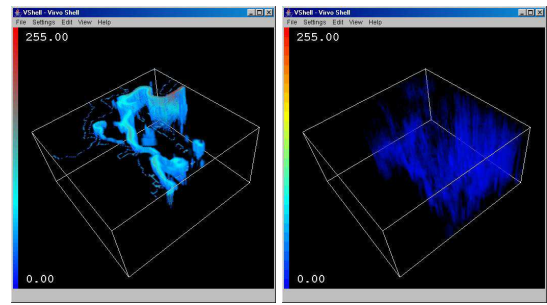


Figure 8: Applying Transfer Functions

5. Conclusion

We have presented a method of visualizing MICOM data using the VIRVO volume rendering API. Volume rendering is an intuitive approach to representing 3-dimensional data. Data is easily converted for visualization and user software requires minimal hardware and software resources. MICOM data can be rendered easily and diagnostics and other surfaces can be added to the volumetric object. Immersive virtual reality and parallelization are current options with VIRVO. Using VIRVO, even desktop PCs can host small, fast, easily-prepared and easy-to-use oceanographic visualizations.

Future work will involve parallelization of image composition, permitting us to tackle oceanographic datasets larger than those that can currently be visualized on the desktop, and adding eddy tracking algorithms adapted from 2D versions.

Acknowledgments

Thanks to Steve Worley at the National Center for Atmospheric Research, who provided us with the Gulf Stream data.

References

- [1] A. C. (ed.) and O. U. Staff, *Ocean Circulation, 2nd Ed.* Elsevier Science and Technology Books, 2001.
- [2] R. Balasubramanian, A. Tandon, B. John, and V. Sood, "Detecting and tracking of mesoscale oceanic features in the miami isopycnic circulation ocean model," in *Proceedings of the 3rd IASTED Computer Graphics and Imaging Conference*. International Association of Science and Technology for Development (IASTED), 2003.
- [3] S. W. J. Schulze-Doebold, U. Wössner and U. Lang., "Volume rendering in a virtual environment," in *Proceedings of 5th IPTW and Eurographics Virtual Environments*. Springer Verlag, 2001.

- [4] B. Hibbard, M. Böttinger, M. Schultz, and J. Biercamp, "Visualization in earth system science," *SIGGRAPH Computer Graphics*, vol. 36, no. 4, pp. 5–9, 2002.
- [5] W. L. Hibbard and D. A. Santek, "Visualizing large data sets in the earth sciences," *Computer*, vol. 22, no. 8, pp. 53–57, 1989.
- [6] D. Silver and X. Wang, "Tracking scalar features in unstructured datasets," in *Proceedings of the conference on Visualization '98*. IEEE Computer Society Press, 1998, pp. 79–86.
- [7] P. Crossno, E. Angel, and D. Munich, "Case study: visualizing ocean currents with color and dithering," in *Proceedings of the IEEE 2001 symposium on parallel and large-data visualization and graphics*. IEEE Press, 2001, pp. 37–40.
- [8] P. Gruzinskas, A. Haas, L. Goon, and M. Adams, "Data mining ocean model output at the naval oceanographic office major shared resource center," in *MTS/IEEE Oceans 2002 Conference*. IEEE, 2002.
- [9] J. Behr and M. Alexa, "Volume visualization in VRML," in *Proceedings of the sixth international conference on 3D Web technology*. ACM Press, 2001, pp. 23–27.
- [10] R. J. Moorhead and Z. Zhu, "Extracting and visualizing ocean eddies in time-varying flow fields," pp. 206–211, Sept. 1995.
- [11] W. de Leeuw and R. van Liere, "Bm3d: motion estimation in time dependent volume data," in *Proceedings of the conference on Visualization '02*. IEEE Computer Society, 2002, pp. 427–434.
- [12] P. P. Li, "Supercomputing visualization for earth science datasets," in *Earth Science Technology Conference*. Earth Science Technology Office (NASA), 2002.
- [13] B. Beeson, D. G. Barnes, and P. D. Bourke, "A distributed-data implementation of the perspective shear-warp volume rendering algorithm for visualisation of large astronomical cubes," *Publications of the Astronomical Society of Australia*, vol. 20, no. 3, pp. 300–313, 2003.
- [14] J. P. Schulze and U. Lang, "The parallelization of the perspective shear-warp volume rendering algorithm," in *Proceedings of the Fourth Eurographics Workshop on Parallel Graphics and Visualization*. Eurographics Association, 2002, pp. 61–69.
- [15] J. Kniss, J. Schulze, U. Wössner, P. Winkler, U. Lang, and C. Hansen, "Medical applications of multi-field volume rendering and vr techniques," in *Proceedings of the Joint Eurographics-IEEE TCVG Symposium on Visualization*. Eurographics Association, 2004, pp. 249–254.
- [16] G. Schaaf, F. Gähler, C. Kohler, U. Koschella, N. Resta, J. Roth, and C. R. and Hans-Rainer Trebin, "Simulation of dislocations in icosahedral quasicrystals with IMD, Tech. Rep. Technical Report No. 172, May 2001.