Visualization of Water Quality Data for the Chesapeake Bay

Adam B. Forgang, Bernd Hamann

Computer Graphics Laboratory
Department of Computer Science
University of California, Davis[1]

Carl F. Cerco

Environmental Laboratory
U.S. Army Corps of Engineers
Waterways Experiment Station[2]

Abstract

We discuss a visualization system for the comparison of simulated and measured water quality. The system extends SCIRT (Site Characterization Interactive Research Toolkit), an interactive system originally developed at the NSF Engineering Research Center for Computational Field Simulation at Mississippi State University.

1 Introduction

Management of environmental quality requires analyses of complex processes which often defy direct insight. The field of "environmental modeling" has evolved to aid in the analysis of complex air and water pollution problems.

A numerical model should yield output that agrees with observations. Comparative visualization of observations and model output presents multiple challenges due to the fact that the observations, sparsely scattered in space and time, generally do not coincide with the locations and time steps for which model output is given. We do not re-sample model output and observations onto one grid; the necessary interpolation might lead to false impressions when comparing model results and observations.

An ongoing study of the Chesapeake Bay presents an ideal study for research in 3D visualization of model-data comparisons. The model [2] simulates 22 environmental parameters, e.g., salinity, temperature, and dissolved oxygen. Computations are made on a 3D grid containing approximately 4,000 grid cells. Typical simulations encompass one year and produce approximately 17,000,000 computations of each variable.

2 Visualization Software

We have extended SCIRT [1] for model-data comparisons. SCIRT assumed that data sets were given on structured, rectilinear grids, but it has been extended to support curvilinear coordinates. Visualization techniques include *Cutting Planes*, *Minicubes*, and *Point Clouds* [4]. SCIRT creates look-up tables for various color maps during initialization

[1]Department of Computer Science, University of California, Davis, CA 95616-8562. Email: {forgang,hamann}@cs.ucdavis.edu

[2]Waterways Experiment Station, Vicksburg, MS 39180-6199. Email: cerco@homer.wes.army.mil
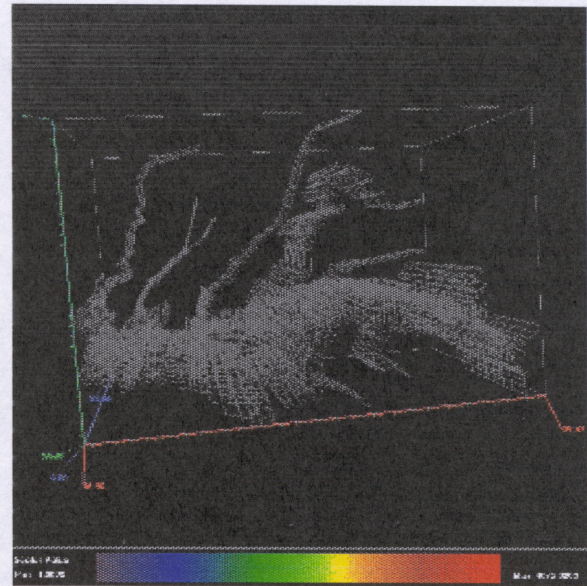
Figure 1: Wireframe of the Chesapeake Bay grid.

and function values are linearly interpolated. The histogram of the data is superimposed on the color map.

Whereas parameter values in the original SCIRT system are associated with grid points, the parameter values for the Chesapeake Bay data are associated with grid cells (hexahedral volumes). Therefore, all previous techniques have been modified to support cell-centered data.

3 Underlying Structure

The model output is given for curvilinear, hexahedral cells C_{ijk} (Fig. 1). A cell's data, associated with its centroid, is a vector-valued parameter F_{ijkl} associated with C_{ijk} and time step t_l (in Julian days). The coordinates are longitude, latitude, and depth. Only one of the constituents of F_{ijkl} is shown at time step τ.

The observations are scattered throughout the Bay for specific times. Parameters are measured at various depth levels along a probe specified by a longitude-latitude pair and time value. A "probe file" specifies one of the constituents of a vector-valued parameter.

4 Probe Files and Preprocessing Steps

When a model output file for time step τ is read, the corresponding constituent's "probe file" is read in. Only probes lying in the user-specified time interval $[\tau - \epsilon, \tau + \epsilon]$ are considered.

After the probe files are read, one determines the grid cells containing probe data based on the cell's Jacobian [3]. When a cell is found, its index is stored with the probe datum and the datum's associated (u, v, w) parameter tuple is used to compute a weight based on spatial distance.

5 Rendering Techniques

There are three principal techniques for rendering the probes: displaying probe outlines, displaying points along a probe, and displaying cells containing at least one probe datum and their observed, weighted data. One can combine these techniques.

5.1 Probe Outlines

The probe outlines are squares swept in the depth "direction" for a particular longitude/latitude location. The square is swept from the probe's minimum to maximum depth with a cross section at each datum's depth.

Since the time steps used in the numerical simulation do not agree with the times associated with observations, we use a color scheme to indicate "time distance" when rendering model output and observations simultaneously. Probes with time values coinciding with a model's time step τ are white. Observations collected before τ have a higher blue component; those collected after τ have a higher red component. The colors range from white to blue (or red) using a banded color map.

5.2 Probe Points

The probe points are rendered as cubes with color representing parameter values. A white line is drawn through the cubes to indicate connectivity.

Some probes have a longitude/latitude measurement that does not lie inside any cell data grid approximating the Bay's shape. Furthermore, some probe points have a depth value greater than that of the numerical model. In order to show this inconsistency, we have implemented a scheme that displays only those points lying within a cell (Fig. 2).

Another method shows the probe points as well as the outlines. This method aids the user in determining which probe corresponds to which time step (Fig. 3).

5.3 Cell Probes

Whereas the previous techniques focus on displaying the probes, this technique is better suited to allow comparison between probe data and model output. A cell containing at least one probe datum is rendered as a wireframe, colored with respect to its simulated data value and the colormap. The probe data in a cell are weighted and a weighted average is displayed at the centroid of the cell; its size and shape is determined by the number of points within the cell and its color is determined by the weighted average (Fig. 4).
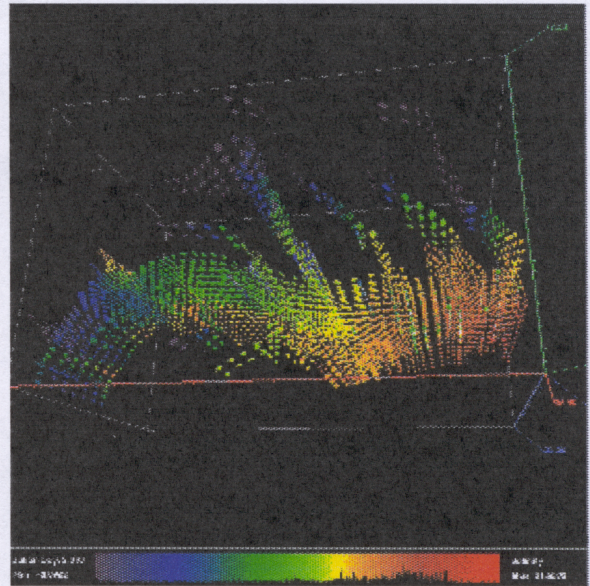

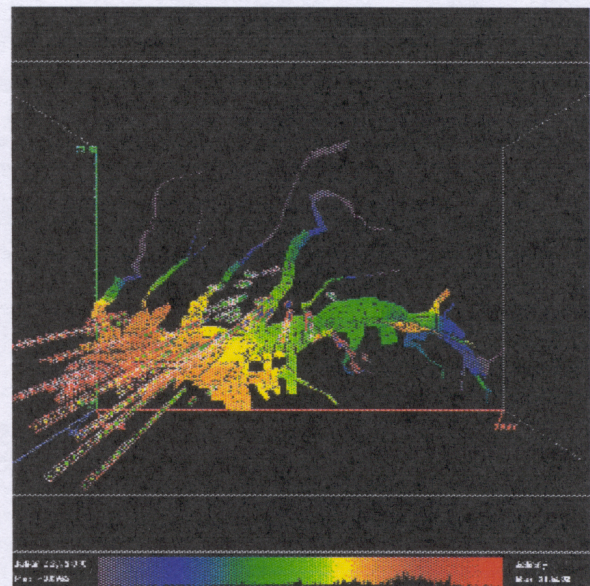Figure 2: Scaled Point Clouds visualized with clipped probe points.

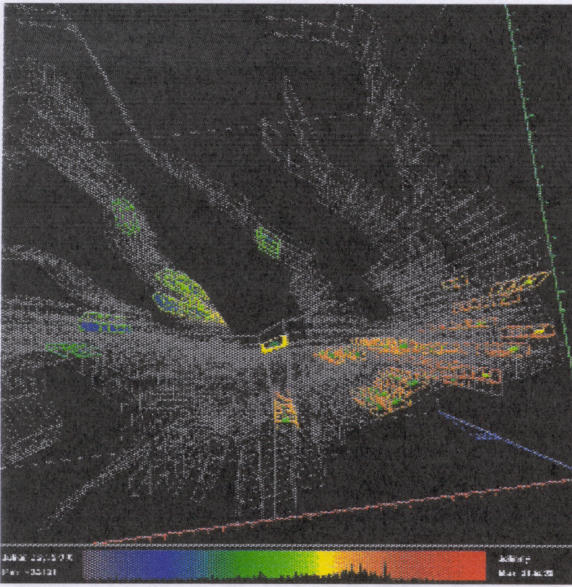Figure 3: Cutting Planes with probe points and probe outlines.

Figure 4: Wireframe of Chesapeake Bay with cell probes (weighted with respect to time).

5.3.1 Probe Data Shape and Size

There may be many points that lie within a single cell. Since it is important to know how many probe data are used to compute the average over a cell, this number is indicated. We use icons [5] to do so since the color is dedicated to the function value. For cells containing less than six points, we use flat-shaded hexahedra, scaled with respect to the number of points. For cells containing between six and ten points, we use scaled octahedra. For cells containing more than ten points, we use scaled, hollow cylinders.

5.3.2 Function values as weighted averages

Since there can be multiple probe data in a single cell, function averaging schemes have been developed. We use three weighting schemes: averaging, temporal weighting, and spatial weighting. Fig. 5a and 5b illustrate the first two schemes. The first scheme simply averages the data, i.e.,

$$\mathbf{F}_{average} = \frac{\sum_{i=0}^{k} f_i}{k+1},$$

considering all data in the time interval $[\tau - \epsilon, \tau + \epsilon]$.

The second scheme weights the data with respect to its distance from the time step, i.e.,

$$\mathbf{F}_{average} = \frac{\sum_{i=0}^{k} w_i f_i}{\sum_{i=0}^{k} w_i},$$

considering all data in the time interval $[\tau - \epsilon, \tau + \epsilon]$ and linearly decreasing weights w_i.

The third scheme is a 3D analog of the second scheme, where a datum at the centroid of a cube has weight one and data at the cube's vertices have weight zero (using a trilinear weight variation).
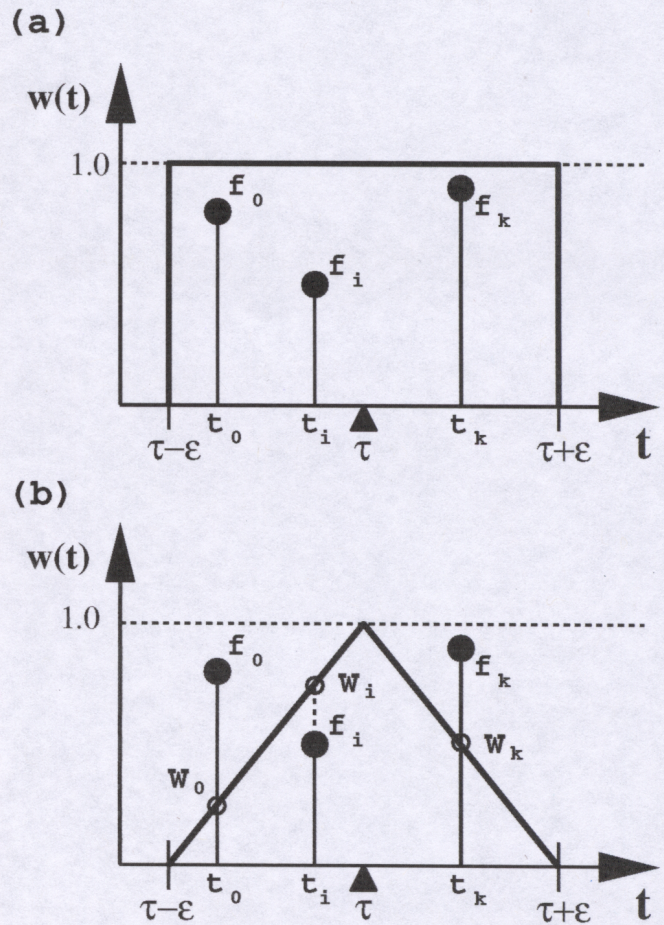

Figure 5: (a) Averaging using a finite "time window." (b) Weighting using hat function and finite "time window."
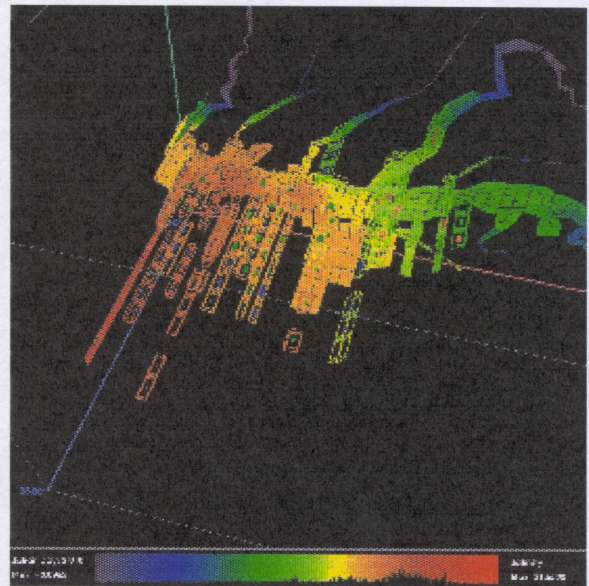

Figure 6: Cutting Planes and Difference Method (probes weighted with respect to position).
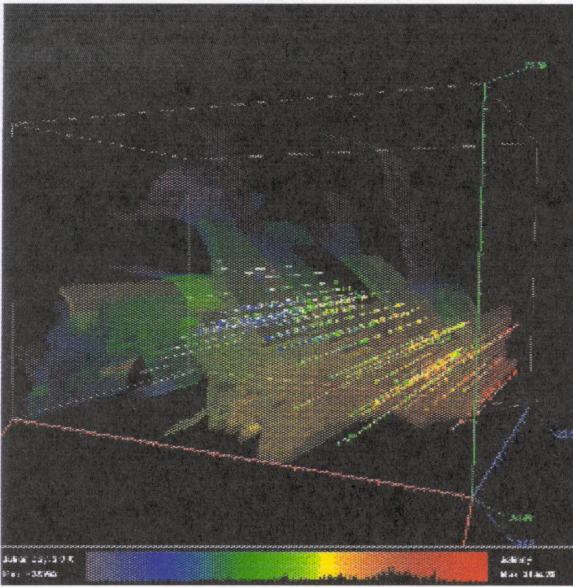
Figure 7: Transparent Minicubes visualized with probe points.

5.3.3 The Difference Method

Rather than letting the icon's color represent the function value, we let it represent the (absolute) difference. We compute the difference between the cell-averaged observational data and the model output, divided by the maximum difference. The great advantage of this rendering method is that the user can clearly see the regions where model output and observations strongly agree or disagree (Fig. 6).

6 Conclusions

All of the described methods can be used concurrently with *Cutting Planes* and *Volume Rendering* techniques. *Cutting Planes* works well with any method. Volume rendering techniques (*Minicubes* and *Point Clouds*) tend to obscure the probes; this can be rectified by using transparency (Fig. 7) or by reducing the size of the volume primitives. Generally, only a few files are read in at a time so the resulting picture is not too crowded and the probes are close (with respect to time) to the time step.

This program, developed on an SGI Indigo^2 Extreme workstation with a 150 MHz processor and 64 Mbytes of main memory, renders the data in real time, considering all rendering methods.

7 Future Work

Observed data might actually deviate a lot from a few, isolated simulated values. Nevertheless, due to the nature of averaging, this might not be evident in a rendering, thus giving a false impression of agreement. Therefore, we plan to add more comparitive methods, such as displaying the maximum absolute deviation of simulated values from known probe data, or the interval that the probe data in a single cell range over.

References

[1] Barlow, J. H. (faculty advisor: Robert J. Moorhead II), "Scientific Visualization of Soil Content Data," *Proc. National Conference on Undergraduate Research* (Kalamazoo, MI, April 14-16, 1994), pp. 1-4.

[2] Cerco, C. and Cole, T., "Three-dimensional eutrophication model of Chesapeake Bay," *Journal of Environmental Engineering* 119(6), pp. 1006-1025.

[3] Hamann, B., Wu, D. and Moorhead, R.J., "On Particle Path Generation Based on Quadrilinear Interpoloation and Bernstein-Bezier Polynomials," *IEEE Transactions on Visualization and Computer Graphics* 1(3), 1995, pp. 210-217.

[4] Nielson, G.M, Foley, T.A., Hamann, B. and Lane, D.A., "Visualizaing and Modeling Scattered Multivariate Data," *IEEE Computer Graphics & Applications* 11(3), 1991, pp. 47-55.

[5] Post, F.J., van Walsum, T., Post, F.H. and Silver, D., "Iconic Techniques for Feature Visualization," *Proc. IEEE Visualization '95* (San Diego, CA, Oct. 1995), pp. 288-295.