Interactive, Web-Based Three-Dimensional Visualizations of Operational Mesoscale Weather Models

Lloyd A. Treinish – IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA

lloydt@us.ibm.com http://www.research.ibm.com/people/l/lloydt

1. Introduction

Visualization is critical to the effective analysis and assessment of data generated by numerical weather prediction. In that regard, our previous work has discussed methods of appropriate mapping of user goals to the design of pictorial content by considering both the underlying data characteristics and the (human) perception of the visualization (*Treinish*, 1999). These have been effective in the development of inter-active applications utilizing either workstation- or PC/game-class three-dimensional graphics systems with sufficient bandwidth for timely access to the model data. When remote access to the visualizations is required the limited bandwidth becomes the primary bottleneck since reasonably current desktop systems have sufficient capacity to enable interactive visualization of model data at typical grid resolutions.

Popular approaches to web-based dissemination in operational environments are problematic as they impose enough compromises in time for access, display fidelity or interactivity to minimize effectiveness. Some utilize a standard workstation-based application with raw data in a compressed and/or sampled fashion to enable faster transmission. But such data reduction eliminates critical information from mesoscale weather models operating at cloud-scale resolution. Others utilize web browsers directly by providing static images or flip-book animations, often presented via Javascript-based players. Two additional problems are raised by this approach when used alone. The first is the need to heavily sample the images to reduce their size and number to be downloaded. This creates a gross mismatch in sampling to the time step used in the simulation, implying that available images are likely to miss important results. The second is the tendency to provide visualizations of most of the variables computed by or post-processed from the model. This can create a situation where users may have difficulty finding images of relevance, particularly if their application is not analysis of numerical weather prediction results. In addition, many of these visualizations may never be used, resulting in wasted computational resources. In some cases, the latter problem has been partially addressed by enabling products to be generated on demand, which has not been considered herein. In this effort, it has been assumed that there are a large number of clients using a relatively limited web server. Hence, the focus is on richer static content that is refreshed with each model run as an alter-native approach for use with operational results.

2. Approach and results

As we have presented before, the resolution of the visualization must match that of the scale of the model to build usable products that are perceptually and scientifically coherent (*Treinish*, 1999). The choice of realization geometry is also affected by the resolution of the data so that perceptual artifacts do not dominate the presentation, especially in animation (*Treinish*, 2000). Web-based dissemination exacerbates this situation. Therefore, abstraction and compression are introduced to capture sufficient content and reduce underlying bulk, respectively, to compensate for this situation.

First, a set of visualizations are presented as an interactive, three-dimensional spreadsheet, although such a paradigm is not new. Previously, it has been utilized for general visualization (*Jankun-Kelly and Ma*, 2000), images (*Hasler et al*, 1994) and three-dimensional interaction (*Hibbard*, 2001; *Rueden*, 2001) primarily to support data analysis by facilitating the comparison of different data sets or ensemble members. Instead, consider the spreadsheet as an abstraction of a model run. Thus, the rows and columns are organized at a high level (e.g., meteorological characteristics vs. model features) to simplify finding relevant visualizations. An example from an operational web site is shown in Figure 1. This spreadsheet becomes a meta-representation of the model and serves as an index for other visualizations and interactions.

This concept is implemented with a customized version of the Regional Atmospheric Modeling System (RAMS) (*Pielke et al*, 1992). Currently, two 24-hour forecasts are produced each day on a 3-way nested configuration of 62x62x31 at 16, 4 and 1 km resolution focused on New York City. Each model run requires about two hours of compute time on twenty-four 375 MHz Power3 processors of an IBM RS/6000 SP. Among the enhancements to RAMS is a suite of visualization tools, which also supports dissemination via web browsers as discussed herein. This complete system has been dubbed "Deep Thunder" (see http://www.research.ibm.com/ weather/NY/NY.html for more information).

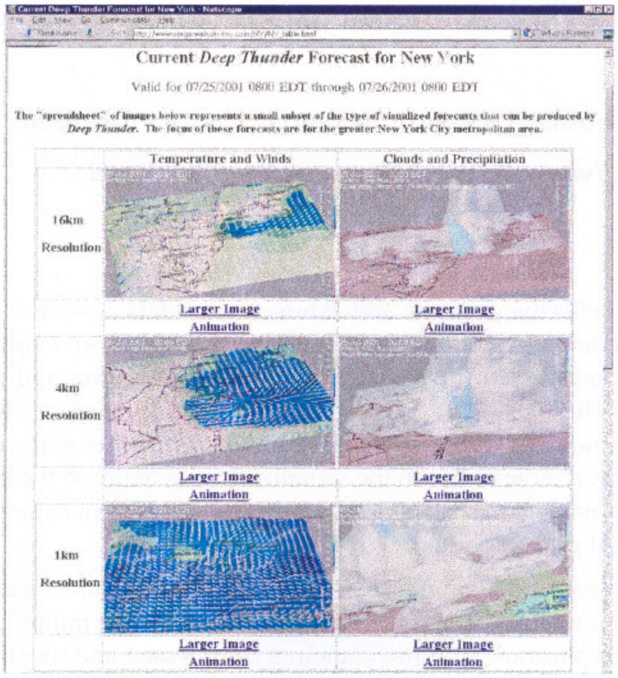

Fig. 1: Web-based 3D image spreadsheet for browse products.

2.1 *Browse products*

Each cell in the spreadsheet shown in Figure 1 provides a simple interactive scene generated from one time step of the model. Since the focus is on a high-level representation, the visualizations are designed to be qualitative for browsing (Class III from *Treinish*, 1999). Each cell shows different variables (columns) at different resolutions (rows).

The left column focuses on atmospheric motion and dynamics by illustrating surface temperature and winds. Each image contains a shaded terrain surface that is colored by contour bands of temperature (°F), following the scale to the upper right. During windy winter days, contour bands of wind chill are shown instead while contour bands of heat index are shown during humid summer days. The terrain is overlaid with maps of coastlines, county and state boundaries and rivers. Individual landmarks and cities are shown by name with the predicted temperature for that time step. The map is also overlaid with arrows corresponding to forecasted winds. The arrows point in the direction of the wind and are colored by speed following the scale to the lower right.

The right column focuses on moisture. Each image contains a terrain map in a three-dimensional scene with predicted clouds. The clouds are shown as a translucent white (boundary) surface derived from a threshold of total cloud water density (liquid and ice) of 10^{-4} kg water/kg air. If the model predicts convective activity such as thunderstorms, then a translucent cyan surface of predicted reflectivities may be visible within the clouds. The region within this cyan surface corresponds to where precipitation is forming (e.g., rain shafts) and where any storm activity would be the most severe. The terrain map is a shaded surface that is colored by contour bands of total precipitation, following the scale to the upper right. If the model predicts no precipitation then a similar visualization of humidity will be shown instead. If areas where precipitation is forecasted are sufficiently cold, then they may be marked with large or small Xs for snow. The smaller markers imply light snow or flurries. In animation, areas of precipitation will appear to "paint" the surface blue.

Each cell also supports limited interaction. One can appear to navigate inside the presented scene, which is provided via a Java applet that displays one of nine distinct images with a fixed angular separation (10° in this case) depending upon the mouse position. The result is context for the product in order to facilitate the selection of alternative visualizations. Currently, only two choices are available. One is to examine a similar visualization, but with images of higher-resolution and greater fidelity. The other choice addresses the time-sampling problem. It provides an animation with frames every 10 minutes presented as an MPEG-1- compressed video viewable on a web page via a Java-based player or through a plug-in. To preserve the fidelity of the animation yet keep the size to only a few MB, source animations are generated at 720p resolution and then interpolated to 25% the number of pixels prior to MPEG encoding.

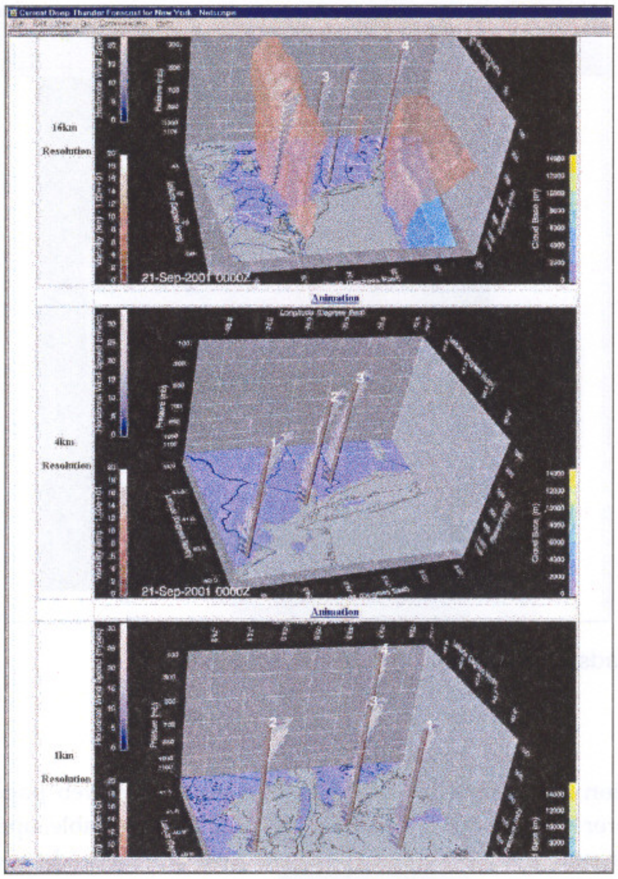

Fig. 2: Web-based 3D spreadsheet column for aviation products.

2.2 *Analysis products*

The ideas used for browse products have been applied in a limited fashion to the generation of images oriented toward analysis. The first case is shown in Figure 2 as single column with an aviation product (Class IV from *Treinish*, 1999). The three images correspond to the three nests from top to bottom. They are interactive like each cell of Figure 1.

Each image contains a brown, translucent surface in vertical pressure coordinates, corresponding to a boundary where the derived visibility is 10 km. This visibility is based upon extinction properties of cloud water, ice and precipitation, derived from the modelled upper air. Thus, the volume inside the surface represents relatively clear air, that is, visibility over 10 km. If no surface is visible then there are no clouds predicted at that time step, and thus, the visibility is high. At the bottom of the scene is a set of colored contours, typically, corresponding to the height of the forecasted cloud base as shown in the color legend to the lower right. Areas in gray imply no cloud data. The cloud base contours are overlaid with maps of coastline and state boundaries and rivers. The volume is marked at the locations of major airports with set of poles color contoured by the derived visibility using the color legend to the lower left. At each of 21 pressure levels, the horizontal wind is shown via arrows. The arrows are colored by horizontal wind speed following the legend to the upper left, which is also encoded by the arrow length. The only other visualization available is an animation as a set of individual JPEG images presented with a Javascript player. This more traditional approach is utilized given that there is one hour of forecast time between each frame.

The second case is shown in Figure 3 as single column with a product oriented toward atmospheric stability, especially for indicating the potential for severe weather (Class II from *Treinish*, 1999). Only one of three interactive images corresponding to the nests is shown in the figure. A colored surface is presented, where the color corresponds to K Index, following the top legend. The surface is deformed linearly by Lifted Index and overlaid with a set of contour lines of forecasted vertical wind speed using the second color legend. Significant updrafts (green to yellow contours) in blue "valleys" on the surface would imply regions of potential for severe convective activity. As with the previous example, the only other visualization available is an hourly animation.

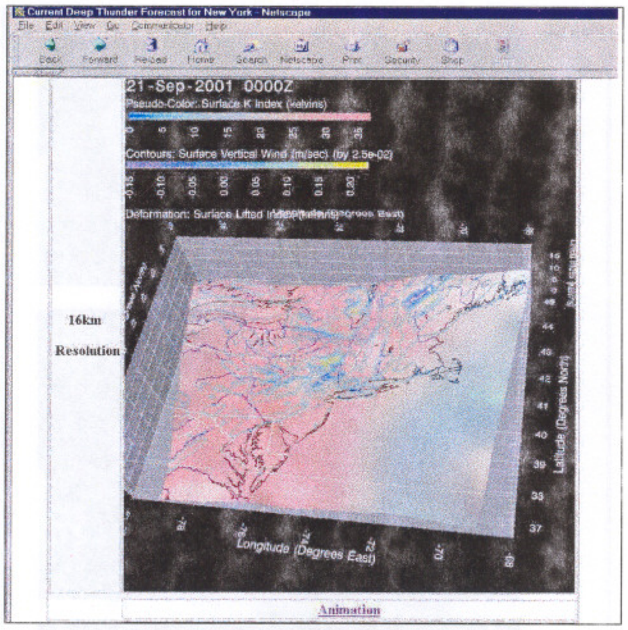

Fig. 3: Web-based 3D spreadsheet column for stability products.

4. Implementation

The data processing, realization and rendering required to populate the web pages discussed herein have been implemented with Data Explorer (DX) (*Abram and Treinish*, 1995). DX is a portable, open source, general-purpose software package for visualization and analysis (http://www.research.ibm.com/dx and http://www.opendx.org). A generic toolkit was used to avoid having to implement a graphics and computational infrastructure. Unlike traditional weather graphics or GIS, DX is parallelized for SMPs and can utilize OpenGL graphics accelerators. DX is built upon an unified data model that enables these applications to operate directly on the native gridded weather data.

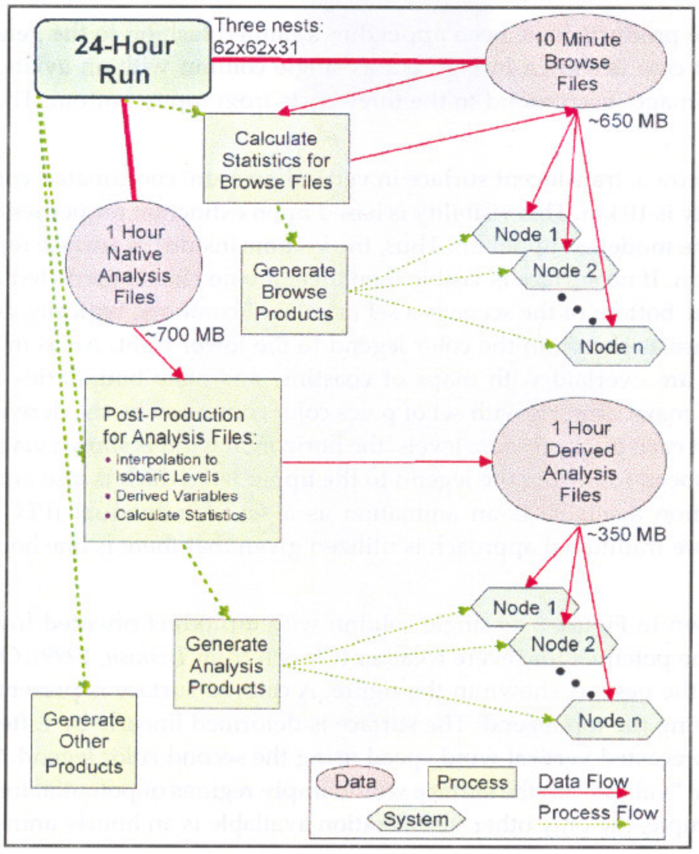

Fig. 4: Procedural and data flow for generation of web browser content.

Ordinarily, DX is used as an application enabler to implement interactive systems. DX is based upon a client-server architecture, where the client is the user interface and the server does the computation. Here, the client is used to develop a visualization, which yields a program in a high-level language that the server interprets and executes as a directed acyclic graph. Once the visualization is defined, it can be executed by the server independently of the user interface, which parameterizes the model run, variables to be processed, techniques to be utilized, style of output to be generated, etc. (as outlined in Treinish, 1999). These parameters then become the specification of a higher-level language for the execution of the program. The result is then a very compact representation of a complex visualization product that can be executed in batch, yet easily extended. Multiple instances of these graphs with differing inputs are then used to generate individual products. The output from each is one or more high quality images (i.e., at least workstation resolution). The images are post-processed employing tools from the open source ImageMagick suite (http://www.imagemagick.org) in order to provide material suitable for presentation in a web browser.

All of the DX and ImageMagick-based processing is embedded in a set of hierarchical scripts, which are invoked automatically after the completion of each model run. That work is illustrated schematically in Figure 4. This processing is parallelized for the same system used for the model execution (seven 4-way SMP nodes), but is supplemented with additional SMP (4-way AIX and 2-way Linux and NT) workstations clustered via a private Gigabit or public 100MB Ethernet. Each of these nodes, which is shown as blue hexagons in Figure 4, supports generation of two major product types - browse and analysis (yellow rectangles). In both cases, summary statistics are generated, which are used to enable scaling for the full dynamic range and to adapt the con-tent depending on model results. Then the work to create individual products (i.e., JPEG or MPEG files) is split up among the available nodes to run simultaneously. This simple parallelism enables the population of each browse spreadsheet and all available products to be completed in under 15 minutes. The html for the target web pages is written as templates. Hence, the files from each model run replace those derived from the previous one. Prior to the generation of the analysis products, a post-processor is executed to interpolate data from the model terrain-following coordinates to isobaric levels.

5. Extreme compression

Another interaction currently under development is focused on the data-sampling problem. It employs a specialized "extreme" level of compression based upon task-specific abstractions of underlying components of a visualization scene composed of several simpler geometric forms that are used multiple times such as in the browse products shown in Figure 1. They each can be represented by higher-order descriptions of the geometry. Hence, renderings of geometry are distributed instead of geometry itself. A typical Mpixel image of this type can be adequately described with only a few kb of data. Thus, animation sequences of a few hundred frames become very inexpensive to store and thus, transmit (i.e., one order of magnitude smaller than MPEG and two orders of magnitude for JPEG sequences). Since the description is geometric, it also provides the potential for limited direct manipulation like the aforementioned spreadsheet cells but unlike a pure animation sequence.

6. Conclusions and future work

This approach to web-based visualization has shown promise in an operational environment. The spreadsheet/abstraction notion can be further expanded to serve as an index for more traditional web-based visualizations as well as visualizations that can be developed on technologies becoming available to support entertainment and e-business applications (e.g., MPEG-4). In addition, the spreadsheet itself can be enhanced to incorporate other model results of relevance.

References

Abram, G. and L. Treinish. An Extended Data-Flow Architecture for Data Analysis and Visualization. Proceedings of the IEEE Visualization 1995 Conference, October 1995, Atlanta, pp. 263-270.

Jankun-Kelly, T. J. and K-L. Ma. A Spreadsheet Interface for Visualization Exploration. Proceedings of the IEEE Visualization 2000 Conference, October 2000, Salt Lake City, pp. 69-76, 541.

Hasler, F., K. Palaniappan and M. Manyin. A High-Performance Interactive Image Spreadsheet (IISS). *Computers in Physics*, 8, May/June 1994, pp. 325-342.

Hibbard, W. Vis5D Version 5.2. ftp://www.ssec.wisc.edu/pub/ vis5d-5.2/README .

Pielke, R. A., W. R. Cotton, R. L. Walko, C. J. Tremback, W. A. Lyons, L. D. Grasso, M. E. Nicholls, M.-D. Moran, D. A. Wesley, T. J. Lee and J. H. Copeland. A Comprehensive Meteorological Modeling System -RAMS. *Meteorology and Atmospheric Physics*, **49**, 1992, pp. 69-91.

Rueden, C. VisAD Visualization SpreadSheet. http:// www.ssec.wisc.edu/~curtis/ss.html .

Treinish, L. Task-Specific Visualization Design. *IEEE Computer Graphics and Applications*, **19**, n. 5, September / October 1999, pp. 72-77.

Treinish, L. Multi-Resolution Visualization Techniques for Nested Weather Models. Proceedings of the IEEE Visualization 2000 Conference, October 2000, Salt Lake City, pp. 513-516, 602.